EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 153 No. 9
September 2024

Research in Computing Science

Research in Computing Science, Año 23, Volumen 153, No. 9, septiembre de 2024, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de septiembre de 2024.

Research in Computing Science, year 23, Volume 153, No. 9, September 2024, is published monthly by the Center for Computing Research of IPN.

Volume 153(9)

Advances in Artificial Intelligence

María de Lourdes Martínez-Villaseñor (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2024

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Redes neuronales artificiales (RNA) para la predicción y análisis del patrón de comportamiento de aspirantes a la educación superior para la selección de institución universitaria, en la región de las montañas de Veracruz

Rita Flores-Asis, Mónica Karina González-Rosas, Marisol Rodríguez-Gasga

Universidad Veracruzana Campus Ixtaczoquitlán, Facultad de Negocios y Tecnologías, México

ritflores@uv.mx

Resumen. El presente trabajo expone el análisis de las variables que intervienen en la decisión de los aspirantes a cursar una carrera universitaria, con el objetivo de predecir el patrón de comportamiento y analizar el impacto de sus decisiones. Para llevar a cabo las predicciones se diseñó un instrumento de recolección de datos (encuesta) que fue aplicada a una muestra representativa de alumnos de los últimos semestres de la Educación Media Superior, con el fin de obtener una base de datos robusta, que fue utilizada para el entrenamiento de una RNA de alimentación hacia delante con entrenamiento de retro propagación programada en Neural Network de Matlab, con el algoritmo Levenberg-Marquardt, utilizando la función de entrenamiento TRAINLM. Las variables se categorizaron en tres segmentos, variables demográficas, variables de oferta educativa y variables de marketing y difusión. Se llevó a cabo el entrenamiento con un porcentaje de predicción superior el 87%, teniendo como resultado que las variables de mayor impacto en la selección del programa educativo, y la institución para estudiar la Universidad, fueron la *oferta educativa* de las universidades, con un 22.93% de impacto en el resultado , esto demuestra que los alumnos, si se interesan en elegir la profesión para la cual consideran tener vocación, también se aprecia que las variables *procedencia* y *distancia*, con un porcentaje 12.91% y 16.31 % respectivamente, también fueron relevantes en la decisión de los aspirantes.

Palabras clave: Redes neuronales artificiales, educación, captación, predicción.

Artificial Neural Networks (ANN) for the Prediction and Analysis of the Behavioral Pattern of Prospective Higher Education Applicants for University Institution Selection, in the Mountain Region of Veracruz

Abstract. This research presents an analysis of the variables involved in the decision-making process of prospective university students, aiming to predict

behavior patterns and analyze the impact of their decisions. To make predictions, a data collection instrument (survey) was designed and applied to a representative sample of students in the final semesters of High School Education. The purpose was to obtain a robust database used for training a feedforward neural network with backpropagation training programmed in Matlab's Neural Network Toolbox, using the Levenberg-Marquardt algorithm and the TRAINLM training function. The variables were categorized into three segments: demographic variables, educational offer variables, and marketing and outreach variables. The training achieved a prediction percentage above 87%, revealing that the most impactful variables in selecting an educational program and institution for university study were the universities' educational offerings, with a 22.93% impact on the outcome. This demonstrates that students are interested in choosing a profession they feel passionate about. Additionally, it was observed that variables such as background and distance, with percentages of 12.91% and 16.31% respectively, were also relevant factors in the decision-making process of the applicants.

Keywords: Artificial neural network, education, recruitment, prediction.

1. Introducción

En México existe un total de 5,003,087 estudiantes de Educación Media Superior (EMS) también conocida como preparatoria o bachillerato [1], es el nivel educativo que se cursa antes del ingreso a la educación superior, en algunos casos, es el último nivel que cursan los alumnos antes de incorporarse al ámbito laboral. Actualmente existen tres modelos educativos: el bachillerato general, el bachillerato tecnológico y el profesional técnico con bachillerato; y cuatro tipos de sostenimiento: federal, estatal, autónomo y privado, sumado a esto, existen 5 tipos de control administrativo y presupuestal que en total forman 35 subsistemas. En el Estado de Veracruz se matricularon 300 102 alumnos durante el ciclo escolar 2021-2023 [2]. A nivel estatal, se ha observado una importante disminución del porcentaje de alumnos egresados de nivel medio superior y superior, ya que en el ciclo escolar 2000-2001 se tuvo un porcentaje del 89.5% de ingresos en nivel medio superior de los cuales egresaron un total de 84% en nivel superior, en comparación con el ciclo 2020-2021, en el que se tiene un porcentaje de ingreso a nivel medio superior del 61.1%, teniendo un egreso en nivel superior de solo 47.7%, lo que indica que la proporción de alumnos que inician el estudio del nivel medio superior y terminan sus estudios de Licenciatura va disminuyendo considerablemente [3]. Es cierto que las razones pueden ser diversas, probablemente los alumnos no eligieron la carrera o universidad adecuada a sus posibilidades económicas, o la carrera elegida no fue la que logró cumplir con sus expectativas, quizá consideran que la universidad no cuenta con los servicios necesarios para la culminación de los estudios, y en muchos casos, puede ser que los alumnos no se encontraron informados adecuadamente de las posibilidades que tuvieron en ese momento, de la gama de universidades que ofrecieron sus servicios, y desconocían varios aspectos relevantes para la selección adecuada de una casa de estudios de nivel superior que cumpliera con sus expectativas.

Sin embargo, existen otras causas que no dependen precisamente de los alumnos, también muchos de estos aspectos se derivan de las universidades, algunas de ellas no lograron cumplir con las expectativas, planes y necesidades que buscaban los egresados del nivel medio superior.

La región de las Altas Montañas en Veracruz se conforma por diversos municipios que se encuentran ubicados en la zona centro del Estado, las principales ciudades en esta región la conforman Córdoba, Orizaba, Ixtaczoquitlán, Fortín y Huatusco. La región cuenta con 50 170 estudiantes de EMS [4] debido a la disponibilidad y las características de la muestra, se selecciona el estrato de estudiantes de la zona de Huatusco para estudiar a su población próxima a egresar de la EMS, para ello, se entrenó una red neuronal artificial (RNA) de alimentación hacia delante con entrenamiento de retropropagación programada en Neural Network de Matlab.

El diseño de la RNA es de tipo perceptrón multicapa, y se caracteriza porque las neuronas están organizadas en capas y sus conexiones entre ellas se orientan estrictamente hacia una sola dirección de una capa a otra. Las RNA´S han demostrado una mejor efectividad frente a otros métodos estadísticos de regresión sin necesidad de cumplir condiciones de linealidad, normalidad o tamaño muestral, [5].

Existen algunos autores que han llevado a cabo la implementación de redes neuronales artificiales para predecir el patrón de comportamiento de estudiantes en otros enfoques de propósito educativo, tal es el caso del estudio realizado en la Facultad de Ingeniería y Tecnología de la Información en Universidad Al-Azhar [6], en este trabajo se desarrolla una RNA para predecir el rendimiento escolar de los estudiantes, se observa que los resultados esperados del entrenamiento, se obtiene un porcentaje de predicción superior al 80% de los casos considerados. Este estudio mostró el potencial de la red neuronal artificial para predecir el desempeño de los estudiantes.

Por otro lado, en el trabajo realizado por [7] en esta investigación se entrenó una RNA para predecir el rendimiento académico estudiantil, los resultados demostraron que se clasificó adecuadamente el 73% de la muestra de prueba, lo que permite concluir que la RNA es correcta para identificar los estudiantes en estatus de reprobación, también menciona que se logró identificar las variables relevantes, el tiempo de estudio, las ausencias y el tiempo de uso de redes sociales fueron los factores más importantes para determinar la probabilidad de que un estudiante apruebe o no un curso.

Las RNA´s, también han sido de utilidad como una herramienta de apoyo para evaluar el proceso de aprendizaje de educación virtual [8] este estudio se evaluaron acciones y estrategias de seguimiento y retención de alumnos, el modelo predijo el rendimiento de los estudiantes con tasa de clasificación del 98,3%.

2. Metodología

2.1. Caracterización de la población meta

Para realizar el análisis de patrón de comportamiento de los alumnos que se encuentran por egresar de la EMS, se realizó una recolección de datos socioeconómicos y demográficos en un conjunto de ciudades que conforman la región montañosa del Estado de Veracruz, para ello, se llevó a cabo una recolección de datos de una muestra por estratos de alumnos aspirantes, las poblaciones de Córdoba, Orizaba, Fortín y

Tabla 1. Tabla de categorización de las variables de entrada.

Demográficos	Criterios académicos	Marketing y difusión
Procedencia	Oferta Educativa	Difusión
Edad	Distancia	Atención al cliente
Género	Becas	
Estado civil	Prestigio	
Ocupación	Modalidad	

Huatusco de la región de las Altas Montañas en Veracruz, México, son las ciudades que concentran un mayor número de estudiantes en la zona.

Para llevar a cabo instrumento de recolección se realizaron entrevistas con expertos psicólogos y pedagogos por medio del método Delphi, para redactar y determinar los planteamientos adecuados que debían ser incluidos en la encuesta que fue aplicada a la población muestreada, con la finalidad de identificar las principales variables que influyen en la selección de la universidad en la que pretender continuar con sus estudios universitarios.

Después de diseñar el instrumento de recolección, se llevó a cabo una investigación de campo, se visitaron diferentes instituciones educativas del nivel medio superior, para aplicarla a una muestra representativa de 6200 alumnos, que se encontraban cursando el quinto semestre, entre las instituciones que colaboraron, se pueden mencionar, alumnos de telebachillerato de Veracruz (TEBAEV), alumnos de bachillerato general y alumnos del Colegio de Bachilleres de Veracruz (COBAEV), cabe mencionar que los datos obtenidos han sido protegidos y han quedado a reserva de la institución, bajo su política de privacidad.

2.2. Entrenamiento de redes neuronales artificiales

Con el deseo de explorar y analizar el comportamiento de los datos sociodemográficos obtenidos por medio de las encuestas aplicadas, se optó por realizar el desarrollo de una red neuronal artificial (RNA) para realizar predicciones sobre los factores que influyen en la decisión de los alumnos de EMS en el momento de seleccionar su carrera profesional y la institución en la que continuaran con sus estudios, se consideraron 12 variables de entrada, que fueron obtenidas de un instrumento de recolección cualitativa, se estimaron parámetros numéricos para interpretar las expresiones lingüísticas resultantes de las encuestas, mismas que fueron categorizadas en 3 segmentos, en la tabla 1, se expone las variables de entrada que son categorizadas en variables demográficas, criterios académicos y variables de marketing y difusión de las universidades.

En la tabla 2, se presentan los parámetros que forman parte de las variables de entrada y salida de la red neuronal artificial, en la que se estima la predicción de la institución universitaria y la carrera profesional de preferencia de la población muestra.

Tabla 2. Definición de parámetros de las variables de entrada de las RNA´S.

Variable	Nombre	Definición	Unidad de medida
***x*1**	Procedencia	Lugar de procedencia	Km
***x*2**	Edad	Edad del aspirante	Numérica/años
***x*3**	Género	Sexo del aspirante	Numérica
***x*4**	Estado civil	Estado civil del aspirante	Numérica
***x*5**	Ocupación	Ocupación del aspirante	Numérica
***x*6**	Oferta educativa	Carreras profesionales ofertadas	Numérica
***x*7**	Distancia	Ubicación del plantel Universitario	Km
***x*8**	Becas	Plan de becas ofertadas por la institución	Numérica
***x*9**	Prestigio	Calificación ponderada sobre la percepción del prestigio institucional	Numérica
***x*10**	Modalidad	Opciones de acceso a la educación presencial, semipresencial o virtual.	Numérica
***x*11**	Difusión	Percepción de la información que difunde cada universidad.	Numérica
***x*12**	Atención al cliente	Percepción de la atención que ofrece cada universidad.	Numérica
***y*1**	Universidad seleccionada	Selección de universidad de la lista de opciones	Numérica
***y*2**	Programa seleccionado	Carrera profesional seleccionada	Numérica

"Una RNA es una técnica de Inteligencia Artificial (I.A.) que funciona a través de un modelo matemático que predice el patrón de comportamiento de sistemas lineales y no lineales" [9]. Las redes neuronales se agrupan en dos categorías en función del patrón de conexiones que presentan, las redes de alimentación hacia delante y las de retroalimentación o recurrentes.

Las redes de alimentación hacia adelante son aquellas donde no existen ciclos o retroalimentaciones, sus conexiones son unidireccionales y solo permiten señales hacia un solo sentido entre las neuronas de cada capa, y pueden ser de tipo Monocapa (de perceptrón simple), Multicapa (de perceptrón multicapa) y de función de base radial.

Los pasos para programar una red neuronal son: diseño de la arquitectura, entrenamiento, validación y prueba [10]. El diseño de una neurona artificial se representa en la Figura 1.

La expresión matemática utilizada en la neurona artificial se expresa en la Ecuación. (1). Los valores de (x_i) son los valores de entrada, (w_ji) son los coeficientes de los pesos de la neurona j, (b) es bias, que es considerado un valor de sesgo que permite ajustar los pesos de la neurona para lograr un error mínimo de salida, (y) son los valores de salida, y (f) es la función de transferencia sigmoide.

Los pesos de los datos de entrada son seleccionados de manera aleatoria, en un rango normalizado entre [-1,1], para lograrlo se utiliza la función de MAPMINMAX de Matlab para normalizar el valor mínimo y máximo de cada elemento de entrada y salida, ecuación (1):

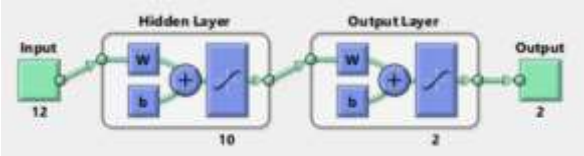


Fig. 2. Diagrama de configuración de la RNA de predicción de la selección Universitaria.

$$y_j = f\left[\sum_{i=0}^{n} w_{ji} x_i + b\right]. \quad (1)$$

La expresión matemática de la función Sigmoide se indica en la ecuación (2):

$$f(z) = \frac{1}{1+e^{-z}}. \quad (2)$$

El método de entrenamiento Levenberg-Marquardt generalmente es el más rápido, seguido del método BFGS Quasi-Newton, se recomiendan utilizarlos para redes que tienen un número pequeño o mínimo de salidas [11] por su mejor ajuste en problemas no lineales para minimizar la suma del error cuadrático medio.

Considerando esta recomendación este trabajo de investigación utiliza una RNA de alimentación hacia delante con entrenamiento de retropropagación programada en Neural Network de Matlab.

Se configuró la RNA con el algoritmo Levenberg-Marquardt, utilizando la función de entrenamiento TRAINLM. Se considera la función de aprendizaje de adaptación LEARNGDM, la función de desempeño MSE que expresa el error cuadrático medio y se definen dos capas ocultas, en la primera utiliza 10 neuronas y un bias, en la segunda capa solo una neurona y un bias. Ver figura 2.

El entrenamiento de la red tiene un error cuadrático medio (MSE) de 0.001 y la raíz del error cuadrático medio (RMSE) es de 0.0316. La validación se realiza con 5600 muestras cuyo resultado fue un MSE de 0.01 y un RMSE de 0.1, el coeficiente de determinación (R2) de predicción de la validación es de 0.89975, la Fig. 3 muestra su comportamiento.

La prueba se realiza con 3500 muestras nuevas para la predicción, logrando un MSE de 0.01 y un RMSE de 0.1, el coeficiente de determinación (R) de predicción de la prueba es 0.87514, teniendo un porcentaje aceptado para las pruebas de validación en una RNA, [12] ver Fig. 4. Los outliers que se observan en las figuras 4 y 5 representan algunos valores atípicos dado que la población no cumple con algunos de los criterios que se podían seleccionar en el instrumento de recolección, debido a que el instrumento de recolección se desarrolló con el objetivo de ser aplicado a diferentes estratos de la población meta, en el apartado de caracterización de la muestra se hace mención que para fines del análisis del presente trabajo, solo ha sido evaluado un estrato de la población objetivo.

3. Resultados y discusión

3.1. Predicción de la RNA

Después de realizar el entrenamiento y la validación del entrenamiento de la RNA, es necesario llevar a cabo la predicción de forma automática en los 5600 y 3500 casos que se consideraron para el entrenamiento de la red, en la tabla 3 se observa un

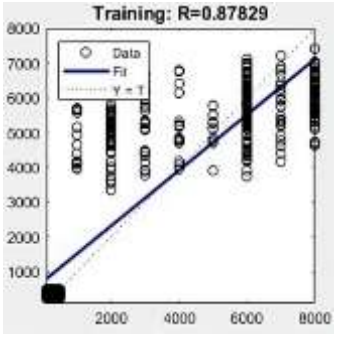


Fig. 3. Coeficiente de determinación (R) de la validación en la predicción durante la etapa de entrenamiento de la RNA en 6500 muestras.

fragmento de los casos predichos. La red fue entrenada en repetidas ocasiones, logrando un porcentaje de predicción superior al 89%, coincidiendo con los resultados obtenidos por [13] el cual menciona que entrenó una RNA con un porcentaje de predicción superior al 84%.

En la tabla 3, se presentaron los resultados obtenidos por la RNA, y en la tabla 4, se muestra la interpretación de parámetros cuantitativos migrados a parámetros cualitativos, en la cual se describe y especifica los valores numéricos que representan las diferentes etiquetas (alternativas) con respecto a las preferencias de los aspirantes a la Educación Superior, en relación con la Institución Universitaria a la que se pretenden postular y el área de conocimiento de su perfil profesional, teniendo como parámetros de entrada las variables $x1 \ldots x12$, los parámetros que se consideran en cada una de estas variables de entrada-salida se encuentran detallados en la tabla 2.

En la tabla 3 se observan 10 muestras de los resultados obtenidos en las predicciones. En la muestra 1, el parámetro 1 hace referencia a la variable de entrada $x1$ que corresponde al lugar de procedencia, dicho parámetro se encuentra estimado en Kilómetros, lo que indica que el lugar de procedencia es alrededor de 1 km de alguna de las alternativas de universidad a la que aspira ingresar el estudiante, $x2$ representa la edad del estudiante, en este caso 17 años, $x3$ se refiere al género, los valores utilizados para convertir de dato cualitativo a dato cuantitativo para facilitar su análisis por medio de una RNA, corresponden a 100 = mujer, 200 = hombre y 300= no binario, la variable $x4$ es el estado civil del encuestado, donde el valor 1 corresponde a la opción soltero y el valor 2 significa estar casado, la variable $x5$ se refiera a si el aspirante solo se dedica a estudiar (1000) si solo trabaja por el momento (2000) o trabaja y estudia (3000), en este caso se presenta un sesgo, ya que todos los aspirantes encuestados se dedican a estudiar, o bien trabajan y estudian, ya que la muestra seleccionada corresponde a estudiantes de EMS, la intención de mantener la opción de solo trabajar, se debe, a que se pretende ampliar la muestra de estudio a otros segmentos.

La variable de entrada $x6$ se refiere a la oferta educativa de interés, se cuentan con 8 opciones de oferta educativa, para la muestra 1, 4 representa el área de diseño digital, para el caso de la variable $x7$, se relaciona con la distancia de la ubicación del plantel, que se encuentra dada en Kilómetros, la variable $x8$ expresa las opciones de beca que ofrece el plantel, $x9$ mide el prestigio del plantel, cuya medida se puntúa en escala de 100 a 800, $x10$ representa la modalidad de estudio, donde 1 es presencial, 2 es virtual

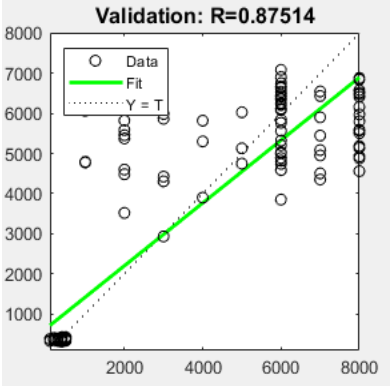


Fig. 4. Coeficiente de determinación (R) de la validación de la predicción durante la etapa de entrenamiento de la RNA en 3500 muestras.

y 3 es sistema abierto, para el caso de la variable $x11$, estima la percepción de la difusión de Universidad, la variable se puntúa de 1000 a 10000, con intervalos del 1000 en 1000, la última variable $x12$, muestra el parámetro de la percepción de calidad en atención al cliente, la medición se expresa de 100 a 100 con intervalos de 100 en 100.

Por último, las variables de respuesta (salida) se representan con $y1$ y $y2$, en la tabla 4 se precisa de manera detallada su equivalencia del parámetro numérico resultante a su etiqueta lingüística (cualitativa). Con estos resultados, podemos mencionar que las RNA´S permiten entrenar valores numéricos, obteniendo parámetros cuantitativos altamente confiables, que pueden ser interpretados para medir y exhibir aspectos cualitativos de la población analizada [14], con el fin de estimar su patrón de comportamiento.

Las redes neuronales artificiales tienen la capacidad de exponer un panorama de las variables relevantes en las predicciones por medio del entrenamiento de datos. Utilizando el software Neural Tools 5.5, de PALISADE, se llevó a cabo un segundo análisis de los datos, por medio de una red neuronal de regresión generalizada utilizada para la predicción y clasificación categórica (PNN/GRNN), con la finalidad de evaluar el impacto, una de las características que ofrece como resultado dicho software, es la posibilidad de ofrecer un porcentaje de impacto en las variables de perdición.

El gráfico de tornado que se muestra en la figura 5, demuestra el porcentaje de impacto de cada variable de entrada, con respecto a la variable de salida, se puede apreciar que la variable que representa el mayor porcentaje de impacto en la selección del programa educativo, y la institución para estudiar la Universidad es la oferta educativa, con un 22.93% , esto demuestra que los alumnos, si se interesan en elegir la profesión para la cual consideran tener vocación, también se aprecia que las variables procedencia y distancia, con un porcentaje 12.91% y 16.31 % respectivamente, lo cual hace referencia a los kilómetros de distancia de su lugar de vivienda y la distancia a la universidad predilecta , son dos variables que son detonantes al momento de decidir su futuro, con esto verificamos que las RNA´S son una herramienta viable para obtener resultados confiables en niveles de presión e interpretación de una gran cantidad de datos [15].

Tabla 3. Fragmento de resultados de la predicción de la RNA.

Muestras	**Variables de entrada**												**Variables de Salida**	
	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	$x7$	$x8$	$x9$	$x10$	$x11$	$x12$	$y1$	$y2$
1	1	100	17	1	1000	4	10	10	800	1	5000	200	4000	5
2	6	100	18	1	2000	7	16	5	100	1	1000	100	4000	2
3	4	200	17	1	1000	2	22	8	150	1	4000	300	1000	2
4	7	100	16	1	1000	1	55	9	200	2	1000	500	5000	1
5	2	200	18	1	1000	2	16	4	800	1	1000	100	4000	3
6	7	200	17	1	3000	8	22	6	700	3	2000	300	5000	2
7	2	200	17	2	1000	2	2	8	800	1	6000	900	2000	2
8	4	100	17	1	2000	6	135	5	100	1	7000	100	3000	1
9	8	200	18	1	1000	1	296	2	300	2	1000	300	4000	5
10	1	100	19	1	1000	4	55	8	100	1	2000	600	7000	3

Tabla 4. Interpretación de los parámetros numéricos resultantes en la elección de la población muestra.

$y1$	**Universidad**	$y2$	**Programa educativo categorizado**
1000	Tecnológico de Huatusco	1	Prof. Humanidades
2000	UTCV	2	Prof. econó-administrativo
3000	BUAP	3	Prof. área médica
4000	UV	4	Prof. en Artes
5000	Tecnológico de Orizaba	5	Prof. en Ingeniería
6000	UNAM		
7000	Virtual		
8000	Privada		

4. Conclusiones

El conocer las necesidades y expectativas de los estudiantes, permitirá identificar la demanda educativa de educación Superior de la región y las carreras profesionales de mayor solicitud por parte de los alumnos, así como identificar la cantidad de estudiantes que prefieren emigrar de la región para buscar otra oferta educativa o incluso otras oportunidades de empleo, también permite saber cuántos alumnos tienen altas intenciones de acceder a los principales programas de apoyo social, ya sea para solicitar una beca o ser parte de uno de los programas de capacitación laboral.

Podemos deducir que las redes neuronales artificiales son una herramienta de minería de datos muy útil para la predicción de datos cuantitativos y cualitativos, que proporcionaron escenarios de las expectativas y decisiones de los aspirantes a la educación Superior, además de mostrar un análisis cuantitativo para estimar

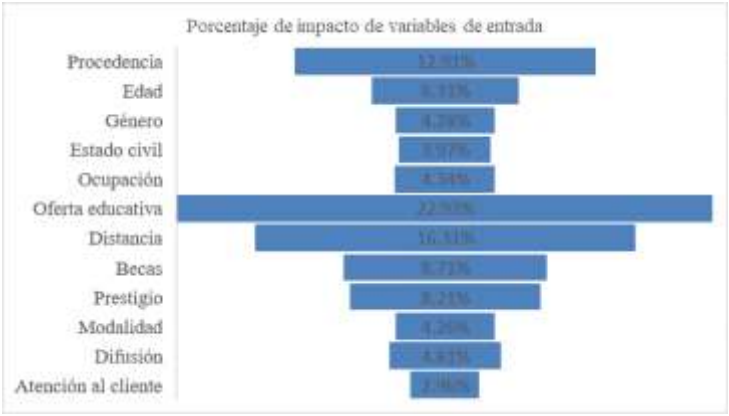


Fig. 5. Gráfica de porcentaje de impacto de las variables de entrada con las variables de salida.

indicadores de migración, la movilidad en las comunidades de la región, la empleabilidad, y el impacto económico. Los resultados expuestos en el presente trabajo de investigación pueden ser utilizados como base o fundamento, para detonar estrategias de captación de alumnos, por parte de las instituciones educativas con el fin de fomentar y gestionar el desarrollo social y educativo de la región de las montañas de Veracruz.

Referencias

1. Toscano, L.R.: Análisis de la educación en México: Barreras y limitantes para la congruencia, la calidad y la cobertura educativa actual. Ciencia Latina Revista Científica Multidisciplinar, vol. 7, pp. 4851–4883 (2023). DOI: 10.37811/cl_rcm.v7i1.4805.
2. Huerta-Estévez, A., Severino-Parra, C.A., León, F.V.: Agenda 2030 y educación de calidad en México, avances en el cumplimiento para el 2030. RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo, vol. 14, no. 27, pp. e517 (2023). DOI: 10.23913/ride.v14i27.1567.
3. INEGI.: Informe anual de actividades y resultados 2022. Órgano Interno de Control. Instituto Nacional de Estadística y Geografía, pp. 1–140. https://ci.inegi.org.mx/docs/InformeAnual2022OIC.pdf. (2022)
4. INEGI.: Anuario estadístico y geográfico por entidad federativa 2017. Instituto Nacional de Estadística y Geografía, pp. 1–641. https://www.inegi.org.mx/contenidos/productos/prod_serv/contenidos/espanol/bvinegi/productos/nueva_estruc/aegef_2017/702825097929.pdf. (2017)
5. Inicio-Flores, F.A., Capuñay-Sanchez, D.L., Estela-Urbina, R.O., Delgado-Soto, J.A., & Vergara-Medrano, S.E.: Diseño e implementación de una red neuronal artificial para predecir el rendimiento académico en estudiantes de Ingeniería Civil de la UNIFSLB. Revista Veritas et Scientia-UPT, vol. 10, no. 1, pp. 107–117 (2021). DOI: 10.47796/ves.v10i1.464.
6. Naser, S.A., Zaqout, I., Ghosh, M.A., Atallah, R., & Alajrami, E.: Predicting student performance using artificial neural network: In the faculty of engineering and information technology. International journal of hybrid information technology, vol. 8, no. 2, pp. 221–228 (2015). DOI: 10.14257/ijhit.2015.8.2.20.

7. Gil-Vera, V.D., Quintero-López, C.: Predicción del rendimiento académico estudiantil con redes neuronales artificiales. Información tecnológica vol. 32, no. 6, pp. 221–228 (2021). DOI: 10.4067/s0718-07642021000600221.
8. Zacharis, N.Z. Predicting student academic performance in blended learning using artificial neural networks. International Journal of Artificial Intelligence and Applications, vol. 7, no. 5, pp. 17–29 (2016). DOI: 10.5121/ijaia.2016.7502.
9. Flores-Asis, R., Méndez-Contreras, J.M., Juárez-Martínez, U., Alvarado-Lassman, A., Villanueva-Vásquez, D., & Aguilar-Lasserre, A.A.: Use of artificial neuronal networks for prediction of the control parameters in the process of anaerobic digestion with thermal pretreatment. Journal of Environmental Science and Health, Part A, vol. 53, no. 10, pp. 883–890 (2018). DOI: 10.1080/10934529.2018.1459070.
10. Purroy-Vasquez, R., Aguilar-Lasserre, A.A., Meza-Palacios, R., & Fernández-Lambert, G. Artificial neural network (ANN) in forecasting of poverty line and economic-energetic efficiencies into the maize-based agroecosystems. Archives of Agronomy and Soil Science, vol. 70, no. 1, pp. 1–17 (2024). DOI: 10.1080/03650340.2023.2287751.
11. Miranda-Ackerman, M.A., Azzaro-Pantel, C., Aguilar-Lasserre, A.A.: A green supply chain network design framework for the processed food industry: Application to the orange juice agrofood cluster. Computers & Industrial Engineering, vol. 109, pp. 369–389 (2017). DOI: 10.1016/j.cie.2017.04.031.
12. Bellido-Anicama, A.B., Schwarz-Diaz, M.: Redes neuronales para predecir el comportamiento del conjunto de activos financieros más líquidos del mercado de valores peruano. Revista Científica de la UCSA, vol. 6, pp. 49–64 (2019). DOI: 10.18004/ucsa /2409-8752/2019.006(01)049-064.
13. Lau, E.T., Sun, L., Yang, Q. Modelling, prediction and classification of student academic performance using artificial neural networks. SN Applied Sciences, vol. 1, e982 (2019). DOI: 10.1007/s42452-019-0884-7.
14. Mohammadi, M., Khorrami, M.K., Ghasemzadeh, H., Noor, P., & Zandbaaf, S.: Artificial neural network for quantitative and qualitative determination of the viscosity of nanofluids by ATR-FTIR spectrometry. Infrared Physics & Technology, vol. 118, pp. 103900 (2021). DOI: 10.1016/j.infrared.2021.103900.
15. Marinó, GC., Petrini, A., Malchiodi, D., Frasca, M.: Deep neural networks compression: A comparative survey and choice recommendations. Neurocomputing, vol. 520, pp. 152–170 (2023). DOI: 10.1016/j.neucom.2022.11.072.

ISSN 1870-4069

Detección de enfermedades cardiacas: Implementación de clasificador probabilístico en un dispositivo embebido

Marcos Julián Benítez-Rodríguez, Diego Pérez-Vega,
Jorge Luis Pérez-Ramos, Selene Ramírez-Rosales,
Luis Antonio Díaz-Jiménez, Ana Marcela Herrera-Navarro,
Hugo Jiménez-Hernández, Daniel Cantón-Enríquez

Universidad Autónoma de Querétaro,
Facultad de Informática,
México

{mbenitez07, dperez126}@alumnos.uaq.mx, daniel.canton@uaq.mx

Resumen. El diagnóstico temprano de enfermedades cardíacas desempeña un papel crucial en la toma de decisiones médicas para mejorar la salud de los pacientes con afecciones cardíacas. Una estrategia eficaz para este propósito implica la aplicación de técnicas de aprendizaje automático, las cuales facilitan la identificación de patrones y la comprensión de síntomas relacionados con estas enfermedades. En este trabajo, se ha implementado un clasificador probabilístico en un dispositivo embebido para la detección oportuna de enfermedades cardíacas. Este modelo utiliza la regla de Bayes junto con hipótesis simplificadoras que se basan en la suposición de independencia probabilística entre las variables clínicas. Durante el proceso de aprendizaje de los parámetros se asumen que cada variable clínica cuenta con una distribución normal. Posteriormente, se evalúa el rendimiento del modelo utilizando métricas de la matriz de confusión. Los resultados obtenidos muestran que el clasificador probabilístico propuesto mejora su rendimiento en comparación con otros trabajos consultados en la literatura. Además, se resaltan las ventajas de implementar el modelo propuesto en un dispositivo embebido. Así como, los trabajos a futuro a realizar en la investigación.

Palabras clave: Aplicaciones médicas, aprendizaje automático, detección de enfermedad cardiaca, razonamiento probabilístico, sistema embebido.

Heart Disease Detection: Implementation of Probabilistic Classifier in an Embedded Device

Abstract. Early diagnosis of heart disease plays a crucial role in medical decision making to improve the health of patients with cardiac conditions. An effective strategy for this purpose involves the application of machine learning techniques, which facilitate the identification of patterns and the understanding of symptoms related to these diseases. In this work, a probabilistic classifier has been implemented in an embedded device for timely detection of heart disease. This model uses Bayes' rule together with simplifying assumptions based on the

assumption of probabilistic independence between clinical variables. During the parameter learning process, each clinical variable is assumed to have a normal distribution. Subsequently, the performance of the model is evaluated using confusion matrix metrics. The results obtained show that the proposed probabilistic classifier improves its performance compared to other works consulted in the literature. Furthermore, the advantages of implementing the proposed model in an embedded device are highlighted. As well as the future works to be carried out in the research.

Keywords: Coronary artery disease, machine learning, heart disease detection, probabilistic reasoning, embedded system.

1. Introducción

Las enfermedades cardiovasculares son la principal causa de muerte en el mundo, cobrando la vida de aproximadamente 17.9 millones de personas cada año según la Organización Mundial de la Salud [1]. En México, los infartos de miocardio y los accidentes cerebrovasculares son responsables de alrededor de 150,000 muertes anuales, lo que destaca la importancia de prestar atención al infarto agudo de miocardio como una prioridad en la atención médica [2].

En esta situación [3], varios elementos dificultan la identificación temprana de enfermedades cardíacas: *i)* la falta de disponibilidad de cardiólogos, profesionales médicos especializados en estas afecciones; *ii)* una distribución desigual de los cardiólogos en el territorio, con una concentración en las principales zonas urbanas, y *iii)* factores relacionados con la alimentación, como el alto consumo de sal, la falta de actividad física y el tabaquismo.

Por otro lado, [4] menciona que el rápido desarrollo económico y el estilo de vida acelerado pueden predisponer a las personas a enfermedades crónicas, donde los síntomas suelen manifestarse en etapas avanzadas, lo que dificulta los tratamientos efectivos en etapas tempranas. Por lo tanto, subraya la importancia de fortalecer la atención médica comunitaria y buscar alternativas para la detección temprana de estas patologías.

El monitoreo continuo de los índices fisiológicos más representativos a través del uso de tecnologías digitales ha tenido una aceptación creciente en los últimos años [5]. Esto ha ocasionado que aspectos como históricos de algunas variables fisiológicas sean utilizadas para la predicción de ciertas enfermedades.

No obstante, las técnicas de aprendizaje automático se erigen como un pilar fundamental en el desarrollo de herramientas computacionales que ayuden en la detección oportuna de enfermedades cardíacas. En estudios relacionados a resolver esta problemática, se ha contado con métodos predictivos, tales como, redes neuronales multicapa [6], *random forest* [7] y máquinas de soporte vectorial [8].

Por otra parte, el procesamiento de información proveniente de sensores, en comparación con las mediciones médicas convencionales, ha demostrado ser especialmente prometedor en la detección temprana de enfermedades cardiacas [9]. En este sentido las investigaciones destacan el papel crucial de estas tecnologías en la personalización de la democratización de la tecnología mediante un acercamiento global a la población, de forma que se puedan promocionar estilos de vida saludables

y accesibles para un mayor número de individuos, así como en la identificación temprana de la necesidad de atención médica [10].

En el presente trabajo, se implementa un clasificador probabilístico en un sistema embebido para la detección oportuna de un tipo de enfermedad cardiaca, arterias coronarias. Para ello, se utiliza una placa Raspberry Pi 4 como dispositivo de aplicación remota para la evaluación de nuevos usuarios que no fueron entrenados previamente.

El resto del artículo está organizado de la siguiente manera: se presentan los materiales y métodos utilizados para la implementación del clasificador probabilístico en un sistema embebido; después, se muestran los resultados obtenidos del aprendizaje y la evaluación del clasificador; posteriormente, la discusión de los resultados; por último, las conclusiones y trabajos a futuro.

2. Marco teórico

En esta sección se explican los fundamentos teóricos en los que se basa la presente investigación. Primero, se habla acerca del dispositivo utilizado, así como detalles técnicos del mismo. Luego, se habla de forma general acerca de las diferentes técnicas de aprendizaje automático que hay en el estado de arte. Por último, se revisa el modelo matemático del clasificador probabilístico implementado.

2.1. Dispositivo embebido

La placa Raspberry Pi 4, está orientada hacia sistemas embebidos y multipropósito. Algunas características de la placa Raspberry Pi 4 se describen en la Tabla 1 [11]. Por otro lado, se ha elegido la placa Raspberry Pi por su versatilidad para aplicar un clasificador probabilístico que aprende mediante análisis clínicos de pacientes que puedan tener una enfermedad de arterias coronarias.

Además, como trabajo a futuro se pretende utilizar un sensor conectado a los pines de la placa Raspberry Pi, por ejemplo, el registro de la actividad eléctrica del corazón. Por último, una característica poderosa de la placa Raspberry Pi 4 es su fila de pines GPIO (general-purpose input/output) a lo largo del borde superior de la placa, véase Figura 1.

2.2. Aprendizaje automático

El aprendizaje automático (AA) es parcialmente un campo de la inteligencia artificial, que se enfoca en la toma de decisiones en condiciones de incertidumbre a partir del aprendizaje de datos. Por otro lado, AA es un proceso de dos fases que implica la selección de características relevantes y la adaptación del modelo en función de estas características [12].

No obstante, el aprendizaje automático se centra en tres conceptos principales: datos, modelo y aprendizaje [13]. Los datos son fundamentales para el funcionamiento del aprendizaje automático, y los modelos se diseñan para detectar patrones útiles en los datos. Además, los algoritmos de aprendizaje automático pueden ser supervisados, no supervisados, o por refuerzo, dependiendo del conocimiento a priori disponible y los objetivos del aprendizaje.

Tabla 1. Principales características de la placa Raspberry Pi 4. Elaboración propia.

Características	**Descripción**
Tamaño	88mm x 58mm x 19.5mm
Procesador	ARM Cortex A72
Velocidad	hasta 2.50GHz
RAM	de 2, 4 y 8 GB
Puertos GPIO	40 pines
Entradas	2 micro HDMI, 2 USB 2.0, 2 USB 3.0 Micro SD, conector de audio tipo jack y alimentación de USB-C
Sistema Operativo	Raspbian

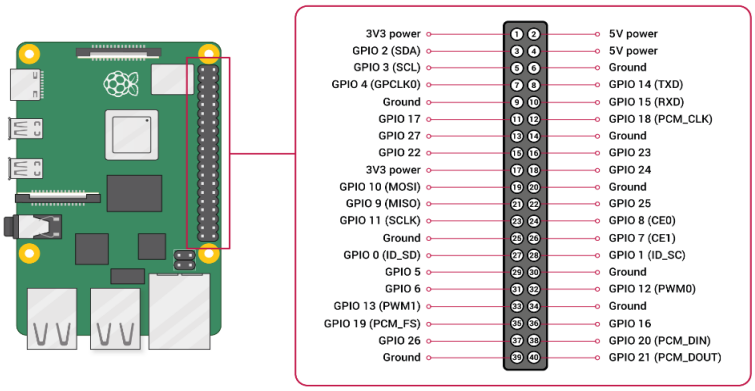


Fig. 1. Ubicación y posición de los pines integrados en la Raspberry Pi 4. Recuperado de [10].

El aprendizaje supervisado implica la deducción de una correspondencia entre entradas y salidas conocidas, mientras que el aprendizaje no supervisado busca modelar la estructura o distribución de los datos sin información previa sobre las salidas. El aprendizaje por refuerzo se basa en el proceso de ensayo y error para maximizar una función de recompensa a largo plazo [14].

Por otro lado, los algoritmos de aprendizaje automático se pueden dividir de acuerdo con el problema que se busca resolver. En la Figura 2, se muestran las principales tareas dentro del aprendizaje automático, las cuales varían de acuerdo con distintos autores [13, 14].

2.3. Clasificador probabilístico

El clasificador probabilístico es un modelo condicional que resuelve un problema de clasificación, representado por un vector $x = (x_1, \dots, x_n)$ donde n representa la n-ésima característica asignada como probabilidad, la misma se define como:

$$P(C_k|F_1, F_2, \dots, F_n), \tag{1}$$

Fig. 2. Tareas de aprendizaje automático de acuerdo con su aprendizaje.

donde cada k son los posibles resultados de una clase C_k. Dicha variable está condicionada al cumplimiento de ciertas variables independientes $x_1, x_2, \dots, x_n$, basadas en el teorema de Bayes, se reescribe la ecuación Ecuación 1 como Ecuación 2:

$$P(C_k|x_1, x_2, \dots, x_n) = \frac{P(C_k) \cdot P(\boldsymbol{x}|C_k)}{P(\boldsymbol{x})}. \tag{2}$$

En la práctica, se presta especial atención en el numerador, ya que el denominador es independiente de C_k. Por lo tanto, se tienen claro conocimiento de los valores x_i de modo que el denominador es constante. Así mismo, en el numerador, se aplica la regla del producto para eventos dependientes, véase Ecuación 3:

$$P(x_1, x_2, \dots, x_n, C_k), \tag{3}$$

donde $P(x_1, x_2, \dots, x_n, C_k)$ es una probabilidad conjunta, es decir, $x_1, x_2, \dots, x_n, C_k \equiv x_1 \cap x_2 \cap \dots \cap x_n \cap C_k$, la cual se reescribe utilizando la regla de la cadena para eventos repetidos de la definición de probabilidad condicional, véase Ecuación 4:

$$\begin{aligned} P(x_1, x_2, \dots, x_n, C_k) = P(x_1|x_2, \dots, x_n, C_k) \cdot P(x_2|x_3, \dots, x_n, C_k) \cdot P(x_{n-1}|x_n, C_k) \\ \cdot P(x_n|C_k) \cdot P(C_k). \end{aligned} \tag{4}$$

Luego, se toma en cuenta el concepto de independencia probabilística, donde, se asume que cada x_i variable es independiente de cualquier otra x_j para $i \neq j$ cuando se encuentran condicionadas a C_k, véase Ecuación 5:

$$P(x_i|x_{i+1}, \dots, x_n, C_k) = P(x_i|C_k). \tag{5}$$

En consecuencia, el modelo de probabilidad conjunta se expresa en la Ecuación 6:

$$\begin{aligned} P(C_k|x_1, x_2, \dots, x_n) \propto P(x_1, x_2, \dots, x_n, C_k) \propto P(C_k) \cdot P(x_1|C) \cdot P(x_2|C_k) \cdots \\ P(x_n|C_k) \propto P(C_k) \cdot \prod_{i=1}^{n} P(x_i|C_k). \end{aligned} \tag{6}$$

Tabla 2. Variables del conjunto de datos utilizado. Elaboración propia.

Nombre	Descripción
Age	Edad en años
Sex	1 = hombre 0 = mujer
Cp	Tipo de dolor en el pecho: 1 = angina típica 2 = angina atípica 3 = pa sin angina 4 = asintomático
Trestbps	Presión arterial en reposo (en mm Hg al ingreso al hospital)
Chol	Colesterol sérico en mg / dl
Fbs	Azúcar en sangre en ayunas >120 mg/dl (1 – verdadero; 0 = falso)
Restecg	Resultados electrocardiográficos en reposo (0 = normal; 1 = teniendo ST-T; 2 = hipertrofia)
Thalach	Frecuencia cardiaca máxima alcanzada
Exang	Angina inducida por ejercicio (1 = sí; 2 = no)
Oldpeak	Depresión del ST inducida por el ejercicio en relación con el reposo
Slope	pendiente del segmento ST de ejercicio pico (1 = pendiente ascendente; 2 = plano; 3 = pendiente descendente)
ca	número de vasos principales (0-3) coloreados por la floración
Thal	1 = normal; 2 = defecto fijo; 3 = defecto reversible

De modo que, bajo los supuestos de independencia anteriores, el modelo probabilístico con enfoque Bayesiano se define en la Ecuación 7:

$$P(C_k|x_1, x_2, \dots, x_n) = \frac{1}{Z} P(C_k) \cdot \prod_{i=1}^{n} P(x_i|C_k), \tag{7}$$

donde Z es un factor que depende exclusivamente de la evidencia de las características. Por último, se combina el modelo Bayesiano con una regla de decisión, en este caso se escoge la hipótesis que sea más probable para minimizar la probabilidad de clasificación errónea. El clasificador resultante se define en la Ecuación 8:

$$\hat{y} = \underbrace{argmax}_{k \in \{1,\dots,k\}} [P(C_k) \prod_{i=1}^{n} P(x_i|C_k)], \tag{8}$$

3. Metodología

La metodología seguida en el proyecto se aborda en la Figura 3. Primero, se describe el conjunto de datos utilizado. Luego, se menciona la etapa de aprendizaje para el entrenamiento del clasificador probabilístico. Por último, se revisa la etapa de evaluación de la implementación.

3.1. Descripción de los datos

El conjunto de datos utilizados fue "*Heart Disease*" del repositorio de datos para aprendizaje automático de la Universidad de California Irving [15]. Los datos fueron medidos y recolectados por la Fundación Clínica de Cleveland (FCC), con 1025 datos

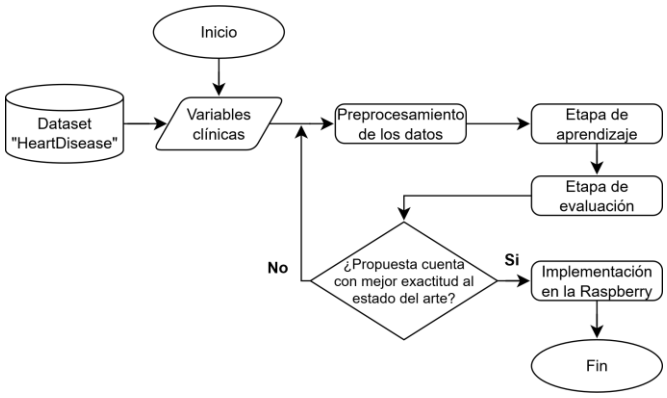


Fig. 3. Esquema de la implementación del clasificador probabilístico en un dispositivo embebido.

de pacientes como muestra. En la Tabla 2, se muestran las trece características del conjunto datos. Además, la distribución de los datos es de 499 casos con enfermedad de arterias coronarias (49%) y 526 casos con ausencia de enfermedad de arterias coronarias (51%). Para la experimentación, el conjunto de datos "*Heart Disease*" fue dividido en dos subconjuntos: el entrenamiento con 75% de los datos, y el de pruebas con 25% de la información restante, seleccionado de mediante un muestreo aleatorio simple.

3.2. Etapa de aprendizaje

Cada característica tiene una distribución empírica especifica, la cual depende de la naturaleza de los datos. En este trabajo, se asumen que las características utilizadas tienen una distribución Gaussiana, véase la Ecuación 9:

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \tag{9}$$

donde el parámetro μ es la media o valor esperado de la distribución, mientras que el parámetro σ es la desviación típica, véase en las Ecuaciones 10 y 11:

$$\mu = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i\,, \tag{10}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}\ . \tag{11}$$

3.3. Etapa de evaluación

La evaluación el clasificador probabilístico se llevó a cabo mediante la matriz de confusión. Los elementos de la matriz de confusión son: *i)* Verdaderos Positivos (VP), pacientes que tenían enfermedades cardiacas y estaban correctamente diagnosticados; *ii)* Verdaderos Negativos (VN), pacientes que no tenían enfermedades cardiacas y

estaban correctamente diagnosticados; *iii)* Falsos Negativos (FN), pacientes que tenían enfermedades cardiacas y fueron mal diagnosticados; y *iv)* Falsos Positivos (FP), pacientes que no tenían enfermedades cardiacas y fueron mal diagnósticos. En la industria médica, los FN son las predicciones más peligrosas. Las diferentes métricas de rendimiento se calcularon utilizando una matriz de confusión. La fórmula de exactitud está dada por la Ecuación 12:

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP}. \tag{12}$$

La precisión es el valor positivo predicho definido por la Ecuación 13:

$$Precisión = \frac{VP}{VP + FP}. \tag{13}$$

La exhaustividad es la proporción de pacientes con enfermedades cardiacas, véase Ecuación 14:

$$Exhaustividad = \frac{VP}{VP+FN}, \tag{14}$$

El valor F1, también conocido como Score-F1, es considerado un promedio armónico entre la precisión y la exhaustividad, véase Ecuación 15:

$$F1_{Score} = 2\left(\frac{Precisión \cdot Exhautividad}{Precisión + Exhautividad}\right). \tag{15}$$

3.4. Implementación en la Raspberry

Los datos de los análisis clínicos se dividen en dos subconjuntos. Primero, el conjunto de entrenamiento es enviado a la placa Raspberry para el aprendizaje de los parámetros. Después, los datos del subconjunto de pruebas son enviados a la placa de Raspberry para predecir si una persona tiene o no una enfermedad de arterias coronarias. Posteriormente, el rendimiento del clasificador se visualiza en una pantalla o monitor que va conectado a la placa Raspberry Pi, véase Figura 4.

4. Resultados

A continuación, se discute el rendimiento del clasificador probabilístico desarrollado en GNU Octave 8.2.0. En la Tabla 3, se contrasta los rendimientos del clasificador probabilístico con respecto a otros trabajos consultados en la revisión de la literatura.

Basado en los resultados, se demuestra que el clasificador probabilístico asumiendo que las características presentan una distribución Gaussiana conlleva a una mejora sustancial en casi todas las métricas, con excepción de la medición F1.

Los resultados anteriores, son importantes dado la importancia de detectar de forma temprana, si una persona tiene o no una enfermedad cardiaca. No obstante, el clasificador propuesto y los demás trabajos consultados pueden llegar a clasificar erróneamente a una persona como enferma cuando no lo esté. Así mismo, se puede clasificar a una persona como no enferma cuando si tenga una enfermedad cardiaca.

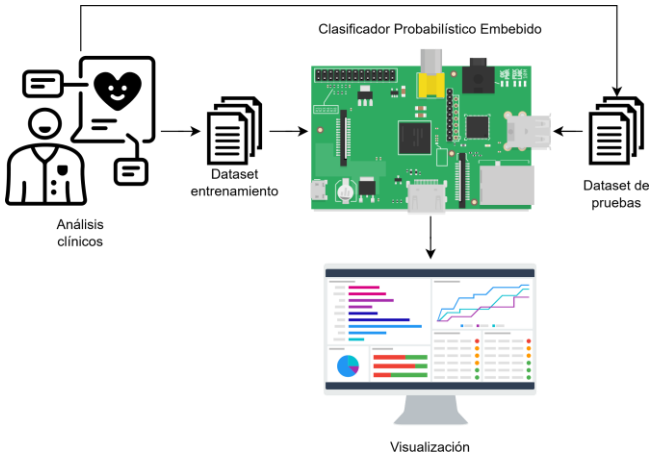


Fig. 4. Arquitectura general.

Tabla 3. Comparación del rendimiento del clasificador probabilístico.

Modelo	Exactitud	Precisión	Exhaustividad	F1
RNA Multicapa [5]	80%	–	–	–
Random Forest [6]	85%	86%	–	**92%**
SVM [7]	82%	90%	–	–
Propuesta (Gaussiana)	**87%**	**94%**	**82%**	88%

Por lo que, se recomienda que el diagnóstico dado por este clasificador sea validado por un médico cardiólogo.

Por otro lado, el uso de la Raspberry Pi 4 puede desempeñar un papel crucial en la detección temprana de enfermedades cardíacas al servir como plataforma de bajo costo. Además, de ser eficiente en la implementación algoritmos de aprendizaje automático, permitiendo así un diagnóstico más rápido y preciso en entornos médicos de difícil acceso.

5. Conclusiones

En este trabajo, se ha elaborado un clasificador probabilístico para realizar diagnósticos tempranos de la enfermedad de arterias coronarias. Este clasificador fue implementado en un sistema embebido de bajos requerimientos usando GNU-Octave. El fundamento de este modelo se basa en la regla de Bayes y en la suposición de independencia entre variables. Sin embargo, aunque el enfoque bayesiano dado al clasificador puede ser efectivo en diversos contextos, en ocasiones su desempeño disminuye debido a la falta de independencia condicional entre las características. Además, algunas características son de tipo discretas, por lo que la suposición de una distribución Gaussiana no siempre es la más apropiada.

Por otro lado, se destaca la importancia de adoptar nuevas tecnologías en el diagnóstico temprano de enfermedades cardíacas, enfatizando su eficacia y seguridad en entornos hospitalarios. La presente implementación se puede emplear estas en áreas remotas o rurales donde hay carencias o no hay disponibilidad de médicos cardiólogos.

Así como, facilitar a los pacientes un acceso más rápido y confiable en sus tratamientos críticos. No obstante, se recomienda que la detección estimada por el clasificador probabilístico sea validad por un médico cardiólogo.

Como futuras investigaciones, se sugiere considerar las dependencias entre las variables, es decir, construir una red Bayesiana que tome en cuenta las relaciones causales entre ellas. Luego, se propone identificar las variables más influyentes en el estudio mediante un análisis multivariable. Además, se destaca que este clasificador probabilístico puede ser aplicado en cualquier otro problema de muestreo. Por último, como trabajo a futuro se pretende utilizar diferentes sensores conectado a la placa Raspberry Pi, por ejemplo, para el análisis de la actividad eléctrica del corazón, entre otras aplicaciones a desarrollar.

Agradecimientos. Los autores agradecen al Centro de Investigación e Innovación en Ciencias de la Computación y Tecnología Educativa (CIICCTE) adscrito a la Facultad de Informática de la UAQ por el espacio brindado para la realización de este trabajo.

Referencias

1. OMS: Enfermedades cardiovasculares. Organización Mundial de la Salud (2021). https://www.who.int/es/health-topics/cardiovascular-diseases (2021).
2. UNAM. Enfermedades del corazón, pandemia permanente. Boletín UNAM de la Dirección de Comunicación Social. https://www.dgc (2020)
3. Schultz, W.M., Kelli, H.M., Lisko, J.C., Varghese, T., Shen, J., Sandesara, P., Sperling, L.S. Socioeconomic Status and Cardiovascular Outcomes: Challenges and Interventions. Circulation, vol. 137, no. 20, pp. 2166–2178, (2018). DOI: 10.1161/CIRCULATIONAHA. 117.029652.
4. Huang, W.: Research on user Satisfaction of Older Community Care based on Structure Equation. In: 2012 Fourth International Symposium on Information Science and Engineering, pp. 489–492 (2012). DOI: 10.1109/ISISE.2012.118.
5. Marschollek, M., Gietzelt, M., Schulze, M., Kohlmann, M., Song, B., Wolf, K.H.: Wearable Sensors in Healthcare and Sensor-Enhanced Health Information Systems: All our Tomorrows?. Healthcare Informatics Research, vol. 18, no. 2, pp. 97–104 (2012). DOI: 10.4258/hir.2012. 18.2.97.
6. Durairaj, M., Revathi, V.: Prediction of Heart Disease using Back Propagation MLP Algorithm. International Journal of Scientific & Technology Research, vol. 4, no. 8, pp. 235–239 (2015).
7. Mohan, S., Thirumalai, C., Srivastava, G.: Effective Heart Disease Prediction using Hybrid Machine Learning Techniques. IEEE Access. 7, pp. 81542–81554 (2019). DOI: 10.1109/ ACCESS.2019.2923707.
8. Dwivedi, A.K.: Performance Evaluation of Different Machine Learning Techniques for Prediction of Heart Disease. Neural Computing and Applications, vol. 29, pp. 685–693 (2018). DOI: 10.1007/s00521-016-2604-1.
9. Bonato, P.: Keynote: Digital Health Technologies and their Role in the Development of Precision Rehabilitation Interventions. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 200–200 (2021). DOI: 10.1109/PerComWorkshops51409.2021.9431126.
10. Aparicio-Montelongo, I., Celaya-Padilla, J.M., Luna-García, H., Galván-Tejada, C.E., Galván-Tejada, J.I., Rosales, H.G.: Predicción de enfermedades cardíacas derivadas de diabetes, mediante algoritmos genéticos: Caso de estudio. Research in Computing Science, vol. 151, no. 6, pp. 159–172 (2022).

11. Documentación de Raspberry Pi: https://www.raspberrypi.com/documentation/ (2024)
12. Plaza, E.: Tendencias en Inteligencia Artificial: Hacia la cuarta década. Nuevas tendencias en Inteligencia Artificial, pp. 379–425 (1992)
13. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press (2021)
14. Deisenroth, M.P., Faisal, A.A., Ong, C.S.: Mathematics for machine learning. Cambridge University Press (2020)
15. Universidad de California Irvine (UCI): Heart Disease Dataset. https://archive.ics.uci.edu/ml/datasets/heart+disease (2024)

ISSN 1870-4069

Transmisión de datos con Xbee Pro de impedancia eléctrica obtenidos con Arduino

Yessica Joselin Palacios-Mogica, Melesio Reyes-Pérez,
María Guadalupe Jiménez-Serrano, Juan Prado-Olivarez,
Javier Díaz-Carmona, José Alfredo Padilla-Medina,
Mauricio Saavedra-Magueyal, Alejandro Israel Barranco-Gutiérrez

Tecnológico Nacional de México en Celaya,
México

{israel.barranco,m2303054,m2303055,
m2303056}@itcelaya.edu.mx

Resumen. La principal materia prima de la inteligencia artificial y redes neuronales son los datos que debe procesar. Este artículo pone a prueba el sistema de comunicación inalámbrica ZigBee con el apoyo de Arduino Mega. Con el fin de cuantificar las ventajas y desventajas del uso de esta tecnología muy poco utilizada en general. Esta tecnología se aplicó para transmitir la impedancia medida por un microcontrolador Arduino Uno, siendo el Xbee el único encargado de la comunicación inalámbrica. El sistema fue sometido a diferentes experimentos con distintas condiciones para revisar su alcance real y sus posibles limitaciones. Por ejemplo, se utilizaron diferentes entornos de prueba con y sin obstáculos a diferentes distancias. Esto permitió observar experimentalmente sus ventajas e inconvenientes. Se observó que la mejor velocidad de transmisión de este protocolo es de 115200 bps, debido a que al utilizar diferentes frecuencias la comunicación tendía a fallar, en forma de ruido, falsos positivos, o información cortada.

Palabras clave: Comunicación inalámbrica, impedancia, zigbee, xbee.

Data Transmission with Xbee Pro of Electrical Impedance Obtained with Arduino

Abstract. The main raw material of artificial intelligence and neural networks is the data it must process. This article evaluates the ZigBee wireless communication system with the support of Arduino Mega. In order to quantify the advantages and disadvantages of using this technology, truly little is used in general. This technology was applied to transmit the impedance measured by an Arduino Uno microcontroller, being the Xbee the only one in charge of the wireless communication. The system was subjected to different experiments with different conditions to review its actual range and possible limitations. For example, different test environments with and without obstacles at different distances were used. This allowed us to experimentally observe its advantages and disadvantages. It was observed that the best transmission speed of this

protocol is 115200 bps, because when using different frequencies, the communication tended to fail, in the form of noise, false positives, or cut information.

Keywords: Wireless communication, impedance, ZigBee, XBee.

1. Introducción

La Inteligencia Artificial (IA) se basa principalmente en datos o información para funcionar de manera efectiva. Muchos de los algoritmos de IA requieren grandes cantidades de datos para aprender patrones, hacer predicciones o tomar decisiones. Los datos se utilizan para entrenar modelos de IA a través de técnicas de aprendizaje automático y aprendizaje profundo. Cuanta más cantidad y calidad de datos se proporcionen a un algoritmo de IA, generalmente mejor será su rendimiento [1].

Es importante destacar que la calidad de los datos y la forma en que se recopilan, procesan, transmiten y utilizan también son aspectos cruciales en el desarrollo de sistemas de IA efectivos [2]. Por lo que la gestión adecuada de los datos es fundamental en el desarrollo y la implementación de soluciones de IA [3].

Por otra parte, la comunicación inalámbrica ha incrementado su relevancia en los últimos años debido a la gran cantidad de sensores y actuadores electrónicos que utilizamos los seres humanos para la vida diaria [4]. Desde aplicaciones médicas, industriales y en el hogar [5, 6]. Esto debido a que permite la transmisión de datos y energía sin la necesidad de cables, lo que puede simplificar la instalación y reducir costos en aplicaciones donde los cables son inconvenientes o imprácticos [7,8,9].

En medicina, la transmisión inalámbrica de impedancia eléctrica se utiliza en técnicas como la tomografía por impedancia eléctrica (TIE), que permite construir imágenes en tiempo real de la distribución de la conductividad eléctrica en tejidos biológicos del cuerpo humano [10]. Esto es útil para la monitorización de la función pulmonar, la detección de cáncer y/o la monitorización de la deshidratación.

También en el ámbito industrial, la medición de la impedancia eléctrica inalámbrica puede ser útil para el monitoreo y control de procesos, como la detección de niveles de líquidos en tanques, la calidad de los cables utilizados en coches y electrodomésticos, la monitorización de la calidad del suelo en la agricultura o la inspección de materiales [11]. En el hogar, la transmisión inalámbrica de impedancia eléctrica puede usarse en dispositivos como cepillos de dientes eléctricos, ritmo cardiaco, oxigenación, nivel de hidratación en sistema de bajo costo [12].

El trabajo presentado en este artículo reporta la adquisición de datos y transmisión inalámbrica de los mismos para que en una etapa posterior del proyecto se construya un sistema de inteligencia artificial (IA) que analice los mismos. Aquí mostramos como Arduino obtiene la impedancia de dispositivos de valores conocidos y las transmite inalámbricamente con una confiabilidad alta en los rangos que se explican posteriormente.

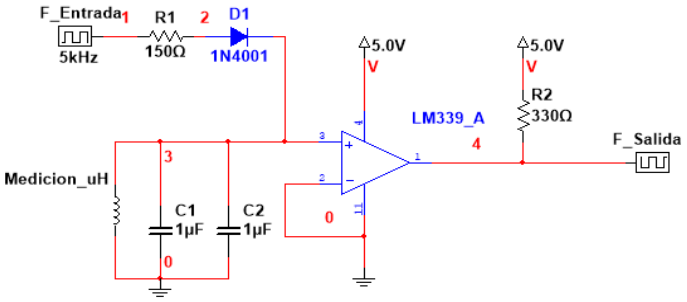


Fig. 1. Diagrama eléctrico que muestra las entradas y salidas del comparador.

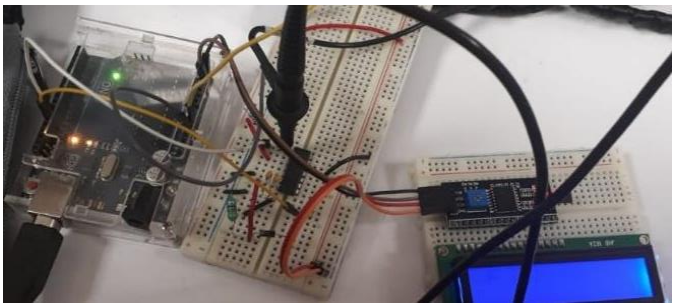


Fig. 2. Circuito armado en Arduino.

2. Materiales y métodos

2.1. Medición de impedancia eléctrica con Arduino

Para medir la impedancia eléctrica, el microcontrolador Arduino inyecta una señal de pulsos cuadrados de 5 volts a 5 KHz al comparador LM339, la salida del comparador crea un pulso cuadrado con periodo relacionado con la impedancia del elemento a medir como se aprecia en la Fig. 1. En la Fig. 2 podemos observar el circuito comparador instalado en el protoboard, la pantalla LCD y su interfaz conectado al Arduino. Estos elementos se enlistan en la tabla 1 para mayor detalle.

Los cálculos mencionados anteriormente se basan en la ecuación (2) que surge de la relación de resonancia de un circuito LC indicado en ecuación (1). La frecuencia en nuestro caso es de 5 KHz y la capacitancia es de 2 microfaradios:

$$F_R = \frac{1}{2\pi\sqrt{LC}}, \tag{1}$$

$$L = \frac{1}{4\pi^2 {F_R}^2 C}, \tag{2}$$

En la fig.3 se puede observar la zona donde se realizaron los experimentos de comunicación, el lugar donde se realizaron fue en el instituto tecnológico de Celaya. El

Tabla 1. Lista de elementos utilizados para construir el circuito medidor de impedancia.

Placa de desarrollo	Arduino Uno
Circuito integrado	Comparador LM339
Capacitor no polarizado	microfaradios
Resistores	150 ohmios y 330 ohmios
Diodo semiconductor	1N4001
Pantalla LCD	I2C
XBee	2 XBee pro

Fig. 3. Lugar de experimentos en vista satelital, en amarillo la distancia entre el receptor y transmisor de los experimentos.

primer experimento se realizó en la cancha de fútbol principal desde la entrada hasta la zona de árboles. El segundo experimento se realizó en el edificio de salones llamados 10 y el último experimento se realizó entre los salones del edificio en forma de H (diagonal al segundo experimento).

2.2 Comunicación ZigBee (Módulo XBee)

La comunicación ZigBee tiene varios beneficios como: alcance, confiabilidad, configuraciones de red, enlaces dedicados. En comparación con otros protocolos de comunicación como WIFI, por lo que es conveniente utilizar esta tecnología. También compatibilidad de su protocolo serial con Arduino proporciona facilidad de uso ya que se presenta con una interfaz simple y con sencillez de integración a múltiples proyectos. Respecto al consumo energético se considera que cuenta con bajo consumo de potencia

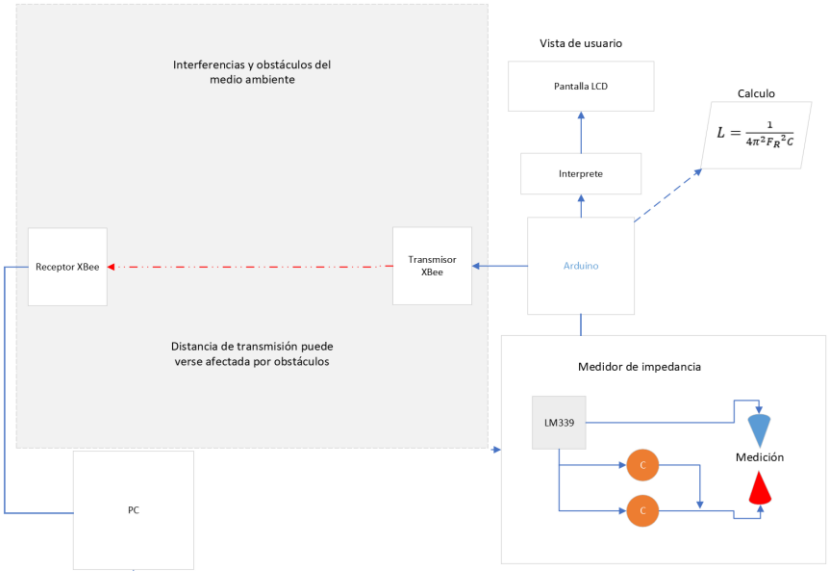


Fig. 4. Diagrama del circuito de transmisión de la medición de impedancia.

eléctrica, alrededor de 6.73 mW [13]. Otro aspecto realmente positivo que considerar es la fiabilidad de los datos transmitidos [9, 14]. En el diagrama de la Fig. 2 podemos observar como el circuito está integrado por los módulos de comunicación XBee para establecer la comunicación inalámbrica, el medio ambiente, el medidor de impedancia, la pantalla LCD y el microcontrolador central que conecta a todos los elementos.

En la Fig. 4 ilustra como dos electrodos miden la impedancia de un inductor o un capacitor y como el Arduino uno inyecta una señal oscilatoria cuadrada al amplificador operacional con la finalidad de medir los tiempos entre la señal de entrada y la señal de salida con base en esta información en el Arduino se calcula la impedancia y estos datos son transmitidos inalámbricamente utilizando los Xbee pro para visualizarlos a distancia en una computadora conectada al Xbee receptor. Para verificar que la información sea correcta se colocó una pantalla LCD a Arduino uno.

En la Fig.5 se muestran los tres tipos de experimentos que se realizaron para poner a prueba la comunicación con los Xbee. En el primer caso tenemos al receptor y al transmisor con línea de vista directa, es decir sin obstáculos entre ellos. En el segundo caso con obstáculos entre transmisor y receptor.

3. Resultados

En la tabla 2, podemos observar diferentes experimentos realizados con los módulos Xbee pro. Se cambiaron las condiciones de los experimentos para observar la calidad de la comunicación.

En la última columna se tiene la evaluación de la comunicación, para esto se utilizó la transmisión de un bit el cual encendía un led, donde se comprobó que por medio de la comunicación a través del teléfono celular de persona a persona era verídico; es decir,

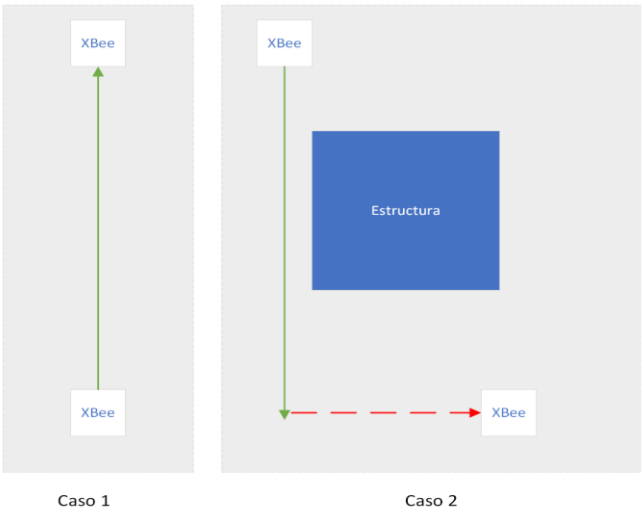


Fig. 5. Diagrama del circuito de transmisión de la medición de impedancia.

cuando el transmisor informaba que había enviado un uno lógico digital el receptor se encargaba de confirmar que coincidiera la información con un led conectado al Xbee pro, si coincide la información se consideraba la calidad de comunicación buena, de lo contrario se consideraba mala.

4. Discusión

Un aspecto peculiar de este proyecto es la medición rápida de la impedancia. Debido a que la medición se realiza cíclicamente en microsegundos la comunicación forzosamente debe darse en los mismos tiempos, es decir, la comunicación debe ser muy rápida porque sucedió que cuando apenas se estaba enviando un dato por los xbee, el Arduino ya tenía varias mediciones en espera de enviarse. Esto impedía el buen funcionamiento del sistema. Por lo que tuvimos que utilizar la velocidad mayor en el experimento. En distancias cortas menores a 90 metros podemos garantizar que el sistema completo funciona correctamente a pesar de que existan obstáculos como edificios, personas y árboles. Respecto a la medición de impedancias, solamente se probó con inductores en un rango de 1 micro Henrio hasta 1 mili Henrio.

5. Conclusiones

Se obtuvo un sistema inalámbrico de medición de impedancia eléctrica funcional, probado con inductores a diferentes inductancias y se transmitieron los datos a diferentes distancias con y sin obstáculos. Probando así la eficiencia del protocolo de comunicación Zigbee en condiciones de baja densidad de datos. Se recomienda aplicar para velocidades en el orden de los milisegundos hasta los microsegundos. Se halló que la velocidad de generación de datos es alta y por tanto la velocidad de transmisión de

Tabla 2. Comparativa de calidad de la comunicación inalámbrica con Xbee pro en diferentes condiciones.

Velocidad de comunicación	Distancia entre módulos Xbee	Obstáculos	Calidad de la comunicación
115200 bps	hasta 90 metros	sin obstáculos	Buena
115200 bps	hasta 50 metros	sin obstáculos	Buena
115200 bps	hasta 15 metros	sin obstáculos	Buena
115200 bps	hasta 10 metros	con obstáculos	Buena
57600 bps	No Aplica	No aplica	Mala

datos también lo debe ser. En un futuro próximo este trabajo se utilizará para hacer mediciones de bioimpedancia para detectar cáncer de mama.

Referencias

1. Lázaro-Mata, D., Morales-Viscaya, J.A., Peralta-Lopez, J.E., Gomez-Cortes, J.C., Padilla-Medina, J.A., Martínez-Nolasco, J.J., Barranco-Gutiérrez, A.I.: Neural Network for Improve ORB-SLAM2 on XZ plane. In: 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, pp. 0380–0386 (2023). DOI: 10.1109/UEMCON59035.2023.10316049.
2. Ponce-Cruz, P.: Inteligencia artificial con aplicaciones a la ingeniería, Alfaomega Grupo Editor, S.A. de C.V., México (2010)
3. Habib, F., Shirazi, S.H., Aurangzeb, K., Khan, A., Bhushan, B., Alhussein, M.: Deep Neural Networks for Enhanced Security: Detecting Metamorphic Malware in IoT Devices. IEEE, vol. 12, pp. 48570–48582 (2024). DOI: 10.1109/ACCESS.2024.3383831.
4. Singandhupe, A., Manh La, H., Feil-Seifer, D.: Reliable Security Algorithm for Drones Using Individual Characteristics from an EEG Signal, IEEE Access, Vol. 6 (2018). DOI: 10.1109/ACCESS.2018.2827362.
5. Halder, T., Roy, B.: Design of a Weather Adoptive and Traffic Adoptive Wireless LED Street Light Control Scheme. In: 2024 IEEE 3rd International Conference on Control, Instrumentation, Energy & Communication, pp. 186–191 (2024). DOI: 10.1109/CIEC 59440.2024.10468073.
6. Boikanyo, K., Zungeru, A.M., Sigweni, B., Yahya, A., Lebekwe, C.: Remote Patient Monitoring Systems: Applications, Architecture, and Challenges. Scientific African, Vol. 20, pp. e01638 (2023). DOI: 10.1016/j.sciaf.2023.e01638.
7. Méndez-Gurrola, I.I., Ramírez-Reyes, A., Barranco-Gutiérrez, A.I: A Review and Perspective on the Main Machine Learning Methods Applied to Physical Sciences. Acta Polytechnica Hungarica, vol. 19, no. 10, pp. 205–220 (2022)
8. Morales-Viscaya, J.A., Alonso-Ramírez, A.A., Castro-Liera, M.A., Gómez-Cortés, J.C., Lazaro-Mata, D., Peralta-López, J.E., Barranco-Gutiérrez, A.I.: Fuzzy Model Parameter and Structure Optimization Using Analytic, Numerical and Heuristic Approaches. Symmetry, vol. 15, no. 7, pp. 1417 (2023). DOI: 10.3390/sym15071417.
9. Khan, M.A., Jha, A.K.: Comparative Analysis of ZigBee and Xbee Wireless Sensor Networks. In: 2016 International Conference on Computing, Communication and– Automation (ICCCA), pp. 996–1000 (2016)
10. Gutiérrez-Lopez, M., Prado-Olivarez, J., Matheus-Troconis, C., Padilla-Medina, A., Barranco-Gutiérrez, A.I., Espinosa-Calderon, A., Díaz-Carmona, J.: A Case Study in Breast Density Evaluation Using Bioimpedance Measurements. Sensors, vol. 22, no. 7, pp. 2747 (2022). DOI: 10.3390/s22072747.

11. Chandra, S., Mahto, A.: Performance Analysis of ZigBee PRO and XBee PRO S2B Radios for Different Topologies in Wireless Sensor Networks. In: 2014 International Conference on Computing for Sustainable Global Development, (INDIACom), pp. 120–125 (2014)
12. Chou, J.C., Chen, J.T., Liao, Y.H., Lai, C.H., Chen, R.T., Tsai, Y.L., Chou, H.T.: Wireless Sensing System for Flexible Arrayed Potentiometric Sensor based on XBEE Module. IEEE Sensors Journal, vol. 16, no. 14, pp. 5588–5595 (2016). DOI: 10.1109/JSEN.2016.2570285.
13. Horvat, G., Sostaric, D., Žagar, D.: Response Surface Methodology based Power Consumption and RF Propagation Analysis and Optimization on XBee WSN Module. Telecommunication Systems, vol. 59, pp. 437–452 (2015). DOI: 10.1007/s11235-014-9904-5.
14. Calvo, I., Abrahams, S., Barambones, O., Chouza, A., Velasco, J., Sáez de Ocári, I., Quesada, J.: A Comparison of Wired and Wireless Technologies for Control Applications. Actas de las XXXIX Jornadas de Automática, Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura, pp. 538–545 (2018). DOI: 10.17979/spudc.978849749 7565.0538.

ISSN 1870-4069

Efecto del triptófano sobre la cinemática de los espermatozoides de cerdo: Análisis de la dinámica de los agrupamientos de las trayectorias, utilizando los descriptores de Fourier

Eder Alejandro Rodríguez-Martínez[1], Cindy Ursula Rivas-Arzaluz[2], Andrés Aragón-Martínez[2]

[1] Universidad Autónoma de Baja California,
Facultad de Ingeniería,
México

[2] Universidad Nacional Autónoma de México,
Facultad de Estudios Superiores Iztacala,
México

armandres@gmail.com

Resumen. La identificación de subpoblaciones cinemáticas espermáticas es esencial en biología reproductiva, particularmente cuando se evalúa la movilidad espermática mediante un sistema CASA. Tradicionalmente, la identificación de las subpoblaciones se realiza con estadística multivariada utilizando como entrada los parámetros de movilidad espermática. A partir de las coordenadas, es posible reconstruir las trayectorias de los espermatozoides. Estas reconstrucciones pueden ser utilizadas como insumo para algoritmos de clasificación. Sin embargo, este enfoque presenta desafíos significativos en términos de costos computacionales y omite el análisis de la dimensión temporal. En este trabajo, desarrollamos un método computacional que hace uso de las coordenadas espermáticas para generar descriptores de Fourier, que sirvieron como entrada para el análisis de componentes principales seguido de agrupamiento jerárquico. Las imágenes de las trayectorias se reconstruyeron en cada agrupamiento para verificar la homogeneidad de los grupos. Trabajamos con muestras de semen de cerdo, fueron tratadas con diferentes concentraciones de triptófano y se evaluó la movilidad con un sistema CASA. Se obtuvieron seis subpoblaciones, la subpoblación tres presentó el valor más alto de movilidad en el control. En esta subpoblación el triptófano indujo una disminución de la velocidad curvilínea y aumentó la linealidad de las trayectorias. El efecto del triptófano puede explicarse por la presencia de la enzima triptófano hidroxilasa (TPH), encargada de convertir el triptófano a serotonina. En conclusión, los descriptores de Fourier son una alternativa al uso de la generación de imágenes que sirvan como entrada para algoritmos de agrupamiento, a la vez que los grupos formados resultan homogéneos.

Palabras clave: Subpoblaciones cinemáticas, triptófano, serotonina, trayectorias espermáticas, descriptores de Fourier.

Effect of Tryptophan on the Kinematics of Boar Sperm: Analysis of Clustering Dynamics of Trajectories Using Fourier Descriptors

Abstract. The identification of kinematic subpopulations of sperm is essential in reproductive biology, particularly when evaluating sperm motility using a CASA system. Traditionally, the identification of sub- populations is performed conventionally with multivariate statistics using the values of sperm motility parameters as input. From the obtained coordinates, it is possible to visually reconstruct the sperm trajectories. These reconstructions can be used as input for classification algorithms. However, this approach presents significant challenges in terms of computational costs and omits the analysis of the temporal dimension. In this work, we developed a computational method that uses sperm coordinates to generate Fourier descriptors, which served as input for principal component analysis followed by hierarchical clustering. The trajectory images were reconstructed in each cluster to verify the homogeneity of the groups. We worked with boar semen samples, which were treated with different concentrations of tryptophan and evaluated for motility with a CASA system. Six subpopulations were obtained, and subpopulation three presented the highest motility value in the control. In this subpopulation, tryptophan induced a decrease in curvilinear velocity while increasing the linearity of the trajectories. The effect of tryptophan can be explained based on its metabolism or the presence of the enzyme tryptophan hydroxylase (TPH), which converts tryptophan to serotonin. In conclusion, Fourier descriptors are an alternative to the use of image generation as input for clustering algorithms, while the formed groups are homogeneous.

Keywords: Kinematic subpopulations, tryptophan, serotonin, sperm trajectories, Fourier descriptors.

1. Introducción

Los sistemas automatizados de análisis de la movilidad espermática (CASA), reconocen la cabeza de los espermatozoides en imágenes consecutivas, y mediante rastreo digital registran las coordenadas de cada célula detectada. Con esas coordenadas es posible: 1) calcular distintos parámetros cinemáticos y 2) reconstruir la trayectoria de los espermatozoides (Figura 1). La identificación de subpoblaciones cinemáticas se realiza de manera convencional analizando los valores de los parámetros cinemáticos [1].

Recientemente, nosotros hemos utilizado las coordenadas espermáticas para reconstruir las imágenes de las trayectorias espermáticas individuales, y hemos utilizado esas imágenes como entrada para algoritmos de agrupamiento jerárquico y SCAN, mediante machine learning [1,2]. En los trabajos citados anteriormente describimos seis [1] y siete clusters [2], respectivamente. Aunque el agrupamiento jerárquico agrupa satisfactoriamente las imágenes, tiene algunos inconvenientes como la sensibilidad a la rotación y traslación.

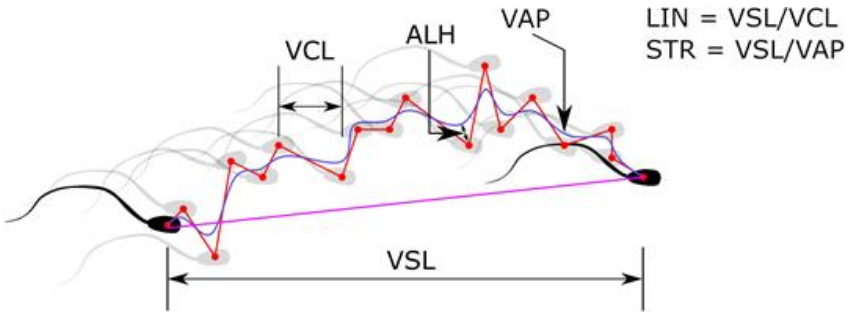


Fig. 1. Esquema de la trayectoria de un espermatozoide, y de algunos de los parámetros cinemáticos que pueden calcularse a partir de las coordenadas. Los puntos rojos son los sitios donde se detectó la cabeza de un espermatozoide en fotogramas consecutivos. El espermatozoide negro a la izquierda indica el punto inicial en el que se detectó al espermatozoide, los espermatozoides grises indican la posición del mismo en fotogramas sucesivos, y el espermatozoide negro a la derecha muestra la posición final del espermatozoide en el segmento de video. La línea roja: la velocidad punto a punto, la línea azul: la velocidad promedio y la línea rosa: la velocidad en línea recta. VCL, velocidad curvilínea; VAP, velocidad promedio; VSL, velocidad en línea recta; ALH, amplitud del desplazamiento lateral de la cabeza con respecto a VAP; LIN y STR son dos índices que dan idea de la linealidad de la trayectoria con respecto a VCL y VAP, respectivamente.

Por su parte, el algoritmo SCAN es robusto a transformaciones de imágenes, por ejemplo, la rotación, la escala y la traslación, entre otras [3]. Esto se debe a que la función de costo es capaz de destruir la información no deseada relativa a las transformaciones de imagen. En consecuencia, los agrupamientos resultantes son más homogéneos que los propuestos por el algoritmo aglomerativo jerárquico. Por su parte, los Descriptores de Fourier (FD) son una representación global de la forma para modelar contornos cerrados utilizando la transformada de Fourier.

Una de las principales ventajas del uso de los descriptores de Fourier es que son fáciles de calcular, simples de normalizar e interpretar [4], además es posible capturar propiedades generales de la forma con solo unos pocos valores numéricos, y el nivel de detalle puede aumentarse o disminuirse, añadiendo o eliminando elementos [5]. Son métodos ampliamente aplicados en varios campos como el análisis de imágenes médicas, bioinformática, biología, gráficos y visión por computadora [6].

Por otra parte, se ha reportado que los espermatozoides expresan proteínas relacionadas con la comunicación serotoninérgica, como receptores, transportadores y proteínas metabolizadoras [7]. Los autores describieron un aumento en las velocidades promedio de los espermatozoides debido a la exposición a la serotonina. En otros estudios se describieron los valores promedio de los parámetros de movilidad de los espermatozoides expuestos a agonistas o antagonistas de los receptores de serotonina [8,9].

Actualmente, se desconoce la estructura de las subpoblaciones cinemáticas de espermatozoides expuestos a sustancias que regulan la comunicación serotoninérgica. Sin embargo, si la serotonina estimula la movilidad de los espermatozoides, entonces el bloqueo de la comunicación serotoninérgica podría cambiar la estructura de las subpoblaciones cinemáticas.

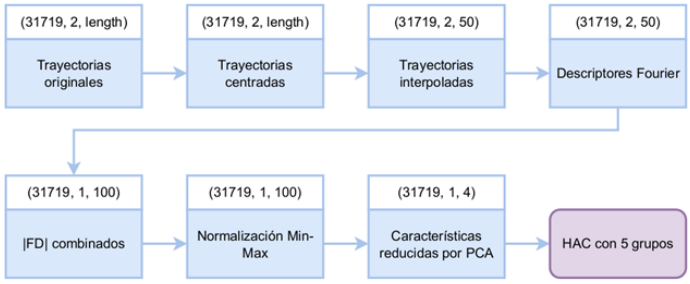


Fig. 2. Representación esquemática del proceso de análisis de trayectorias, ilustra los pasos secuenciales desde los datos crudos de las trayectorias hasta el agrupamiento. El proceso comienza con la fase de pre-procesamiento, donde las trayectorias son centradas a su media e interpoladas de manera uniforme para asegurar consistencia. Las etapas subsecuentes involucran el cálculo de los FDs capturando los patrones geométricos, la normalización de los vectores y la reducción de la dimensionalidad mediante un PCA para retener el 95% de la varianza de los datos. El conjunto final de características reducidas sirvió como entrada para el HAC. Los números y el texto en paréntesis son: Número de elementos, dimensiones de características, número de muestras de características.

Recientemente, nosotros demostramos que, en una subpoblación de espermatozoides, la inhibición del receptor 2 de serotonina aumenta la velocidad y disminuye la linealidad de las trayectorias dependiendo de la duración de la exposición [1]. En Biología reproductiva, la identificación-cuantificación de subpoblaciones espermáticas con trayectorias definidas podrían servir como indicadores de fertilidad masculina. En este trabajo utilizamos la coordenada de espermatozoides expuestos a diferentes concentraciones de triptófano, como entrada para el algoritmo que genera los descriptores de Fourier [10]; mientras que el algoritmo de agrupamiento jerárquico se realizó sobre esos descriptores, y la descripción estadística del efecto del triptófano se realizó utilizando los parámetros cinemáticos asociados a las trayectorias espermáticas.

2. Materiales y métodos

2.1. Muestras espermáticas

Se utilizaron muestras de semen de cerdo obtenidas del Centro de Enseñanza, Investigación y Extensión en Producción Porcina (CEIEPP). Las muestras fueron diluidas con un diluyente comercial y se mantuvieron en alícuotas a 38°C en un baño seco durante 25 minutos antes de su evaluación.

2.2. Evaluación de la movilidad espermática

La movilidad espermática se evaluó en un sistema CASA open-source a los 0 y 30 minutos después de la incubación con diferentes concentraciones de triptófano (10 nM, 100 nM, o 1000 nM). Se colocaron 15 µl de suspensión espermática sobre un portaobjetos, se colocó un cubreobjetos limpio (22×22 mm) y se observó en un microscopio de contraste de fases (B3 CLINLAB, Motic, British Columbia, Canada)

Algorithm 1 Proceso de análisis de trayectorias

Require: Ajuste de las trayectorias $T = \tau_1, \tau_2, \ldots, \tau_n$.
Ensure: Etiqueta de cluster para cada trayectoria.
1: **for all** $\tau \in T$ **do**
2: Centrado τ_i a su media: $\tau_i' \leftarrow \tau_i - \mu(\tau_i)$
3: Interpolación τ_i' a la longitud L: $\tau_i'' \leftarrow$ interpolación (τ_i', L)
4: Computación de FD: $FD_i \leftarrow$ Descriptores de Fourier (τ_i'')
5: Normalizado de los FD: $FD_i \leftarrow$ Normalizado FD_i
6: **end for**
7: FD$\leftarrow$\{FD_1',FD_2', ..., FD_n'\}
8: Ejecución de PCA sobre **FD**, retiene el 95% de la varianza $FD_{PCA} \leftarrow PCA(FD, 0.95)$
9: Ejecución de HAC FD_{PCA}:C$\leftarrow HAC(FD_{PCA}, método{=}Ward)$
10: Determinación del número de clusters *k* a partir del dendograma **C**
11: Asignación de las etiquetas de los clusters con base en k: $L \leftarrow \text{ClustersLabels}(C, k)$
12: **return L**

equipado con una platina térmica a 38°C. Para cada muestra espermática se tomaron cuatro secuencias de imágenes, de distintos campos visuales, a 100X (se evaluaron al menos 500 espermatozoides por muestra). Las secuencias de imágenes fueron capturadas con una cámara Stingray, modelo F 033B (Allied Vision Technologies Inc., Exton, PA, USA) y se almacenaron en una computadora hasta su análisis. Cada secuencia de imágenes se capturó a 60 cuadros por segundo (60 Hz), durante dos segundos, 640 × 480 pixeles, usando el software µManager, versión 1.4 [11].

Las secuencias de video se analizaron con el software ImageJ [12], versión 1.50d y el plugin CASA-RA (modificado para espermatozoides de cerdo) [13,14]. Los parámetros de movilidad analizados para cada espermatozoide fueron: velocidad de la ruta promedio (VAP, µm/sec), velocidad curvilínea (VCL, µm/sec), velocidad rectilínea (VSL, µm/sec), frecuencia de cruce de batido (BCF, Hz), linealidad (LIN, VSL/VCL), rectitud (STR, VSL/VAP), amplitud del desplazamiento lateral de la cabeza (ALH, µm) y bamboleo de la cabeza (WOB, VAP/VCL). Después del análisis, el software generó una hoja de resultados que contenía los valores de los parámetros de movilidad y las coordenadas de cada espermatozoide analizado.

2.3. Construcción de la base de datos

Los algoritmos fueron implementados en un cuaderno de Jupyter (JupyterLab versión 3.3.2) en Anaconda, versión 3.21.5 ejecutando Python, versión 3.8.10 y las librerías utilizadas fueron Scikit-Learn [15], versión 1.0.2; Pandas, versión 1.4.2; Numpy, versión 1.21.5; y Matplotlib, versión 3.6.

2.4. Enfoque para identificar subpoblaciones cinemáticas utilizando los descriptores de Fourier

Este enfoque aprovecha los Descriptores de Fourier (FD) para el análisis de las trayectorias de los espermatozoides, utiliza un método que ofrece una representación integral de los patrones de forma y movilidad.

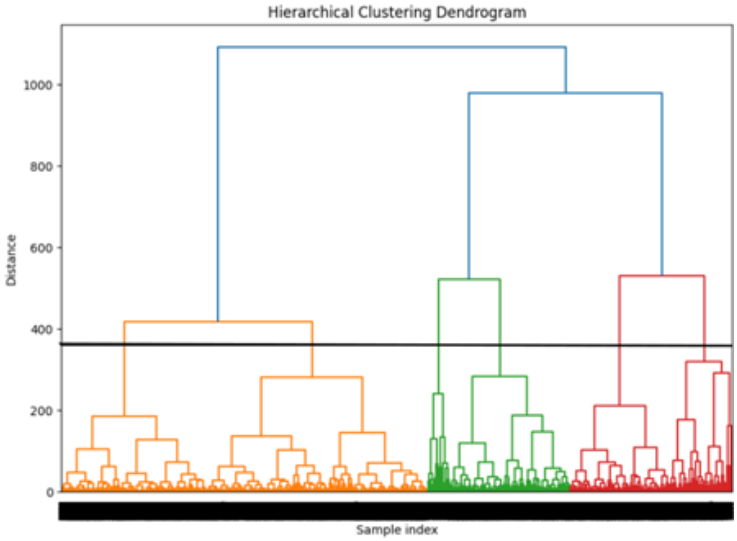


Fig. 3. Dendograma resultante del HAC. Ilustra la estructura jerárquica de los clusters obtenidos. La línea negra horizontal indica el umbral que define el número óptimo de clusters. Este umbral se seleccionó evaluando la distancia vertical más larga que no tenga intersección con las líneas horizontales extendidas, lo que facilita la identificación de los clusters.

A diferencia de los métodos convencionales, que se basan únicamente en los parámetros de movilidad, los FD proporcionan una perspectiva multidimensional al encapsular tanto la frecuencia como la amplitud de los movimientos, ofreciendo así una comprensión matizada de las características de la movilidad [16].

Inicialmente, cada trayectoria espermática es transformada en una secuencia de números, con la subsecuente aplicación de la Transformada Discreta de Fourier (DFT) para calcular los FDs. Esta transformación convierte los datos de la trayectoria espacial a dominios de frecuencia, los cuales son capaces de capturar la forma y el movimiento inherentes a la movilidad espermática [17].

Para abordar la alta dimensionalidad de los FD y enfocarse en los componentes más significativos, se llevó a cabo un Análisis de Componentes Principales (PCA). La selección de los componentes principales (PCs) sigue el criterio de Kaiser, asegurando que solo se retengan los componentes con eigenvalues mayores que uno, capturando así la mayoría de la varianza del conjunto de datos con menos dimensiones [18].

Posterior a la reducción de la dimensionalidad, se aplicó el algoritmo de agrupamiento aglomerativo jerárquico (HAC) a los PCs, utilizando el criterio de Ward para la unión de los clusters o agrupamientos. El criterio de Ward opera bajo el principio de minimización de la varianza, asegurando que la unión de cualquier par de clusters no aumente significativamente la varianza total dentro del cluster, manteniendo así la homogeneidad dentro de las subpoblaciones [19].

Las etapas de procesamiento seguidas hasta obtener los clusters, se muestran en la Figura 2. Para determinar el número de clusters, inicialmente se utilizaron dos métodos cuantitativos: el método del codo (elbow method) y el método de la silueta (silhouette

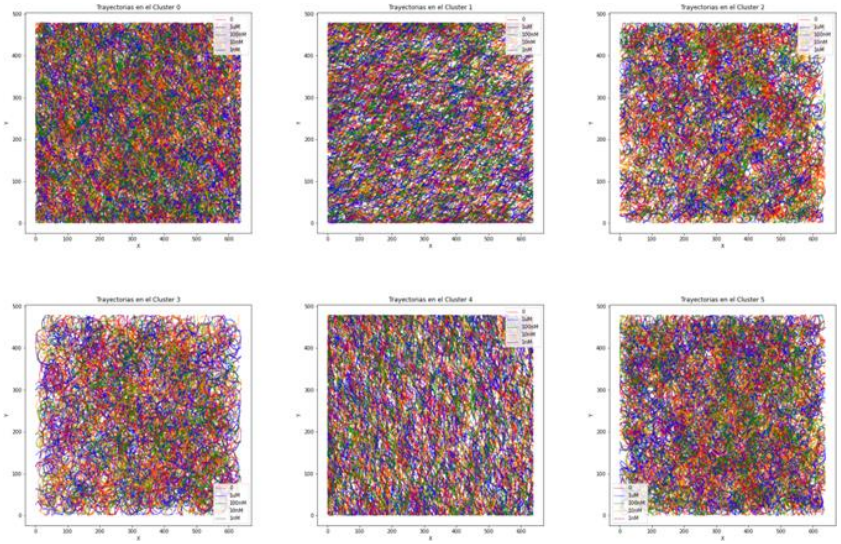


Fig. 4. Gráficos Pollock de los agrupamientos de las imágenes de las trayectorias espermáticas individuales, cada trayectoria fue mapeada de acuerdo con los tratamientos de triptófano. El número de trayectorias en cada gráfico "Pollock" corresponden al 80% del total de las trayectorias en cada subpoblación. Las trayectorias de las subpoblaciones 2, 3 y 5 presentaron trayectorias curvas; sin embargo, las trayectorias de la subpoblación 3 fueron más largas. Las imágenes de las trayectorias cortas y lineales se encontraron en la subpoblación 1 y 4.

method); sin embargo, las imágenes de las trayectorias en los clusters sugeridos por los métodos mencionados, fueron muy heterogéneas.

Por lo tanto, se realizó un análisis visual del dendograma, y se priorizó un buen balance entre la variación de los clusters, el número de clusters y la homogeneidad de las trayectorias dentro de cada cluster. Este refinamiento metodológico ofrece un marco sólido para la delimitación de subpoblaciones espermáticas, proporcionando información sobre la heterogeneidad de los patrones de movilidad espermática con posibles implicaciones para comprender la biología reproductiva [20].

2.5. Preprocesamiento

Se construyeron dos bases de datos (dataframe) con ayuda del plugin CASA- RA [1], en el software ImageJ versión 1.53 [12]. Un dataframe contiene las coordenadas de los espermatozoides representadas por secuencias de (x, y); el segundo dataframe contiene los parámetros de movilidad de cada espermatozoide analizado. Cada trayectoria $\tau \in T$ se centró en su posición media para neutralizar la influencia de la posición original. Para una trayectoria τ que consiste de coordenadas (x_i, y_i) donde $i = 1, 2, \ldots, n$, la posición media (μ_x, μ_y) fue calculada como sigue:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i\,, \quad \mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i\,, \tag{1}$$

donde n es el número de puntos en la trayectoria. Nuestra base de datos contiene $|T| = 31719$ trayectorias con un promedio $\overline{\tau} = 1/T| = \sum|n| \approx 76.42$ puntos en cada trayectoria con una desviación estándar:

$$\sigma = \sqrt{(\Sigma\,(\tau - \bar{\tau})^2/(|T|) - 1)} \approx 33.245\,. \tag{2}$$

La trayectoria centrada $\tau = (x',\ y')$ fue restando la posición media a cada punto:

$$x_i' = x_i - \mu_x, \qquad y_i' = y_i - \mu_y\,. \tag{3}$$

Para estandarizar el número de puntos en todas las trayectorias, se aplicó una interpolación lineal. Para cada trayectoria centrada $\tau = (x_i',\ y_i')$ generamos un conjunto de puntos $\tau'' = \left(x_j'',\ y_j''\right)$,llamado trayectoria interpolada, donde $j = 1, 2, \ldots, m$ (con $m = 50$ en nuestro caso) usando funciones de interpolación lineal f_x y f_y:

$$f_x(t) = x_i' + \frac{x_{i+1}' - x_i'}{t_{i+1} - t_i}(t - t_i), f_y(t) = y_i' + \frac{y_{i+1}' - y_i'}{t_{i+1} - t_i}(t - t_i)\,. \tag{4}$$

2.6. Extracción de características usando los descriptores de Fourier

Los Descriptores de Fourier [6] son calculados para cada trayectoria interpolada para capturar sus componentes de frecuencia y representar la forma de su trayectoria. Los FDs son obtenidos aplicando la Transformada Discreta de Fourier (DFT) [21] – específicamente la Transformada Rápida de Fourier (DFT) [22] – a las coordenadas x y y de manera separada:

$$FD_x(k) = \sum_{n=0}^{L-1} x\,(n) e^{\frac{-2pii}{L}kn}, \qquad FD_y(k) = \sum_{n=0}^{L-1} y\,(n) e^{\frac{-2pii}{L}kn}\,, \tag{5}$$

donde $L = 50$ es la longitud de la trayectoria interpolada, y k es el índice del componente de frecuencia. La magnitud de cada FD, |FD|, es usada para representar la trayectoria, combinando los descriptores de ambas dimensiones en un solo vector de características:

$$FD_{\text{combined}} = \left[|FD_x(0)|, \ldots, |FD_x(L-1)|, \left|FD_y(0)\right|, \ldots, \left|FD_y(L-1)\right|\right]. \tag{6}$$

2.7. Normalización y reducción de la dimensionalidad

Los vectores FD concatenados fueron normalizados al rango [0,1] para asegurar una contribución uniforme en todas las características. Los datos normalizados se utilizaron como entrada para el análisis de componentes principales (PCA) [23]. Este paso reduce la complejidad computacional y enfoca el análisis en las características significativas.

Tabla 1. Efecto del triptófano sobre los valores promedio de los parámetros de movilidad espermática al tiempo 0 y 30, en la subpoblación 3. (media + S.E).

Tiempo (min)	Triptófano (nM)	Parámetros cinemáticos								
		VCL (um/s)	VAP (um/s)	VSL (um/s)	LIN (%)	STR (%)	WOB (%)	BCF (Hz)	ALH (um)	n
0	0	117.13+47.67	62.62+26.58	47.79+23.38	0.42+0.15	0.77+1.17	0.54+0.13	35.75+6.52	4.43+1.84	541
	10	99.44+34.97*	55.82+20.78*	44.03+19.32*	0.45+0 14*	0.79+0.16	0.56+0.11*	36.16+6.31	3.89+1.42*	646
	100	101.40+36.37*	53.90+20.87*	43.71+18.75*	0.43+0.11	0.81+0.13*	0.53+0.11	37.08+5.69*	3.95+1.48*	564
	1000	111.46+44.72	59.86+23.16	44.63+19.69	0.42+0.15	0.76+0.17	0.54+0.12	36.22+6.44	4.27+1.68	545
30	0	100.21+40.62	57.25+26.58	47.55+20.68	0.49+0.16	0.84+0.15	0.58+0.14	36.62+6.69	3.83+1.65	540
	10	93.37+28.48*	56.13+21.53	46.74+20.23	0.5+0.14	0.83+0.14	0.59+0.12	36.01+5.98	3.68+1.25	604
	100	99.30+30.36	58.32+22.21	48.68+20.55	0.49+0.14	0.83+0.12	0.58+0.12	35.75+6.51	3.91+1.30	944
	1000	91.69+27.34*	52.99+21.48*	45.81+20.40	0.49+0.14	0.86+0.11	0.57+0.13	36.43+6.57	3.57+1.25*	875

2.8. Agrupamiento jerárquico aglomerativo

Los componentes principales obtenidos en la etapa previa, sirvieron como entrada para el algoritmo de agrupamiento jerárquico aglomerativo (HAC).

El HAC se ejecutó utilizando el criterio de enlace de Ward, el cual minimiza la varianza dentro de los clusters. El proceso no requiere la pre-especificación del número de clusters. En su lugar, se genera un dendograma para visualizar la jerarquía de agrupamiento, a partir del cual se puede elegir un corte apropiado para elegir los clusters finales:

$$D(i,j) = \sqrt{\sum_{k=1}^{m} \left(x_{ik} - x_{jk}\right)^2}, \tag{7}$$

donde $D(i,j)$ es la distancia entre clusters i y j, y m es el número de características en el espacio de características reducido. El número óptimo de clusters se determinó analizando el dendograma y seleccionando un umbral que separe de mejor manera los distintos patrones de movilidad. Se presenta el pseudocódigo del flujo de trabajo en el Algoritmo 1.

3. Resultados

Se obtuvieron los FD a partir de los datos de coordenadas espermáticas. Del análisis de PC se obtuvieron 4 PCs que describieron el 95% de la varianza de los datos. El dendrograma obtenido del PC sugiere la existencia de tres clusters; sin embargo, el agrupamiento de las trayectorias con tres clusters no resultó significativo debido a la heterogeneidad de los agrupamientos. Se decidió entonces utilizar seis grupos para el ajuste de las condiciones del análisis de agrupamiento jerárquico (Figura 3). El número de espermatozoides en las subpoblaciones 0 a 5 en el tiempo 0 de incubación con triptófano fue de 8239, 2731, 982, 2296, 532 y 228, respectivamente; mientras que al tiempo 30 fue de 7500, 3290, 1526, 2963, 945 y 487, respectivamente.

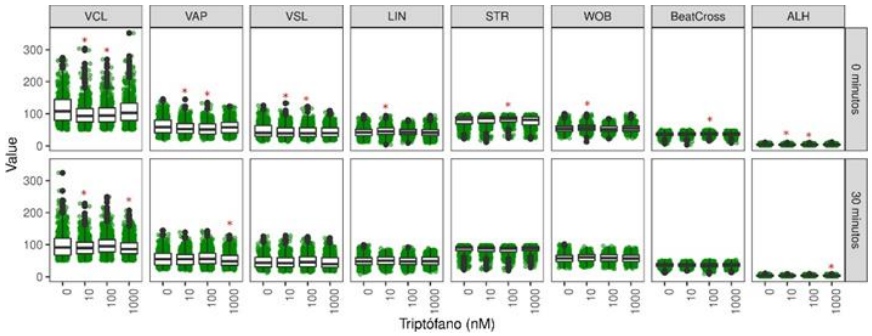


Fig. 5. Efecto del triptófano sobre los parámetros de movilidad en la subpoblación 3 en los dos tiempos de evaluación. Gráficas de cajas con bigote por cada parámetro de movilidad de la subpoblación 3. Las líneas horizontales en cada caja representan el cuartil 2 (mediana), el cuartil 1 y cuartil 3. Los bigotes representan el valor mínimo y máximo. Los puntos verdes corresponden a cada espermatozoide evaluado por concentración de triptófano. Los puntos negros indican valores extremos. *P<0.05 vs Control (0), ANOVA de una vía, seguida de Tuckey.

El agrupamiento de las imágenes en las subpoblaciones 3 corresponden a trayectorias cortas y lineales; mientras que las trayectorias con curvas más cerradas, incluso formando bucles se observaron predominantemente en la subpoblación 5 (Figura 4). La exposición al triptófano indujo una disminución en VCL, VAP y VSL; mientras que los valores de LIN y STR aumentaron en las concentraciones intermedias en el clúster 3 (Figura 5).

Aquí se describe el efecto en el clúster 3 por ser el que presentó la velocidad de VCL más alta en el control; los valores promedio de los parámetros de movilidad espermática en las subpoblaciones 0, 1, 2, 3, 4 y 5 a los tiempos 0 o 30 de incubación se pueden observar en[1].

4. Discusión

En este trabajo diseñamos un modelo computacional, que utiliza las coordenadas del rastreo de la cabeza de los espermatozoides en imágenes sucesivas, para generar descriptores de Fourier y caracterizar las trayectorias espermáticas con base en los parámetros de movilidad espermática. Posteriormente, los descriptores de Fourier se utilizaron en un proceso de reducción de la dimensionalidad y agrupamiento jerárquico. Finalmente, las subpoblaciones identificadas se describieron con base en los descriptores de movilidad espermática, y se utilizó estadística inferencial para identificar el efecto de la exposición de los espermatozoides a triptófano.

Los descriptores de Fourier se han utilizado para describir la geometría de imágenes biológicas [24]. No obstante, de acuerdo con nuestro conocimiento, esta es la primera vez que se utilizan los descriptores de Fourier para describir la forma de las trayectorias espermáticas.

El uso de los descriptores de Fourier tiene ventajas sobre el agrupamiento de imágenes de trayectorias, puesto que es insensible a la rotación, la

[1] github.com/armandres/COMIA_2024_Descriptores_de_Fourier/blob/main/COMIA_2024_Tablas_Parametros_de_movilidad.pdf

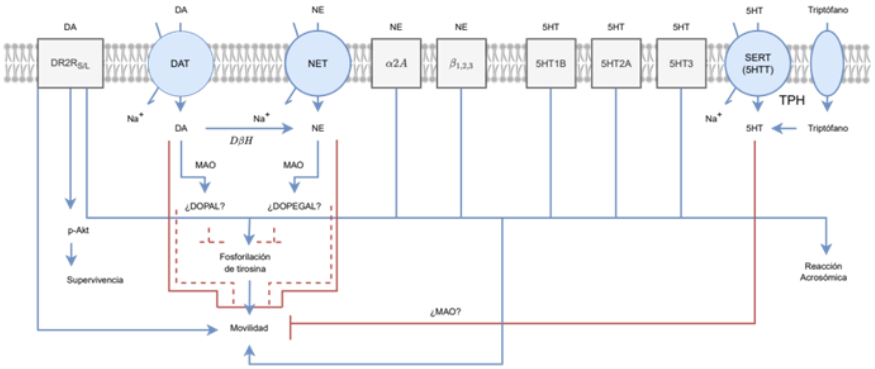


Fig. 6. Descripción de la acción de las monoaminas en la función espermática. DA, dopamina; NE, norepinefrina; 5HT, serotonina; DR2R, receptor 2 de dopamina, DAT, transportador de dopamina; NET, transportador de norepinefrina; α2A, receptor α2 adrenérgico; β1,2,3, receptor βadrenérgico; 5HT1B, 5HT2A y 5HT3, receptores de serotonina; SERT (5HTT), transportador de serotonina; TPH, triptófano hidroxilasa; D βH, dopamina- β-hydroxylasa; MAO, monoamino oxidase; DOPAL, 3,4-dihidroxifenilacetaldehido; DOPEGAL, 3,4-dihidroxifenilglicolaldehido. En este modelo bifásico de la regulación de la funcionalidad espermática, están incluidas dos acciones, los efectos de activación mediados por los receptores de monoaminas (DR2RS/L, α2A, β1, 2, 3, 5HT1B, 5HT2A, 5HT3), y los efectos inhibidores mediados por la recaptura de los transportadores (DAT, NET y SERT) y los derivados oxidados de las catecolaminas producidas por la acción de MAO (Modificado de Ramírez- Reveco et al., 2017).

traslación y el escalado [10]. El uso de los descriptores de Fourier, es computacionalmente menos costoso que el agrupamiento jerárquico sobre imágenes generadas a partir de coordenadas.

Sin embargo, de manera semejante a resultados previos [1], consideramos que tenemos un área de oportunidad en la selección automática del número de clusters, puesto que cuando analizamos los gráficos "Pollock", derivados del agrupamiento jerárquico con ajuste de tres, cinco o seis clusters, de acuerdo con el dendograma obtenido, no se apreciaron grupos homogéneos.

Visualmente, nosotros encontramos una mayor homogeneidad cuando se seleccionó seis clusters como ajuste para el agrupamiento jerárquico. Se ha demostrado que la serotonina se encuentra presente en el semen humano [25], y que puede tener un efecto sobre la fisiología reproductiva femenina [26].

Aunque se desconoce el sitio de síntesis de la serotonina presente en el eyaculado, es claro que tiene un efecto, no solo en el tracto reproductor femenino, sino también sobre los espermatozoides. La comunicación serotoninérgica puede ser exógena o bien, endógena. La síntesis de serotonina endógena inicia con la modificación química del aminoácido triptófano [27] (Figura 6).

Recientemente, nosotros demostramos que la inhibición de la comunicación serotoninérgica exógena, mediante la exposición a la ketanserina (un inhibidor del receptor 2 de serotonina), afecta a los espermatozoides de cerdo en la subpoblación más veloz, induciendo un aumento de la velocidad, a la vez que las trayectorias de los espermatozoides se tornan más curvas [1].

En el presente trabajo observamos que la exposición de los espermatozoides a triptófano, afecta los espermatozoides de la subpoblación que es más veloz en condiciones control, disminuyendo la velocidad curvilínea y tornando más lineales las trayectorias. Así, vemos que la comunicación serotoninérgica tiene efectos opuestos cuando la disrupción ocurre de manera exógena o de manera endógena. En este momento desconocemos la explicación celular y molecular por la cual solamente un subconjunto (subpoblación) de espermatozoides responda de esta manera al triptófano.

Una posible explicación es que la subpoblación responsiva expresa intensamente a la enzima TPH (triptófano hidroxilasa), que es la encargada de convertir el triptófano a serotonina (Figura 6). Otra posibilidad es que el triptófano entre a otra ruta metabólica, distinta a la de la síntesis de serotonina; y que los productos bioquímicos derivados de esa ruta afecten de alguna manera la movilidad. Previamente se reportó que el triptófano puede convertirse a kinurenina [28]; posiblemente esa u otras moléculas puedan participar en él. En conclusión, el método computacional que hace uso de las coordenadas como entrada para generar a los descriptores de Fourier, para que a su vez estos descriptores sirvan de entrada para una estrategia de reducción de la dimensionalidad y de agrupamiento jerárquico, permite obtener subpoblaciones homogéneas de trayectorias espermáticas. De tal manera, que los descriptores de movilidad asociados a las trayectorias espermáticas permitieron caracterizar la estructura cinemática de las subpoblaciones; así como identificar el efecto de la exposición a triptófano sobre la cinemática de la movilidad espermática a nivel subpoblacional. fenómeno de movilidad espermática.

Agradecimientos. Este proyecto se desarrolló gracias al apoyo del programa DGAPA-PAPIIT de la UNAM número IT201021.

Referencias

1. Rodríguez-Martínez, E.A., Rivas, C.U., Ayala, M.E., Blanco-Rodríguez, R., Juarez, N., Hernandez-Vargas, E.A., Aragón, A.: A New Computational Approach, based on Images Trajectories, to Identify the Subjacent Heterogeneity of Sperm to the Effects of Ketanserin. Cytometry Part A, vol. 103, no. 8, pp. 655–663 (2023). DOI: 10.1002/cyto.a.24732.
2. Rodríguez-Martínez, E.A., Rivas-Arzaluz, C.U., Aragón-Martínez, A.: Aplicación del algoritmo SCAN en el agrupamiento de imágenes de trayectorias espermáticas: Identificación de la heterogeneidad de la respuesta espermática a la Ketanserina. Research in Computing Science, vol. 152, no. 8, pp. 253–266 (2023)
3. Van-Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van-Gool, L.: Scan: Learning to Classify Images without Labels. In: Proceedings of the European Conference on Computer Vision, vol. 12355, pp. 268–285 (2020). DOI: 10.1007/978-3-030-58607-2_16.
4. El-ghazal, A., Basir, O., Belkasim, S.: A New Shape Signature for Fourier descriptors. In: 2007 IEEE International Conference on Image Processing, vol. 1, pp. 161–164 (2007). DOI: 10.1109/icip.2007.4378916.
5. Burger, W., Burge, M.J.: Digital Image Processing: An Algorithmic Introduction using Java. Texts in Computer Science, Springer London (2016). DOI: 10.1007/978-1-4471-6684-9.
6. Valizadeh, G., Babapour-Mofrad, F.: A Comprehensive Survey on Two and Three-Dimensional Fourier Shape Descriptors: Biomedical Applications. Archives of

Computational Methods in Engineering, vol. 29, no. 7, pp. 4643–4681 (2022). DOI: 10.1007/s11831-022-09750-7.

7. Jiménez-Trejo, F., Tapia-Rodríguez, M., Cerbón, M., Kuhn, D.M., Manjarrez-Gutiérrez, G., Mendoza-Rodríguez, C.A., Picazo, O.: Evidence of 5-ht Components in Human Sperm: Implications for Protein Tyrosine Phosphorylation and the Physiology of Motility. Reproduction, vol. 144, no. 6, pp. 677–685 (2012). DOI: 10.1530/rep-12-0145.
8. Fujinoki, M.: Serotonin-Enhanced Hyperactivation of Hamster Sperm. Reproduction, vol. 142, no. 2, pp. 255–266 (2011). DOI: 10.1530/rep-11-0074.
9. Sakamoto, C., Fujinoki, M., Kitazawa, M., Obayashi, S.: Serotonergic Signals Enhanced Hamster Sperm Hyperactivation. Journal of Reproduction and Development, vol. 67, no. 4, pp. 241–250 (2021). DOI: 10.1262/jrd.2020-108.
10. Raghavendra, G.S., Danish, M., Khan, S.I., Venkateswarlu, S.C.: Fourier Descriptors for Shape-based Image Retrieval. International Journal of Engineering Research and Technology, vol. 2, no. 4, pp. 857–863 (2013)
11. Edelstein, A.D., Tsuchida, M.A., Amodaj, N., Pinkard, H., Vale, R.D., Stuurman, N.: Advanced Methods of Microscope Control using μ Manager Software. Journal of Biological Methods, vol. 1, no. 2, pp. 1 (2014). DOI: 10.14440/jbm.2014.36.
12. Schneider, C.A., Rasband, W.S., Eliceiri, K.W.: NIH Image to Image: 25 Years of Image Analysis. Nature Methods, vol. 9, no. 7, pp. 671–675 (2012). DOI: 10.1038/nmeth.2089.
13. Giaretta, E., Munerato, M., Yeste, M., Galeati, G., Spinaci, M., Tamanini, C., Mari, G., Bucci, D.: Implementing an Open-Access CASA Software for the Assessment of Stallion Sperm Motility: Relationship with other Sperm Quality Parameters. Animal Reproduction Science, vol. 176, pp. 11–19 (2017). DOI: 10.1016/j.anireprosci.2016.11.003.
14. Rivas, C.U., Ayala, M.E., Aragón, A.: Effect of Various pH Levels on the Sperm Kinematic Parameters of Boars. South African Journal of Animal Science, vol. 52, no. 5, pp. 693–704 (2023). DOI: 10.4314/sajas.v52i5.13.
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830 (2011)
16. Stéphanou, A., Ronot, X., Tracqui, P.: Analysis of Cell Motility Combining Cytomechanical Model Simulations and an Optical flow Method. Mathematics and Biosciences in Interaction, pp. 91–112 (2003). DOI: 10.1007/978-3-0348-8043-5_9.
17. Díaz-Guerrero, D.S., Montoya, F., Hernández, H.O., Hernández-Herrera, P., Darszon, A., Corkidi, G.: Computation of Human-Sperm Local Flagellar Instantaneous Velocity. In: Proceedings of the XLVI Mexican Conference on Biomedical Engineering, International Federation of Medical and Biological Engineering, vol. 96, pp. 59–66 (2023). DOI: 10.1007/978-3-031-46933-6_7.
18. Auerswald, M., Moshagen, M.: How to Determine the Number of Factors to Retain in Exploratory Factor Analysis: A Comparison of Extraction Methods under Realistic Conditions. Psychological Methods, vol. 24, no. 4, pp. 468–491 (2019). DOI: 10.1037/met0000200.
19. Ward, J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association, vol. 58, no. 301, pp. 236–244 (1963). DOI: https://doi.org/10.2307/2282967.
20. Martínez-Pastor, F., Tizado, E.J., Garde, J.J., Anel, L., de-Paz, P.: Statistical Series: Opportunities and Challenges of Sperm Motility Subpopulation Analysis. Theriogenology, vol. 75, no. 5, pp. 783–795 (2011). DOI: 10.1016/j.theriogenology.2010.11.034.
21. An, M., Gertner, I., Rofheart, M., Tolimieri, R.: Discrete Fast Fourier Transorm Algorithms: A Tutorial Survey. Advances in Electronics and Electron Physics, vol. 80, pp. 1–67 (1991). DOI: 10.1016/s0065-2539(08)60607-1.

22. Rajaby, E., Sayedi, S.M.: A Structured Review of Sparse fast Fourier Transform Algorithms. Digital Signal Processing, vol. 123, pp. 103403 (2022). DOI: 10.1016/j.dsp.2022.103403.
23. Shah, J., Sharif, M., Raza, M., Azeem, A.: A Survey: Linear and Nonlinear PCA based Face Recognition Techniques. International Arab Journal of Information Technology, vol. 10, no. 6, pp. 536–545 (2013)
24. Tomakova, R., Komkov, V., Emelianov, E., Tomakov, M.: The use of Fourier Descriptors for the Classification and Analysis of Peripheral Blood Smears Image. Applied Mathematics, vol. 8, no. 11, pp. 1563–1571 (2017). DOI: 10.4236/am.2017.811114.
25. Platek, S.M.: Female Infidelity and Paternal Uncertainty: Evolutionary Perspectives on Male Anti-cuckoldry Tactics. Cambridge University Press, pp. 256 (2006). DOI: https://doi.org/10.1017/CBO9780511617812.
26. Burch, R.L.: Semen and vaginal chemistry. Encyclopedia of Evolutionary Psychological Science, Springer, Cham, pp. 6999–7001 (2021). DOI: 10.1007/978-3-319-19650-3_2008.
27. Ramírez-Reveco, A., Villarroel-Espíndola, F., Rodríguez-Gil, J.E., Concha, I.I.: Neuronal Signaling Repertoire in the Mammalian Sperm Functionality. Biology of Reproduction, vol. 96, no. 3, pp. 505–524 (2017). DOI: 10.1095/biolreprod.116.144154.
28. Oxenkrug, G.F.: Metabolic Syndrome, Age-associated Neuroendocrine Disorders, and Dysregulation of Tryptophan—Kynurenine Metabolism. Annals of the New York Academy of Sciences, vol. 1199, no. 1, pp. 1–14 (2010). DOI: 10.1111/j.1749-6632.2009.05356.x.

ISSN 1870-4069

Mantenimiento predictivo de motores de corriente directa empleando redes neuronales artificiales

Jonathan Villanueva Tavira, Juan González Serna,
Andrés Blanco Ortega, Héctor Buenabad Arias,
Edgardo de Jesús Carrera Avendaño

Tecnológico Nacional de México,
Centro Nacional de Investigación y Desarrollo Tecnológico
México

andres.bo@cenidet.tecnm.mx

Resumen. El Mantenimiento Predictivo es la técnica que utiliza la Inteligencia Artificial con la finalidad de predecir fallas. La principal contribución de este trabajo es la metodología para procesar y utilizar los datos obtenidos a partir de las vibraciones de los motores de corriente directa, ya que son elementos muy empleados en proyectos de robótica móvil. Posteriormente, con los datos almacenados tanto de motores de corriente directa en buen y mal estado, se construye una red neuronal artificial entrenada con un algoritmo de aprendizaje no supervisado, con la finalidad de hacer grupos con los datos recabados de los motores de corriente directa. Para concluir la etapa de experimentación se realizaron experimentos con otros métodos de aprendizaje automático como: árboles de decisión, máquinas de soporte vectorial y k vecinos más cercanos.

Palabras clave: Redes neuronales artificiales, inteligencia artificial, mantenimiento predictivo.

Predictive Maintenance of Direct Current Motors Using Artificial Neural Networks

Abstract. Predictive Maintenance is the technique that utilizes Artificial Intelligence to predict failures. The main contribution of this work is the methodology for processing and using the data obtained from the vibrations of direct current motors, as they are widely used in mobile robotics projects. Subsequently, with the stored data from both good and faulty direct current motors, an artificial neural network is built and trained with an unsupervised learning algorithm to cluster the collected data from the direct current motors. To conclude the experimentation phase, experiments were conducted with other machine learning methods such as decision trees, support vector machines, and k-nearest neighbors.

Keywords: Artificial neural networks, artificial intelligence, predictive maintenance.

1. Introducción

Los ingenieros en la industria constantemente buscan maneras de evitar el mantenimiento correctivo con la finalidad de reducir los costos que implica. El control, seguimiento y mantenimiento de los equipos que se encuentran en una línea de producción son actividades fundamentales para la calidad y desempeño de los procesos productivos [1, 2, 3, 4, 5]. Los sensores y principalmente los motores juegan un papel importante para máquinas como: bandas transportadoras, generadores, mezcladoras, compresores, hornos, soldadoras, entre otras. Para garantizar su óptimo funcionamiento estas deben de estar constantemente monitoreadas y realizar su respectivo mantenimiento [5].

El Mantenimiento Predictivo es la técnica que utiliza la Inteligencia Artificial con la finalidad de predecir fallas. Este enfoque permite a los ingenieros o técnicos encargados del mantenimiento realizar la corrección de los equipos antes de que se averíen, lo que hace que el tiempo de vida útil de los equipos se prolongue.

Este tipo de mantenimiento presenta varias ventajas como: reducción del tiempo de inactividad no planificada, ayuda a la identificación de fallas del equipo mediante el monitoreo, y finalmente, disminuye el tiempo de inactividad al reducir el tiempo para inspeccionar y realizar reparaciones. Sin embargo, la desventaja que presenta este enfoque, es la necesidad de contar con sistemas basados en Internet de las Cosas, por lo que implantarlo en un inicio podría representar un costo alto. Este tipo de mantenimiento no solo permite predecir alguna falla, si no también identificar las partes que pueden estar fallando en sus equipos estimando el tiempo de la falla en la maquinaria (Ver Figura 1) [6].

Los costos que conlleva el mantenimiento en una empresa o industria pueden representar desde el 15% al 60% total del costo del producto fabricado. Por ejemplo, en la industria de los alimentos el costo promedio del mantenimiento asciende al 15% del total del bien producido; en contraste con las industrias del ramo metalmecánica que utilizan en sus productos materiales como: hierro, acero y papel representa un costo del 60% del total de producto fabricado.

La industria estadounidense cada año gasta más de 200 mil millones de dólares en mantenimiento de equipos e instalaciones en las plantas de producción. El resultado de una mala administración al ejecutar un plan de mantenimiento representa una pérdida de más de 60 mil millones de dólares cada año y tiene un impacto directamente sobre el tiempo de producción y la calidad del producto [6].

Se entiende por mantenimiento a un conjunto de acciones o técnicas que permiten prolongar el tiempo de vida útil de un equipo, asegurando el costo mínimo y garantizando la seguridad para el usuario [6].

La importancia de mantenimiento industrial radica en que en la Industria Mexicana es necesario mantener el equipo y la maquinaria funcionando de forma continua y eficientemente. El mantenimiento se puede clasificar de en tres tipos: preventivo, correctivo y predictivo (ver Figura 1).

El pronosticar el tiempo de la falla en una línea de producción o en un equipo puede ayudar a programar de una forma más eficiente el mantenimiento del mismo [7] (ver Figura 2).

Fig. 1. Clasificación de los tipos de Mantenimiento Industrial [7].

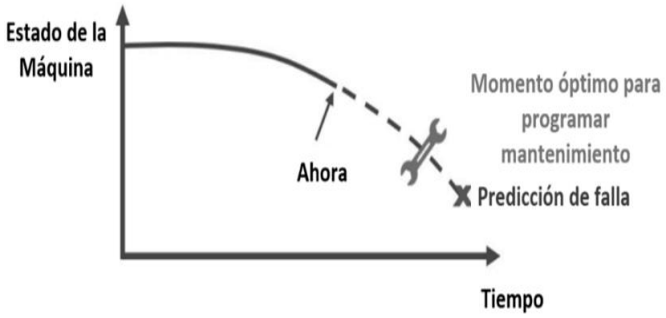


Fig. 2. Descripción gráfica del Mantenimiento Predictivo [7].

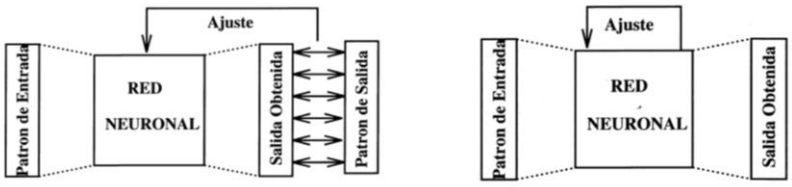


Fig. 3. Diagrama a bloques de los tipos de aprendizaje [8].

2. Red neuronal artificial de Kohonen

Esta arquitectura de Red Neuronal Artificial (RNA) tiene una característica muy particular a diferencia de las demás Redes Neuronales Artificiales. Se trata de una RNA de aprendizaje no supervisado, que son aquellas que no requieren un supervisor externo para realizar su aprendizaje. Esto consiste en que la red descubra por sí sola características o categorías con los datos de entrada y se obtengan de forma codificada a la salida [8] (Ver Figura 3).

El objetivo de esta RNA es categorizar los datos de entrada. Se trata de que los datos que son muy parecidos sean clasificados como pertenecientes a la misma categoría. En estos modelos suele existir una capa de clasificación compuesta por tantas neuronas como categorías puedan existir en los datos. Cada categoría está representada por un

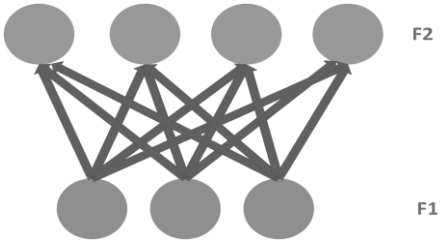


Fig. 4. Arquitectura del aprendizaje competitivo [8].

prototipo cuyas características son una especie perteneciente a un grupo con características similares. En la capa de clasificación, cada neurona corresponde a un prototipo. La arquitectura de una red neuronal artificial con aprendizaje competitivo consiste en dos capas denominadas F1 y F2, donde F1 es la capa de entrada y recibe las señales del entorno. La capa F2 es la que se encarga de producir la salida [8] (Ver Figura 4).

El científico filandés, Teuvo Kohonen, diseñó un modelo adaptable a las características de los sistemas neuronales, consiste en una red neuronal de dos capas, una primera capa de entrada y una segunda de competencia. La capa de entrada recibe la señal de entrada, su dimensión depende de los atributos de entrada. Cada neurona de entrada está conectada a todas las células de la capa de competición. Este algoritmo no concluye después de presentarle una vez todos los patrones de entrada, ya que este proceso debe de repetirse varias veces para conseguir que la red neuronal pueda realizar una clasificación más exacta. [9, 10]. A continuación, se describe el algoritmo de la red neuronal de Kohonen:

a) En primer lugar, se inicializan los pesos wij con valores aleatorios menores a uno y se fija la zona inicial de la vecindad entre las neuronas de salida (Ver Figura 5).
b) De tal forma que la matriz de pesos quedaría de la siguiente forma, con un radio R=0 y una tasa de aprendizaje del $\alpha = 0.6$.

$$\begin{bmatrix} 0.2 & 0.6 & 0.4 & 0.4 & 0.2 \\ 0.3 & 0.5 & 0.7 & 0.6 & 0.8 \end{bmatrix}.$$

c) Se debe de calcular la distancia euclídea para saber cuál neurona se parece más a los datos de entrada. Para este paso se realiza mediante la siguiente expresión:

$$D(j) = \sum (w_{ij} - x_i)^2 . \quad (1)$$

d) Posteriormente, se presenta a la red la información de entrada (los patrones a reconocer) en forma de vector (0.3,0.4). Y se calcula la distancia euclídea para ver cuál de las neuronas es más parecida a los datos de entrada.

$$D(1) = (0.2 - 0.3)^2 + (0.3 - 0.4)^2 = 0.02 ,$$

$$D(2) = (0.6 - 0.3)^2 + (0.5 - 0.4)^2 = 0.10 ,$$

$$D(3) = (0.4 - 0.3)^2 + (0.7 - 0.4)^2 = 0.10 ,$$

$$D(4) = (0.4 - 0.3)^2 + (0.6 - 0.4)^2 = 0.50 ,$$

$$D(5) = (0.4 - 0.3)^2 + (0.6 - 0.4)^2 = 0.17 .$$

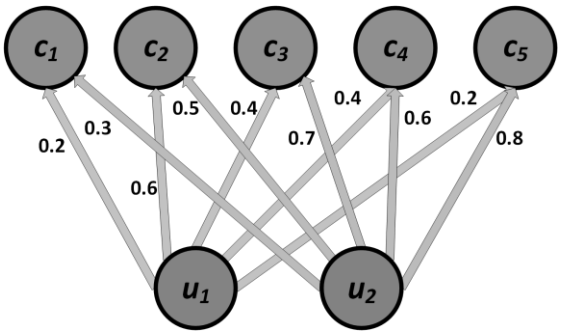


Fig. 5. Topología de la Red Neuronal Artificial de Kohonen [9].

e) Como se observa la primera neurona en este caso, es la que más se parece a los datos de entrada, por lo que se deben de actualizar los pesos de ella y de sus vecinas. Para la actualización de pesos se realiza a partir de la siguiente expresión:

$$w_{ij(new)} = w_{ij(old)} + \propto (x - w_{ij(old)}), \tag{1}$$

$$\begin{aligned} w_{11(new)} &= w_{11(old)} + 0.3(0.3 - 0.2), \\ w_{11(new)} &= 0.2 + 0.3(0.3 - 0.2), \\ w_{11(new)} &= 0.23, \end{aligned} \tag{2}$$

$$\begin{aligned} w_{21(old)} &+ \propto \left(x - w_{21(old)}\right), \\ w_{21(new)} &= 0.3 + 0.3(0.4 - 0.3), \\ w_{21(new)} &= 0.33. \end{aligned} \tag{3}$$

f) Una vez que se actualizaron los pesos, se deberán de sustituir esos pesos en la matriz de pesos de la red antes de la actualización de las vecinas:

$$\begin{bmatrix} 0.2 & 0.6 & 0.4 & 0.4 & 0.2 \\ 0.3 & 0.5 & 0.7 & 0.6 & 0.8 \end{bmatrix}, \qquad \begin{bmatrix} 0.23 & 0.6 & 0.4 & 0.4 & 0.2 \\ 0.33 & 0.5 & 0.7 & 0.6 & 0.8 \end{bmatrix}.$$

Debido a la actualización de los pesos, se deben de actualizar también las neuronas vecinas a la neurona ganadora. Para ello, también aplicamos la actualización de los pesos a la neurona dos:

$$\begin{aligned} w_{12(new)} &= w_{12(old)} + \propto \left(x - w_{12(old)}\right), \\ \mathrm{w}_{12(\mathrm{new})} &= 0.6 + 0.3(0.3 - 0.6), \\ w_{12(new)} &= 0.51, \end{aligned} \tag{4}$$

$$\begin{aligned} w_{22(new)} &= w_{22(old)} + \propto \left(x - w_{22(old)}\right), \\ \mathrm{w}_{22(\mathrm{new})} &= 0.5 + 0.3(0.4 - 0.5), \\ w_{22(new)} &= 0.47. \end{aligned} \tag{5}$$

Una vez que se actualizaron los pesos, se deberán de sustituir esos pesos en la matriz de pesos de la red. Este proceso deberá de repetirse hasta que los pesos se estabilicen, algunos autores proponen que el entrenamiento debe de realizarse en 300 épocas para lograr unos pesos estables [9, 10]:

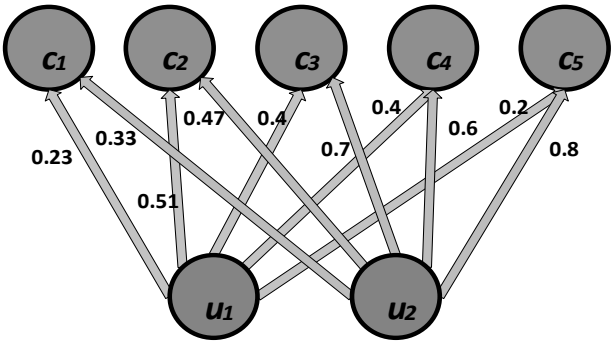


Fig. 6. Red Neuronal Artificial de Kohonen con sus pesos [10].

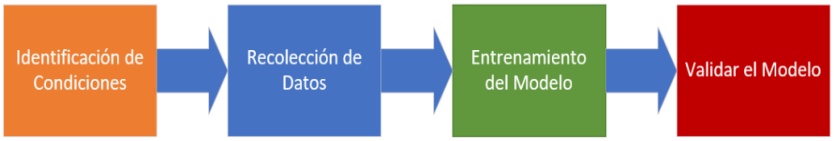


Fig. 7. Descripción gráfica de la metodología de solución [6].

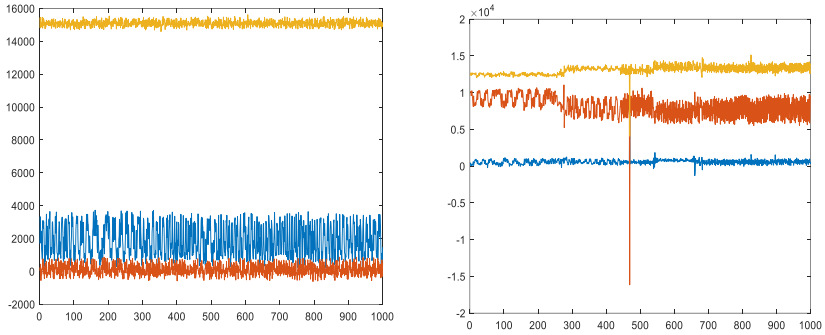
$$\begin{bmatrix} 0.23 & 0.6 & 0.4 & 0.4 & 0.2 \\ 0.33 & 0.5 & 0.7 & 0.6 & 0.8 \end{bmatrix}, \qquad \begin{bmatrix} 0.23 & 0.51 & 0.4 & 0.4 & 0.2 \\ 0.33 & 0.47 & 0.7 & 0.6 & 0.8 \end{bmatrix}.$$

En la figura 6 se observa la topología de la red neuronal artificial con sus pesos al finalizar la primera parte del entrenamiento [10].

3. Metodología de solución

La Inteligencia artificial en conjunto con el Internet de las Cosas (IoT) han permitido potencializar este último tipo de mantenimiento en la industria. En la actualidad es más sencillo obtener los datos de los equipos industriales mediante las herramientas del IoT. Estos datos, combinados con técnicas de Inteligencia Artificial como lo son las Redes Neuronales Artificiales se emplean para construir los modelos de predicción de fallas [11]. A continuación, se detallan los pasos empleados en esta metodología de solución (ver Figura 7).

a) **Identificación de las Condiciones.** Para realizar los experimentos se toman como parámetros las vibraciones de los motores de corriente directa. Las vibraciones mecánicas están relacionadas con el mantenimiento predictivo, ya que proporcionan indicadores de fallos presentes o posibles a mediano plazo. Las vibraciones pueden causar aumento de esfuerzos y tensiones, deformaciones superiores al margen elástico, fatiga en los materiales, desgaste y pérdida de energía y mayor consumo energético. La razón por la que se eligió la variable vibración es porque el estudio de las mismas permite detectar un estado general del



a) Motor en buen estado b) Motor en mal estado

Fig. 8. Datos generados por el motor de corriente directa en buen y mal estado obtenidos por el giroscopio.

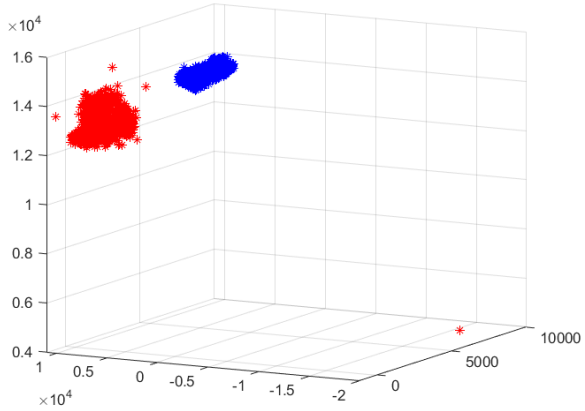


Fig. 9. Datos generados por el motor en mal estado obtenidos por el giroscopio vistos en 3D.

funcionamiento de las máquinas, así como el desequilibrio, desalineamientos, defectos en rodamientos y cojinetes, fallos de lubricación, entre otros [6].

b) Recolección de los datos. En lo que concierne a esta segunda etapa, mediante una rutina de adquisición de datos provenientes de un giroscopio y un acelerómetro, estos se colocaron directamente en los motores con la finalidad de poder recolectar la información correspondiente para cada motor. En este caso, se emplearon dos motores de corriente directa de las mismas características, uno en buen estado y uno desgastado con la finalidad de contrastar las lecturas provenientes de los sensores. Esta información recopilada, conforma el Dataset para experimentar con los datos obtenidos por los sensores. En la figura 8 se muestran los 2000 datos tomados en un periodo de 1 segundo cada uno de los motores. Las gráficas resultantes muestras una variación de las medidas proporcionadas por el sensor de acuerdo al estado del motor.

Adicionalmente, en la figura 9 se muestran estos mismos datos graficados en tres dimensiones con la finalidad de tener otra perspectiva de los datos generados por los sensores de las vibraciones de los motores de corriente directa.

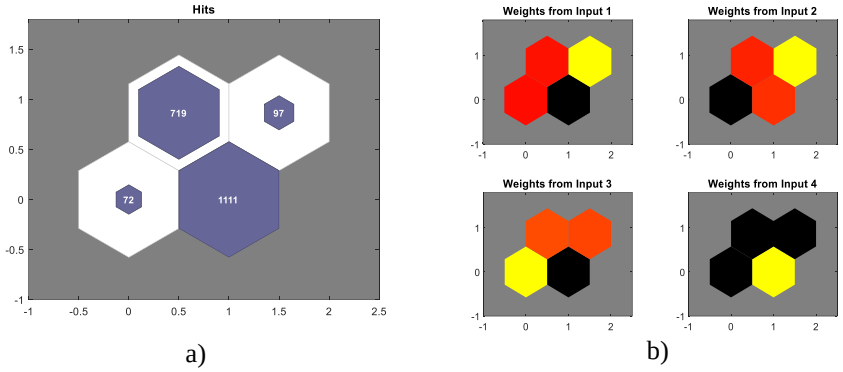


Fig. 10. a) Plano de Conjuntos, b) Plano de Pesos de la Red Neuronal Artificial.

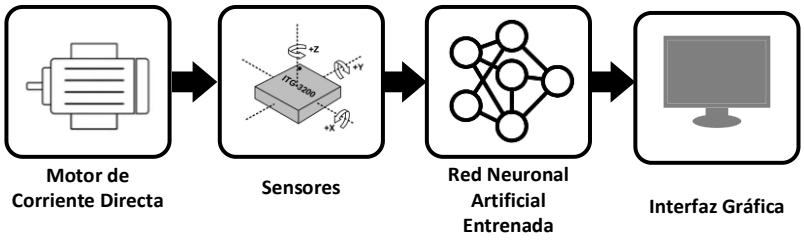


Fig. 11. Diagrama a bloques de la metodología de la solución.

c) Entrenamiento del Modelo. En esta etapa, se realiza el entrenamiento de la red neuronal a partir de los datos con la finalidad de agrupar en distintos clusters las lecturas provenientes de los sensores tanto del motor en buen estado, como en mal estado. En la figura 10 se observa las ubicaciones de las neuronas en la topología de la red neuronal artificial e indica la cantidad de observaciones que están asociadas para cada una de las neuronas. La topología seleccionada es de 2x2, por lo que existen 4 neuronas en total en la red.

d) Validación del Modelo. Para esta etapa, con la finalidad de comprobar el funcionamiento de la red neuronal artificial, se realiza la exportación de la misma a una función de tipo MATLAB para poder emplearla en una interfaz. La finalidad de esta etapa consiste en monitorear el estado del motor con los sensores que nos emiten las vibraciones y estas mismas sirven como entrada a la Red Neuronal Artificial desplegando el resultado de la red en una interfaz gráfica (Ver Figura 11).

4. Comparación con otra técnica de Red Neuronal Artificial

En esta sección se presentan las pruebas realizadas, así como los resultados obtenidos después de aplicar el algoritmo de entrenamiento para cada uno de los conjuntos de datos. Adicionalmente, se realiza la comparación empleando una red neuronal artificial de aprendizaje supervisado con el algoritmo de retropropagación con la finalidad de contar con dos algoritmos de apoyo para predecir alguna posible

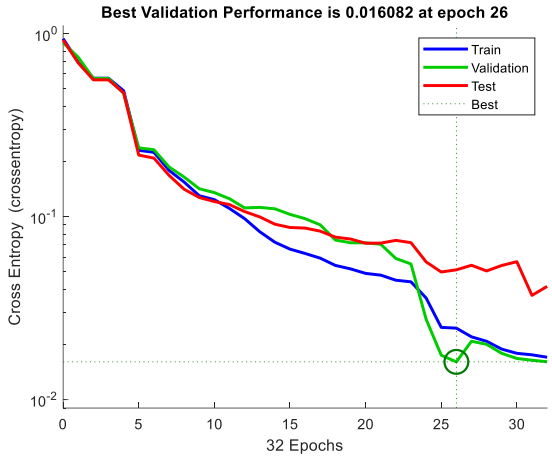


Fig.12. Gráfica del error de la red neuronal artificial.

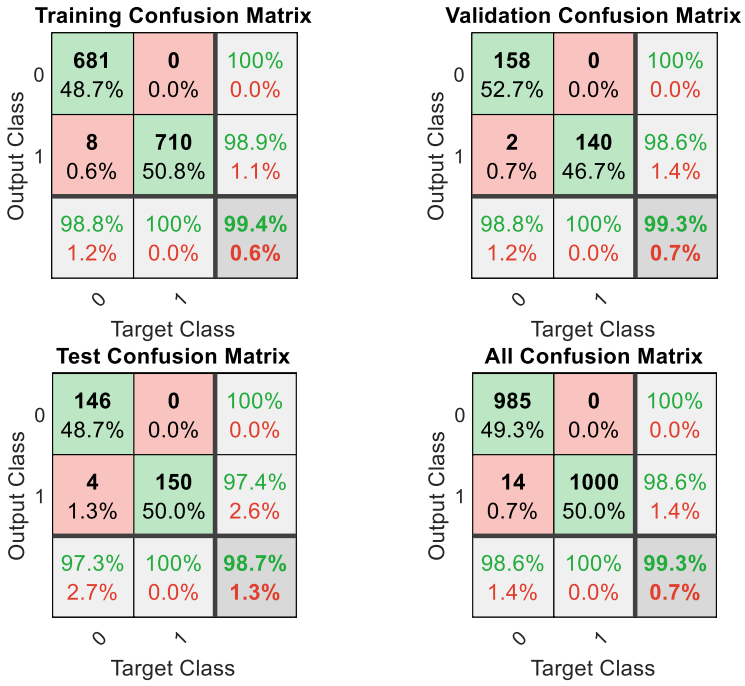


Fig. 13. Matriz de Confusión de la Red Neuronal Artificial.

descompostura de los motores, detectando en tiempo y forma vibraciones anormales en los parámetros de su funcionamiento. Para ello se emplea una red neuronal multicapa entrenada con el algoritmo backpropagation. (Ver Figura 12)

En la figura 13 se muestra la Matriz de Confusión que despliega los datos de la red neuronal al artificial con el porcentaje de clasificación correcta y errónea. En lo que

Tabla 1. Comparación con otros métodos de clasificación.

Algoritmo	Precisión
Árbol de Decisión	99.8 %
Máquina de Soporte Vectorial	96.5 %
KNN	99.9 %
Máquina de Soporte Vectorial Cúbica	99.4 %

respecta a esta técnica empleada para realizar la clasificación de las vibraciones de motores de corriente directa en bueno y mal estado. Esto permite hacer nuestro sistema más robusto, ya que las dos técnicas se aplicarán en paralelo con la finalidad de detectar las vibraciones anormales en el funcionamiento de un motor en buen estado.

Los resultados obtenidos en el entrenamiento de la red neuronal artificial muestran que estos modelos logran el objetivo, ya que se implementaron en un sistema en tiempo real para realizar el monitoreo de los motores de corriente directa y detectar alguna falla asociada a la vibración de los motores. Es importante mencionar, que las redes neuronales artificiales presentan la desventaja que son vistas como cajas negras y no pueden explicar la forma en que llegaron a un resultado. Por lo anteriormente, esbozado se realiza la comparación con otras técnicas de clasificación como árboles de decisión y máquinas de soporte vectorial, obteniendo los siguientes resultados (Ver Tabla 1).

5. Conclusiones

Cada día más las empresas buscan formas más eficientes de aplicar técnicas que apoyen en la toma de decisiones en la línea de producción prediciendo cuándo puede presentarse alguna falla en algún equipo o proceso reduciendo el costo y la carga de trabajo. Los objetivos de este trabajo y sus principales contribuciones son proponer una metodología para utilizar los datos de las vibraciones de los motores de corriente directa y con ellos, construir y entrenar dos modelos de redes neuronales artificiales capaces de predecir cuándo puede ocurrir una falla. En primera instancia se implementa una red neuronal artificial de aprendizaje no supervisado y en segunda se emplea un modelo de aprendizaje supervisado. Finalmente, el segundo modelo se compara con otras técnicas de clasificación obteniendo un nivel alto de precisión.

Referencias

1. Patan, K., Korbicz, J., Głowacki, G.: DC Motor Fault Diagnosis by Means of Artificial Neural Networks. In: Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics, Angers, France, pp. 11–18 (2007). DOI: 10.5220/0001625400110018.
2. Gongora, W.S., Silva, H.V.D., Goedtel, A., Godoy, W.F., da Silva, S.A.O.: Neural Approach for Bearing Fault Detection in Three Phase Induction Motors. In: Proceedings of the 2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED), Valencia, Spain, 27–30, pp. 566–572 (2013). DOI: 10.1109/DEMPED.2013.6645771.

3. Kouki, M., Dellagi, S., Achour, Z., Erray, W.: Optimal Integrated Maintenance Policy based on Quality Deterioration. In: Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, Malaysia, pp. 9–12, pp. 838–842 (2014). DOI: 10.1109/IEEM.2014.7058756.
4. Amihai, I., Gitzel, R., Kotriwala, A.M., Pareschi, D., Subbiah, S., Sosale, G.: An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health. In: Proceedings of the 2018 IEEE 20th Conference on Business Informatics (CBI), pp. 11–14, pp. 178–185 (2018). DOI: 10.1109/CBI.2018.00028.
5. Scalabrini-Sampaio, G., Vallim-Filho, A.R.A., Santos da Silva, L., Augusto da Silva, L.: Prediction of Motor Failure Time Using an Artificial Neural Network. Sensors, vol. 19, no. 19, pp. 4342 (2019). DOI: 10.3390/s19194342.
6. Mobley, R.K.: An Introduction to Predictive Maintenance, Elsevier Science (2002)
7. Introduction to Maintenance with MATLAB.
8. Isasi Viñuela, P., Gaván Leon, I.M.: Redes de Neuronas Artificiales: Un enfoque práctico. Pearson Prentice Hall (2004)
9. Hernando, J.R.: Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones. Ra-Ma (1994)
10. Sivanandam, S.N.: Introduction to Neural Networks usign MATLAB 6.0. Mc Graw Hill Education (2006)
11. Abbasi, J.A.: Predictive Maintenance in Industrial Machinery using Machine Learning. Lulea University of Technology Department of Computer Science, Electrical and Space Engineering (2021)

ISSN 1870-4069

Clasificación del infarto de miocardio en mujeres

Ricardo Daniel Lozano Sánchez[1], María Dolores Torres Soto[1], Aurora Torres Soto[1], Yoselin Esparza Monreal[2], Cinthya Judith López Ramírez[2], Esperanza Sánchez Alemán[3]

[1] Benemérita Universidad Autónoma de Aguascalientes, Departamento de Sistemas de Información, México

[2] Centenario Hospital Miguel Hidalgo, Cardiología Clínica, Aguascalientes, México

[3] Benemérita Universidad Autónoma de Aguascalientes, Departamento de Morfología, Centro de Ciencias Básicas, México.

al236294@edu.uaa.mx, yosmonreal992@hotmail.com, dracinthyaramirezchmh@gmail.com, {mdtorres, atorres, espesanchez}@correo.uaa.mx

Resumen. El infarto de miocardio es la principal causa de muertes en el mundo, siendo los mayores índices de mortalidad en mujeres de edades avanzadas. Proponemos el uso de dos algoritmos de aprendizaje automático para entrenar un modelo de clasificación que utiliza registros clínicos de mujeres con infarto para clasificar mujeres con alto riesgo de mortalidad y mejorar su esperanza de vida. Descubrimos que los datos clínicos de las mujeres pueden permitirnos crear un modelo basado en el aprendizaje automático para predecir la mortalidad en el 90 % de las pacientes que sufrieron de un infarto. Usando Create ML de Apple, entrenamos modelos basados en bosque aleatorio y máquina de soporte vectorial utilizando 105 registros de mujeres hospitalizadas del Instituto de Cardiología de Faisalabad y el Hospital Aliado en Faisalabad (Punjab, Pakistán). Nuestros resultados muestran que bosque aleatorio supera a máquina de soporte vectorial con una precisión del 90 % en comparación con el 85 % de precisión del modelo de Support Vector Machine. Esta herramienta muestra la capacidad de este tipo de algoritmos para crear modelos que permitan a los médicos proporcionar la atención necesaria a las mujeres que están en riesgo de muerte.

Palabras clave: Infarto de miocardio en mujeres, aprendizaje automático, create ML.

Myocardial Infarction Classification in Women

Abstract. Myocardial infarction is the leading cause of death in the world, being the highest mortality rates in older women. We propose the use of two machine learning algorithms to train a classification model that uses clinical records of women with heart attack to classify women at high risk of mortality and improve

their life expectancy. We discovered that women's clinical data can allow us to create a model based on machine learning to predict mortality in 90% of patients who suffered from a heart attack. Using Apple's Create ML, we train models based on Random Forest and Support Vector Machine using 105 records of hospitalized women from the Faisalabad Institute of Cardiology and the Allied Hospital in Faisalabad (Punjab, Pakistan). Our results show that random forest surpasses the vector support machine with an accuracy of 90% compared to the 85% accuracy of the Support Vector Machine model. This tool shows the ability of this type of algorithm to create models that allow doctors to provide the necessary care to women who are at risk of death.

Keywords: Myocardial infarction in women, machine learning, create ML.

1. Introducción

Según la Organización Mundial de la Salud (OMS), el infarto de miocardio (IM) es una de las principales causas de muerte en todo el mundo, afectando a 9 millones de personas en el 2019, representando el 16% de todas las muertes a nivel mundial [1]. El MI afecta a casi 2.7 millones de mujeres en los Estados Unidos, con una mayor tasa de prevalencia de MI en mujeres mayores y de poblaciones minoritarias [2]. Las mujeres tienden a experimentar síntomas atípicos del IM y a menudo se les diagnostica mal o se diagnostica demasiado tarde, debido a las diferencias fisiológicas en el corazón de una mujer que pueden plantear un desafío para los médicos [2].

En México, durante la pandemia de COVID-19, la principal causa de muerte en las mujeres fue por enfermedades relacionadas con el corazón, superando las 97,000 muertes por enfermedades cardiovasculares (ECV) en comparación con las 70,000 muertes causadas por el síndrome respiratorio agudo coronavirus (SARS-CoV-2) [3], tendencia que continúo al año siguiente, de enero a julio de 2021 [4].

Históricamente, el IM en las mujeres ha sido poco tratado y poco estudiado, lo que ha llevado a un diagnóstico erróneo y a un tratamiento inadecuado [5]. Además, las mujeres tienen menos probabilidades que los hombres de recibir tratamiento debido a los retrasos en el reconocimiento del infarto agudo de miocardio (IAM) por parte de los proveedores de atención médica o los pacientes [6].

Un estudio encontró que el 18% de los pacientes con IM sin elevación del ST recibieron otro diagnóstico inicial no relacionado con el corazón, y las mujeres con IM con elevación del ST tenían un 59% más de probabilidades de un diagnóstico erróneo inicial en comparación con los hombres [7-8]. Una atención pobre y retrasada pone a las mujeres en un alto riesgo de mortalidad [9].

Estas condiciones están relacionadas con el desconocimiento de la enfermedad, razones socioculturales y financieras que pueden resultar en retrasos prehospitalarios y menores tasas de acceso a la atención [10]. El conjunto de datos seleccionado contiene 105 registros de mujeres que sufrieron de un IM, estos registros abarcan de 40 a 95 años. Fue publicado por Davide Chicco, Giuseppe Jurman, contiene datos de 299 pacientes del Instituto de Cardiología de Faisalabad y del Hospital Aliado de Faisalabad (Punjab, Pakistán) durante abril a diciembre del 2015.

Sus resultados mostraron que la creatinina sérica y la fracción de eyección pueden determinar la mortalidad de un paciente diagnosticado con IAM, encontrando el

modelo de bosque aleatorio (RF) [11] con una precisión del 74% en la validación de su modelo [12]. Los autores proponen que únicamente la creatinina sérica y la fracción de eyección pueden predecir la supervivencia de los pacientes [12].

Nosotros utilizamos todas las características contenidas en el conjunto de datos, esto debido a que obtuvimos una mejor puntuación en la validación del modelo. Para la creación de la herramienta clasificadora utilizamos (RF) [11] y una máquina de soporte vectorial (SVM) [13] utilizando las herramientas Create ML de Apple [14].

El objetivo de este artículo es desarrollar una herramienta precisa de clasificación del riesgo de mortalidad en las mujeres que han experimentado un IM utilizando datos clínicos reales y lograr una tasa de precisión superior al 80%. Con la finalidad contribuir con una herramienta valiosa en el campo de la atención médica, reduciendo las tasas de mortalidad hospitalaria y mejorar la esperanza de vida de los pacientes con infarto de miocardio.

Profundizamos en el desarrollo y la viabilidad de un clasificador diseñado para distinguir a las mujeres que han sufrido un IM y evaluar su riesgo de mortalidad. Este documento contiene una sección de "marco teórico", aquí discutiremos algunas enfermedades relacionadas con el corazón, la enfermedad por MI, algunos factores de riesgo y críticos en las mujeres. Después de eso, en la sección "Material y método" discutiremos en profundidad el conjunto de datos elegido, los algoritmos utilizados y cómo se llevarán a cabo los experimentos, en la sección "Resultados" discutiremos cualquier hallazgo de la fase de experimentación.

2. Marco teórico

Históricamente, las enfermedades relacionadas con el corazón a menudo se han percibido como principalmente hombres de afecto, sin embargo, un cuerpo significativo de investigación ha encontrado que el impacto de las enfermedades vasculares cerebrovasculares, como la enfermedad arterial coronaria, la insuficiencia cardíaca y el accidente cerebrovascular, son la principal causa de mortalidad en las mujeres a nivel mundial [15-17].

2.1. Infarto de miocardio

Un infarto de miocardio es una afección en la que el tejido cardíaco sufre una falta de suministro de sangre (isquemia), lo que causa su muerte (necrosis); cuando una parte del tejido cardíaco tiene un infarto, puede producir complicaciones graves para la salud de alguien, incluido el riesgo de muerte. Los síntomas comunes incluyen dolor en el pecho, dificultad para respirar, dolor en el brazo o el cuello izquierdo, entre otros. La causa de un infarto de miocardio podría deberse a varios factores, como: la acumulación de placa, un paño o la vasoconstricción de las arterias coronarias del corazón, lo que resulta en que el músculo deje de funcionar, causando la muerte del tejido [18-19]:

1. **Disparidades de Género en las ECV.** A lo largo de los años, estudios han revelado profundas disparidades de género en la manifestación, el diagnóstico y el tratamiento del IM en las mujeres, a menudo exhiben síntomas únicos que difieren del clásico dolor torácico en hombres y en su lugar, pueden presentarse con signos sutiles como dificultad para respirar, náuseas, fatiga, síntomas gastrointestinales,

de ansiedad o emocionales [10,20]. Esta divergencia de síntomas ha llevado a un mal tratamiento, diagnósticos equivocados y un retraso en la atención [5,21].

2. **Factores de Riesgo y Causas Subyacentes del IAM.** Las ECV en mujeres tienen factores de riesgo adicionales a los que son comunes en los hombres, como sufrir complicaciones durante el embarazo, fluctuaciones hormonales, estrés crónico, entre otros [20]. Comprender esos factores de riesgo emergentes es esencial para una evaluación precisa del riesgo y permitir desarrollar estrategias de prevención específicas para las mujeres [10].
3. **Cambios Hormonales.** Las hormonas, específicamente los estrógenos contribuyen significativamente a la salud de las mujeres, siendo objeto de investigación sobre la salud del corazón en las mujeres. Los estrógenos tienen efectos cardiovasculares protectores, como el mantenimiento de niveles saludables de colesterol en sangre. Siendo las mujeres que tienen niveles más bajos de estrógenos, específicamente durante la menopausia, más propensas a las ECV, incluido el IAM [2,15].

2.2. Inteligencia artificial

La IA es la simulación de la inteligencia humana en las máquinas, capaz de aprender, resolver problemas y pensar como los humanos. La IA encierra una amplia gama de tecnologías y técnicas aplicables a problemas y tareas del mundo real, dirigidas específicamente a tareas que normalmente requieren inteligencia humana, tales tareas requieren la comprensión y la capacidad del lenguaje natural, el reconocimiento de imágenes, la resolución de problemas complejos y la toma de decisiones [22]:

1. **Aprendizaje Automático.** Es una técnica utilizada por la inteligencia artificial (IA) que se centra en el desarrollo de algoritmos que permiten a las computadoras a aprender, hacer predicciones y clasificaciones basadas en datos. Los algoritmos de aprendizaje automático utilizan técnicas estadísticas para aprender patrones de grandes conjuntos de datos. Los algoritmos de aprendizaje automático se pueden clasificar en tres tipos: Aprendizaje supervisado: el modelo se entrena con datos etiquetados. Aprendizaje no supervisado: identifica patrones y estructuras, como la agrupación. y Aprendizaje por refuerzo: aprende a tomar decisiones interactuando con el entorno, recibe retroalimentación en forma de recompensas o sanciones [23].
2. **Algoritmo de Bosque Aleatorio.** Es un algoritmo de tipo supervisado utilizado para tareas de regresión y clasificación, pertenece a los métodos de aprendizaje conjunto. Combina predicciones de múltiples algoritmos de árboles de decisión para hacer una predicción más precisa y robusta que cualquier modelo individual. También tiene una característica clave para seleccionar cada característica para los nodos de cada árbol y asegura una predicción diversa y robusta [11] Actualmente se utiliza en la salud, las finanzas, la ecología y la clasificación de imágenes [24- 26].
3. **Algoritmo de Máquina de soporte Vectorial.** Es un algoritmo de aprendizaje automático supervisado que se utiliza para tareas de clasificación y regresión. Es muy adecuado para datos donde se pueden dibujar márgenes claros de separación entre diferentes clases o grupos, encuentra el hiperplano óptimo que mejor separa

Tabla 1. Características del conjunto de datos.

Nombre de la característica	Descripción
Age	La edad del paciente
Anaemia	Valor booleano dependiendo de si el paciente tiene anemia
Creatinine Phosphokinase	Niveles de enzima
Diabetes	CPK en la sangre, mcg/L
Ejection Fraction	Valor booleano dependiendo de si el paciente tiene diabetes
High Blood Pressure	Porcentaje de sangre que sale del hogar en cada contracción
Platelets	Valor booleano si el paciente tiene presión arterial alta
Serum Creatinine	Recuento de plaquetas en la sangre, kiloplaquetas/mL
Serum Sodium	Nivel de creatinina sérica en la sangre mg/dl
Sex	Nivel de sodio sérico en la sangre mEq/L
Smoking	Valor binario si el paciente es una mujer u hombre
Time	Valor booleano si el paciente fuma
Death Event	El período de seguimiento en días

diferentes clases con el margen máximo, que es la distancia entre el hiperplano y los puntos de datos más cercanos que separa cada clase. Maximizar el margen garantiza una mejor generalización y robustez del modelo. Además, el truco del kernel permite que los datos lineales y no lineales se manejen transformando las características de entrada en un espacio dimensional más alto [13,27].

4. **Diseño factorial de experimentos.** Enfoque que utiliza la repetición de diferentes experimentos para estudiar la influencia de múltiples variables simultáneamente. Este método permite explorar los efectos individuales de varias variables en una variable dependiente. Como son conceptos básicos, tenemos factores: denotados como las variables que se están estudiando, niveles como los diferentes valores de esos factores y combinaciones [28-29].

3. Material y método

Presentamos un relato detallado de las metodologías empleadas en nuestro estudio sobre la clasificación de las mujeres adultas en riesgo de mortalidad después de tener un IAM, utilizando técnicas de aprendizaje automático y utilizando algoritmos SVM y RF.

3.1. Conjunto de datos

El conjunto de datos publicado en el artículo "La creatinina sérica y la fracción de eyección pueden predecir la supervivencia de los pacientes por insuficiencia cardíaca

contiene 105 mujeres y 194 hombres, que oscilan entre 40 y 95 años, este conjunto de datos se creó utilizando los datos recopilados en el Instituto de Cardiología de Faisalabad y en el Hospital Aliado en Faisalabad durante abril y diciembre del 2015. De los 299 pacientes, solo 96 murieron (32,10 %), específicamente, de las 105 mujeres, solo 34 murieron (32,38%), por otro lado, de los 194 hombres, 62 murieron, lo que corresponde a un 31,95 %.

Los 299 pacientes tenían disfunción sistólica del ventrículo izquierdo y tenían insuficiencia cardíaca previa [12]. Características como el sexo y el tiempo (Tabla 1) no son necesarias, nos centraremos en pacientes mujeres y esas características no proporcionan información valiosa.

3.2. Descripción del conjunto de datos

Comparamos un conjunto de características categóricas con la variable dependiente, este proceso se hace para tener una mejor comprensión del conjunto de datos. La Figura 1 no muestra que las mujeres anémicas sean especialmente más propensas a tener un resultado fatal.

Como muestra la Figura 2, las mujeres con IAM son igualmente propensas a la muerte si se les diagnostica diabetes. La Figura 3, revela que las mujeres fumadoras tienen una mayor incidencia de IAM mortal.

3.3. Algoritmos

Create ML de Apple es un conjunto de algoritmos de aprendizaje automático diseñados específicamente para desarrolladores que utilizan plataformas macOS y iPadOS.

Introducido por Apple en 2019; este marco permite a los desarrolladores construir, entrenar y probar modelos de aprendizaje automático. Se puede utilizar su aplicación fácil de usar o directamente mientras se codifica utilizando su librería de Swift [14]:

1. **Selección de algoritmos.** El artículo “Modelo mejorado de predicción de enfermedades cardiovasculares utilizando un algoritmo bosque aleatorio”. Logrando una precisión de más del 99% al clasificar a los pacientes con ECV, superando a la SVM, Regresión Logística, K-Means, entre otros [26]. Aunque SVM no funcionó tan bien como RF; el artículo “Early Coronary Heart Disease Deciphered via SVM: Insights from Experiments” demuestra que SVM es un algoritmo adecuado para un conjunto de datos de pacientes con IAM, siendo capaz de clasificar a los pacientes con una precisión del 87.8% usando el kernel sigmoide [30]. Decidimos usar la implementación de la herramienta Create ML de SVM y de RF. Ambos algoritmos proporcionan modelos de aprendizaje eficientes, complementa la facilidad de uso y aprovecha la capacidad de la arquitectura ARM para entrenar de manera eficiente [31].

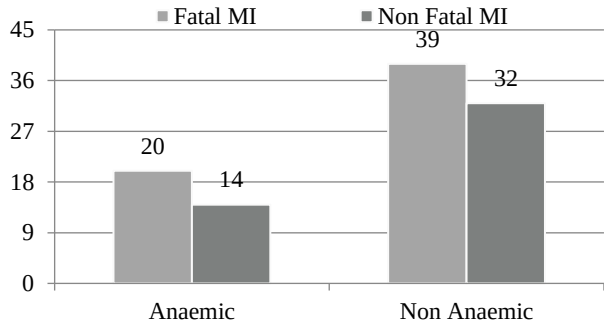


Fig. 1. Evento de muerte.

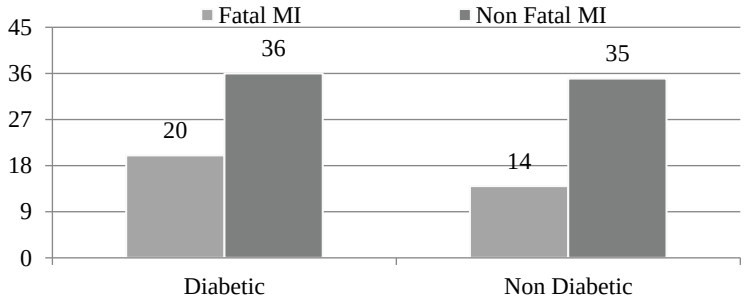


Fig. 2. Evento de muerte.

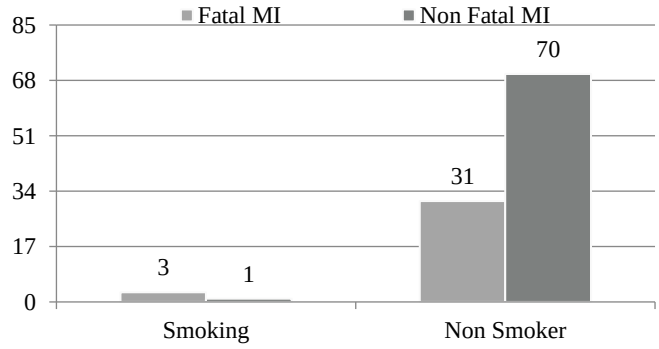


Fig. 3. Evento de muerte.

3.4. Experimentos

El conjunto de datos antes mencionado es utilizado para entrenar el modelo con los algoritmos de SVM y RF. Primero se lleva a cabo un análisis empírico para encontrar el mejor conjunto de parámetros para cada algoritmo, para enseguida, en base a los resultados anteriores, se realiza una búsqueda exhaustiva de parámetros utilizando un diseño de experimento factorial, luego se evalúa el modelo. La figura 4 representa los

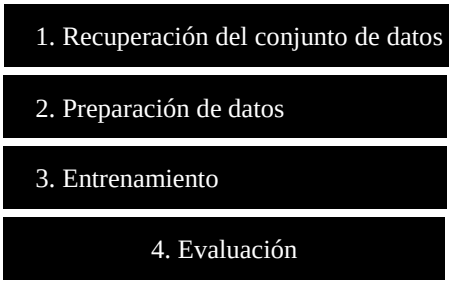


Fig. 4. Metodología.

pasos tomados para diseñar, entrenar y evaluar el modelo utilizando ambos algoritmos, SVM y RF:

1. **Recuperación del conjunto de datos.** El conjunto de datos se obtuvo durante la investigación destinada a predecir IAM utilizando una aplicación basada en el aprendizaje automático, centrada en las mujeres. Este conjunto de datos se obtuvo del documento "Machine learning can predict the survival of patients with heart failure from serum creatinine and the ejection fraction alone" de Chicco y Jurman (2020). Estos datos comprenden información clínica recopilada de pacientes en el Instituto de Cardiología de Faisalabad en Faisalabad (Punjab, Pakistán) entre abril y diciembre de 2015. De los 299 registros de este conjunto de datos, 105 pertenecen a mujeres, de los cuales 34 murieron desafortunadamente durante el período de estudio.
2. **Preparación de datos.** Nuestro enfoque se centra en los casos femeninos; para garantizarlo, hemos filtrado el conjunto de datos completo para incluir únicamente registros femeninos, creando un nuevo conjunto de datos específicamente adaptado para el propósito de este artículo. Para mantener una distribución de la función "DEATH_EVENT", asegurando una proporción aproximada de 3:10 entre las muertes y la supervivencia, hemos seleccionado cuidadosamente el conjunto de datos. Este conjunto de datos equilibrado se divide en entrenamiento y prueba, que sirven como base para pasos siguientes. Establecimos la división de entrenamiento comprenderá el 80% de este conjunto de datos filtrado, y la división de pruebas abarca el 20% restante. Con el fin de eliminar las características no correlativas, refinamos aún más este conjunto de datos, eliminamos las características de "sexo" y "tiempo", ya que el sexo no proporcionó ningún dato significativo, ya que todos los registros restantes donde la "mujer" y el período de seguimiento no brindan información sobre la mortalidad de un individuo en este contexto.
3. **Entrenamiento.** Nuestro enfoque inicial implicó una exploración empírica para identificar los parámetros óptimos para SVM [13] y RF [11], Después de encontrar un conjunto de parámetros para cada algoritmo individualmente, empleamos el diseño factorial de experimentos [29] utilizando una amplia gama de parámetros, cada resultado de cada combinación se registró en un archivo CSV, lo cual nos permitió encontrar la configuración de parámetros más efectiva. Durante esta fase, se observó que Create ML genera automáticamente una división de validación. Aunque Apple no ha revelado detalles específicos, como los parámetros utilizados

Tabla 2. Parámetros de SVM para el diseño factorial de experimentos.

Parámetros	**Valores**	**Valor *k***
Max Iterations	100 and 1,000	$k_1 = 2$
Penalty	25 to 65, steps by 0.1	$k_2 = 400$
Convergence Threshold	0.001	$k_3 = 1$
Feature Rescaling	1	$k_4 = 1$

Tabla 3. Resultados del diseño factorial de experimentos para SVM.

Parámetros	**Matriz de confusión**	**Precisión**
Max Iterations: 100 Penalty: 35.0 Convergence Threshold: 0.001 Feature Rescaling: true	True\Pred false true false 14 0 true 3 4	85.71%

en este mecanismo de división [14], observamos que sirve como retroalimentación para cada iteración durante la fase de entrenamiento, mejorando la precisión de los modelos.

4. **Evaluación.** En esta fase, evaluamos los resultados obtenidos de la frase anterior, revisando los archivos creados con la precisión de cada resultado combinado del Diseño del Experimento Factorial e implicamos rigurosas pruebas y optimización para nuestros modelos de aprendizaje automático, con esta información.

4. Resultados

El algoritmo SVM obtiene una puntuación del 85% de precisión después de evaluar el modelo utilizando el conjunto de datos de prueba, después de eso, el algoritmo RF dio la mejor puntuación de evaluación con un 90% de precisión.

4.1. Support Vector Machine

Al probar el parámetro de iteraciones máximas de SVM como parámetro de convergencia, encontramos que 100 iteraciones máximas dieron resultados prometedores, estos valores fueron suficientes para que el algoritmo proporcionara un modelo maduro con resultados prometedores, aunque, al probar algunos valores necesitaba más iteraciones; decidimos también probar con 1,000 iteraciones máximas, para el parámetro de penalización encontramos que los valores de 25 a 65 el algoritmo estaba dando resultados prometedores. El umbral de convergencia se mantuvo en un valor de 0.001 para asegurar resultados, el parámetro kernel se utilizó para todos los experimentos. Los parámetros para el diseño factorial de experimentos se muestran en la tabla 2:

Table 4. Random forest parameters for the factorial experiment design.

Parámetros	Valores	Valor k
Max Depth	4 to 6, steps of 1	$k_1 = 3$
Max Iterations	15 to 130, steps of 15	$k_2 = 8$
Min Loss Reduction	0.001 to 0.41, steps by 0.05	$k_3 = 9$
Min Child Weight	0.001 to 0.41, steps by 0.05	$k_4 = 9$
Row Subsample	80%	$k_5 = 1$
Column Subsample	80%	$k_6 = 1$
Random Seed	46	$k_7 = 1$

Table 5. Factorial experiment design results for random forest.

Parámetros	Matriz de confusión			Precisión
Max Depth: 4 Max Iterations: 60 Min Loss Reduction: 0.001 Min Child Weight: 0.001 Random Seed: 46 Row Subsample: 0.8 Column Subsample: 0.8	**True\Pred**	**False**	**True**	90.48%
	False	14	0	
	False	2	5	

$$30(2 \times 400) = 24{,}000\,. \tag{1}$$

Como muestra la fórmula (1), el número de iteraciones del experimento factorial es de 24.000. Este experimento nos dio el mejor resultado, como se revela en la Tabla 3 Los resultados del modelo en la Tabla 3 no cumplen con nuestras expectativas.

4.2. Bosque aleatorio

Cuando se entrenó el algoritmo de RF con mil iteraciones máximas para converger, nos dio peores resultados, ya que era el parámetro de parada. Encontramos que el número de iteraciones que nos dieron resultados prometedores para este parámetro estaba entre 19 y 120 iteraciones máximas. Para el parámetro de profundidad máxima, los mejores resultados que obtuvimos de 4 y el algoritmo se detuvo para dar buenos resultados al nivel de profundidad 6. La reducción de la pérdida mínima y los parámetros de peso mínimo del niño dieron los mejores resultados entre 0.001 y 0.4.

Por último, los parámetros de submuestra de fila y submuestra de columna obtuvieron el mejor rendimiento a 0.8. Realizamos múltiples iteraciones para encontrar los parámetros óptimos para la convergencia. Encontramos que el uso de mil iteraciones condujo a malos resultados. Descubrimos que el punto óptimo para el número de iteraciones que proporcionaba mejores resultados, oscilaba entre 19 y 120. Del mismo modo, para determinar el parámetro de máxima profundidad, encontramos al algoritmo produjo constantemente resultados favorables dentro del rango de 4 a 6 niveles.

Específicamente, el algoritmo dejó de producir mejoras significativas más allá del sexto nivel. Además, el ajuste de los parámetros de “Min Loss Reduction” y “Min Child

Weight" resultó crítico. Los mejores resultados se lograron cuando estos valores se establecieron entre 0.001 y 0.4. Además, observamos un rendimiento óptimo al establecer los parámetros de "submuestra de fila" y "submuestra de columna" en 0.8. Estos hallazgos destacan la importancia de un ajuste meticuloso de los parámetros para aprovechar el verdadero potencial del algoritmo del bosque aleatorio. Teniendo en cuenta los resultados, los parámetros utilizados en el diseño factorial de experimentos [28-29] se muestran en la Tabla 4:

$$30(3 \times 8 \times 9 \times 9) = 58{,}320\,. \tag{2}$$

Como muestra la fórmula (2), el número de iteraciones del experimento factorial, que es de 58,320. En nuestra configuración actual, tardó varias horas en completarse, pero finalmente los resultados. Estos resultados finales mostraron en la Tabla 5 que RF superó a SVM con una mejor precisión del modelo en comparación con los resultados de la Tabla 3 de SVM, pero, como se ve en la matriz de confusión de los resultados de ambos modelos (Tabla 3 y Tabla 5), los modelos tuvieron problemas en identificar casos verdaderos, esto debido a la escasa exposición de los algoritmos a los casos verdaderos, lo que lo hace más débil en esta clase. En este caso, el algoritmo de RF superó a SVM, esto podría deberse al espacio de búsqueda estrecho para SVM, que en el caso de RF el espacio de búsqueda para los hiperparámetros dio buenos resultados sin modificar el espacio de búsqueda.

5. Conclusiones

Este modelo funciona de manera efectiva, logrando predicciones precisas en pacientes dentro de grupos de edad y características similares. Además, es crucial reconocer que la raza juega un papel importante en la influencia de la precisión del modelo, teniendo experiencia únicamente para esta población.

5.1. Trabajo futuro

La expansión del alcance de este clasificador médico implica el potencial de mejorar la esperanza de vida de las mujeres con IAM. Teniendo al alcance un grupo más grande y diverso de pacientes, que permita capturar un espectro amplio de perfiles de salud, lo cual podría permitir que el modelo reconozca patrones y correlaciones con otros factores que puedan ser significativos para la enfermedad. Es importante tomar en cuenta rango de edad, lo cual permitirá tener en cuenta los factores fisiológicos únicos en las diferentes etapas de la vida. Además, considerar una región geográfica más amplia puede introducir variables que podrían tener un impacto significativo en los resultados de salud.

Referencias

1. World Health Organization: The top 10 causes of death. http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (2024)

2. Williams, L., McKnight, E., Gillard, C.: Myocardial infarction and older women. U.S. Pharmacist. http://www.uspharmacist.com/article/myocardial-infarction-and-older-women (2016)
3. Instituto Nacional de Estadística y Geografía: Características de las defunciones registradas en México durante 2020. Comunicado de Prensa, no. 402, vol. 21 (2021)
4. Instituto Nacional de Estadística y Geografía: Estadística de defunciones registradas de enero a junio de 2022. Comunicado de Prensa, no. 29, vol. 23 (2023)
5. Wenger, N.K.: Women and Coronary Heart Disease: A Century After. Herrick: Understudied, Underdiagnosed, and Undertreated. Circulation, vol. 126, no. 5, pp. 604–611 (2012). DOI: 10.1161/circulationaha.111.086892.
6. Stehli, J., Martin, C., Brennan, A., Dinh, D.T., Lefkovits, J., Zaman, S.: Sex Differences Persist in Time to Presentation, Revascularization, and Mortality in Myocardial Infarction Treated with Percutaneous Coronary Intervention. Journal of the American Heart Association, vol. 8, no. 10 (2019). DOI: 10.1161/jaha.119.012161.
7. Wu, J., Gale, C.P., Hall, M., Dondo, T.B., Metcalfe, E., Oliver, Batin, G., Hemingway, P.D., Timmis, H., West, A., Robert, M.: Editor's Choice - Impact of Initial Hospital Diagnosis on Mortality for Acute Myocardial Infarction: A National Cohort Study. European Heart Journal: Acute Cardiovascular Care, vol. 7, no. 2, pp. 139–148 (2016). DOI: 10.1177/2048872616661693.
8. Kwok, C.S., Bennett, S., Azam, Z., Welsh, V., Potluri, R., Loke, Y.K., Mallen, C.D.: Misdiagnosis of Acute Myocardial Infarction: A Systematic Review of the Literature. Critical Pathways in Cardiology: A Journal of Evidence-Based Medicine, vol. 20, no. 3, pp. 155– 162 (2021). DOI: 10.1097/hpc.0000000000000256.
9. Arnstein, P.M., Buselli, E.F., Rankin, S.H.: Women and Heart Attacks: Prevention, Diagnosis, and Care. The Nurse Practitioner, vol. 21, no. 5, pp. 57–71 (1996). DOI: 10.1097/00006205-199605000-00005.
10. Chandrasekhar, J., Gill, A., Mehran, R.: Acute Myocardial Infarction in Young Women: Current Perspectives. International Journal of Women's Health, vol. 10, pp. 267–284 (2018). DOI: 10.2147/ijwh.s107371.
11. Breiman, L.: Random Forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001). DOI: 10.1023/a:1010933404324.
12. Chicco, D., Jurman, G.: Machine Learning can Predict Survival of Patients with Heart Failure from Serum Creatinine and Ejection Fraction Alone. BMC Medical Informatics and Decision Making, vol. 20, no. 1, pp. 16 (2020). DOI: 10.1186/s12911-020-1023-5.
13. Amari, S., Wu, S.: Improving Support Vector Machine Classifiers by Modifying Kernel Functions. Neural Networks, vol. 12, no. 6, pp. 783–789 (1999). DOI: 10.1016/s0893-6080(99)00032-5.
14. Apple Developer Documentation: Create ML–Create Machine Learning Models for Use in your App. http://developer.apple.com/documentation/createml (2023)
15. Fedorova, E.L., Bondareva, Z.G., Kuimov, A.D., Nesterenko, E.V.: Myocardial Infarction in Women: Risk Factors and Clinical Features. Klinicheskaya Meditsina (Moskva), vol. 81, no. 6, pp. 28–32 (2003)
16. Arroyo-Quiroz, C., Barrientos-Gutierrez, T., O'Flaherty, M., Guzman-Castillo, M., Palacio-Mejia, L., Osorio-Saldarriaga, E., Rodriguez-Rodriguez, A.Y.: Coronary Heart Disease Mortality is Decreasing in Argentina, and Colombia, but Keeps Increasing in Mexico: A

Time Trend Study. BMC Public Health, vol. 20, no. 1, pp. 162 (2020). DOI: 10.1186/s12889-020-8297-5.
17. World Health Organization: Deaths by Sex and Age Group for a Selected Country or Area and Year. WHO Mortality Database. http://platform.who.int/mortality/themes/theme-details/MDB/all-causes (2024)
18. Libby, P., Theroux, P.: Pathophysiology of Coronary Artery Disease. Circulation, vol. 111, no. 25, pp. 3481–3488 (2005). DOI: 10.1161/circulationaha.105.537878.
19. Institute of Medicine (US): Committee on Social Security Cardiovascular Disability Criteria: Cardiovascular Disability: Updating the Social Security Listings. National Academies Press (2010). DOI: 10.17226/12940.
20. Wizemann, T.M., Pardue, M.-L.: Exploring the Biological Contributions to Human Health: does Sex Matter? The National Academies Collection: Reports Funded by National Institutes of Health (2001). DOI: 10.17226/10028.
21. Vest, A.R., Cho, L.: No Woman Left Behind: Recognizing and Responding to Cardiogenic Shock in Younger Women. Circulation: Heart Failure, vol. 13, no. 10 (2020). DOI: 10.1161/circheartfailure.120.007782.
22. Hunt, E.B.: Artificial Intelligence. Academic Press (1975)
23. Zhou, Zhi-Hua: Machine Learning. Springer Singapore (2021). DOI: 10.1007/978-981-15-1967-3
24. Cutler, D. Richard, Edwards, Thomas C., Beard, Karen H., Cutler, A., Hess, K.T., Gibson, J., Lawler, J.J.: Random Forests for Classification in Ecology. Ecology, vol. 88, no. 11, pp. 2783–2792 (2007). DOI: 10.1890/07-0539.1.
25. Belgiu, M., Drăguţ, L.: Random Forest in Remote Sensing: A Review of Applications and Future Directions. (ISPRS) Journal of Photogrammetry and Remote Sensing, vol. 114, pp. 24–31 (2016). DOI: 10.1016/j.isprsjprs.2016.01.011.
26. Al-Manaseer, H., Abualigah, L., Alsoud, A. R., Zitar, R.A., Ezugwu, A.E., Jia, H.: A Novel Big Data Classification Technique for Healthcare Application using Support Vector Machine, Random Forest and J48. Classification Applications with Deep Learning and Machine Learning Technologies, Studies in Computational Intelligence, Springer, Cham, vol. 1071, pp. 205–215 (2022). DOI: 10.1007/978-3-031-17576-3_9.
27. Suthaharan, S.: Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning. Integrated Series in Information Systems, Springer US (2016). DOI: 10.1007/978-1-4899-7641-3.
28. Yates, F.D.: The Design and Analysis of Factorial Experiments. Imperial Bureau of Soil Science, pp. 96 (1978)
29. Spall, J.C.: Factorial Design for Efficient Experimentation. IEEE Control Systems, vol. 30, no. 5, pp. 38–53 (2010). DOI: 10.1109/mcs.2010.937677.
30. Akhtar, F., Heyat, B.B., Parveen, S., Singh, P., Hassan, M.F.UI, Parveen, S., Hayat, M.A.B., Sayeed, E., Ali, A., Li, J.P., Sawan, M.: Early Coronary Heart Disease Deciphered Via Support Vector Machines: Insights from Experiments. In: Proceedings of the International Computer Conference on Wavelet Active Media Technology and Information Processing, pp. 1–7 (2023). DOI: 10.1109/iccwamtip60502.2023.10387051.
31. Machine Learning Research: Deploying Transformers on the Apple Neural Engine. Computer Vision, Research Area Speech and Natural Language Processing. http://machinelearning. apple.com/research/neural-engine-transformers (2022)

ISSN 1870-4069

Estudio de técnicas de aprendizaje automático para la estimación de la humedad del suelo en agricultura

Noel A. Zavala-Díaz, Juan C. Olivares-Rojas,
Jonathan Zavala-Díaz, Enrique Reyes-Archundia,
Adriana Téllez-Anguiano, Gerardo M. Chávez-Campos,
Arturo Méndez-Patiño

Tecnológico Nacional de México
División de Estudios de Posgrado e Investigación
México

{m21121662, juan.or, d19123006, enrique.ra, adriana.ta,
gerardo.cc, arturo.mp}@morelia.tecnm.mx

Resumen. La humedad del suelo es crucial en diversos campos, y su monitoreo para guiar el riego representa un desafío. El aprendizaje automático ha surgido como una herramienta prometedora para predecir con precisión los niveles de humedad del suelo. Este estudio se centra en evaluar técnicas de aprendizaje automático para esta tarea, entrenando modelos con variables meteorológicas y mediciones directas de humedad del suelo. Se implementaron cuatro algoritmos de aprendizaje automático, destacando el Gradient Boosting Regressor como el más efectivo. Además, se presenta un conjunto de datos procesado que combina mediciones meteorológicas y de humedad del suelo, esperando que sea útil para futuras investigaciones. Este enfoque busca mejorar la comprensión y la capacidad de previsión de la humedad del suelo, crucial para la planificación agrícola y la gestión del agua en la agricultura.

Palabras clave: Humedad del suelo, aprendizaje automático, modelos de regresión.

Study of Machine Learning Techniques for the Estimation of Soil Moisture in Agriculture

Abstract. Soil moisture is crucial in various fields and monitoring it to guide irrigation represents a challenge. Machine learning has emerged as a promising tool to accurately predict soil moisture levels. This study focuses on evaluating machine learning techniques for this task, training models with meteorological variables and direct soil moisture measurements. Four machine learning algorithms were implemented, highlighting the Gradient Boosting Regressor as the most effective. In addition, a processed data set that combines meteorological and soil moisture measurements is presented, hoping that it will be useful for future research. This approach seeks to improve the compression and predictability of soil moisture, crucial for agricultural planning and water management in agriculture.

Keywords: Soil moisture, machine learning, regression models.

1. Introducción

El contenido de humedad del suelo es de vital importancia para una variedad de campos, incluyendo la biología, la hidrología, la agronomía, la ingeniería, la ecología y la geología del suelo. Su monitoreo es cada vez más extenso, especialmente con el incremento de inversiones en infraestructura de riego de precisión y sistemas de control. Sin embargo, la tarea de monitorear la humedad del suelo para guiar el riego presenta desafíos significativos. Los regantes deben seleccionar cuidadosamente el equipo adecuado para su sistema de riego y las características específicas de su parcela de tierra [1].

La humedad del suelo desempeña un papel crucial en el suministro de agua para la agricultura, siendo este su recurso principal. A pesar de su importancia, la medición directa en el campo enfrenta desafíos significativos, lo que subraya la necesidad de predecirla con precisión para respaldar actividades de planificación agrícola e investigaciones pertinentes [2].

El uso del aprendizaje automático ha dado lugar al desarrollo de algoritmos innovadores capaces de pronosticar de manera precisa los niveles de humedad del suelo, los cuales pueden ser empleados posteriormente en actividades de riego u otros propósitos [1]. Actualmente existen trabajos que aplican aprendizaje automático en la predicción de humedad del suelo. En [3] realizan la estimación de la humedad del suelo mediante aprendizaje profundo basado en datos satelitales.

Los autores en [4] utilizan la técnica de regresión denominada Máquina de Vectores de Soporte (SVM) para estimar la humedad del suelo mediante el uso de datos de teledetección. En el trabajo [5], se desarrollan y examinan modelos híbridos que combinan máquinas de aprendizaje extremo (ELM) con inteligencia de datos para realizar predicciones mensuales de humedad del suelo. En [6] hacen una estimación de la humedad del suelo basada en datos de teledetección y aprendizaje profundo.

Este estudio se enfoca en el uso de técnicas de aprendizaje automático para estimar la humedad del suelo. Motivado por encontrar una correlación entre datos meteorológicos y mediciones de humedad en el suelo dada la ausencia de sensores propios con datos abundantes que nos den una base sólida en cual sustentar otras investigaciones. Se entrenaron modelos con variables meteorológicas y mediciones directas de humedad del suelo.

Implementamos cuatro algoritmos de aprendizaje automático: Random Forest Regressor, K-Nearest Neighbors, Gradient Boosting Regressor y Regresión Lineal Múltiple. Al evaluar estos modelos con predicciones, encontramos que el Gradient Boosting Regressor demostró un error cuadrático medio y error absoluto medio menor en comparación con los otros modelos, especialmente al probar en intervalos de tiempo diferentes al de entrenamiento.

Finalmente, se aplicó este modelo a datos recientes de la estación meteorológica del Instituto Tecnológico de Morelia, obteniendo estimaciones de humedad del suelo consistentes y congruentes con el comportamiento esperado a lo largo del tiempo.

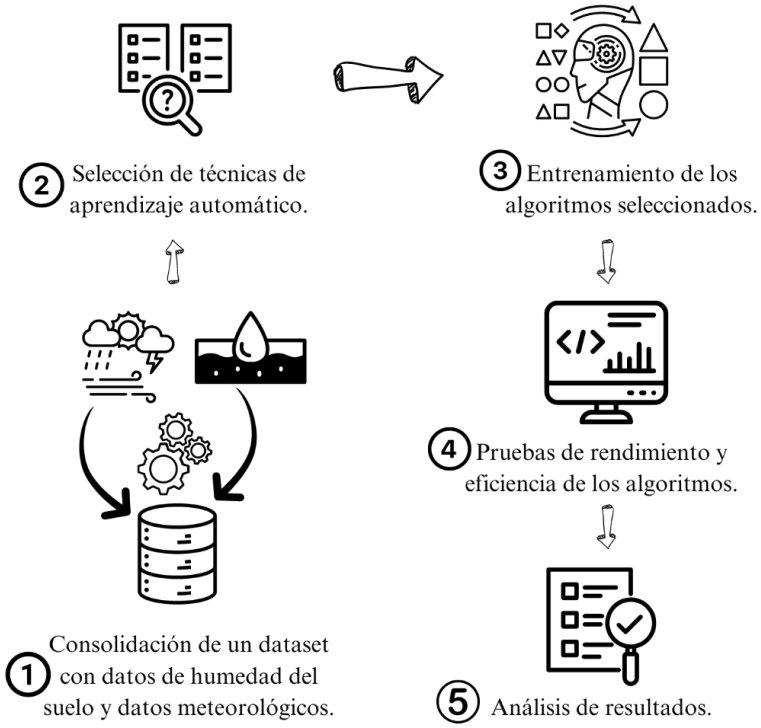


Fig. 1. Metodología.

Entre las contribuciones de este estudio, destacamos la presentación de un conjunto de datos creado por Gasch et al. [11]. Este conjunto de datos se procesó para ser presentado en un archivo CSV, extrayéndolo de su formato original en TXT.

Identificamos las ubicaciones de los sensores con la menor cantidad de datos faltantes y utilizamos el método de interpolación de vecinos más cercanos dado su estructura temporal para llenar los valores faltantes. Además, enriquecimos este conjunto de datos con información de una estación meteorológica cercana a las ubicaciones donde se tomaron las mediciones de humedad del suelo. De esta manera, creamos un conjunto de datos que integra mediciones meteorológicas y de humedad del suelo. Este conjunto de datos servirá de base para futuras investigaciones, ya sea para buscar patrones o realizar análisis temporales.

2. Marco teórico

2.1. Humedad del suelo

La agricultura y el agua están profundamente vinculadas, siendo el agua un factor esencial en la producción agrícola. Los métodos agrícolas influyen en el ciclo

Tabla 1. Variables encontradas en dataset meteorológico, definiciones y valores nulos.

Variable	Definición	Valores nulos
time	Tiempo	0
temp	Temperatura	213
dwpt	Punto de rocío	256
rhum	Humedad relativa	256
prcp	Precipitación	6558
snow	Profundidad de nieve	**80283**
wdir	Dirección del viento	23414
wspd	Velocidad media del viento	414
wpgt	Ráfaga máxima de viento	**80283**
pres	Presión	1125
tsun	Tiempo de sol	**80283**
coco	Código de condición climática.	**80283**
Total		**353368**

hidrológico mediante la evapotranspiración, la recarga de acuíferos y el flujo de aguas superficiales.

Una humedad adecuada en el suelo es crucial para varios procesos biológicos y físicos, incluyendo la germinación de semillas, el desarrollo vegetativo, el ciclo de nutrientes y la conservación de la biodiversidad del suelo. La medición de la humedad del suelo es esencial no solo para evaluar la disponibilidad de agua para la agricultura, sino también para entender la salud del suelo y su capacidad para retener agua, lo cual es vital para el mantenimiento de un agroecosistema sostenible [7].

La humedad del suelo es un factor crucial en la agricultura, pues influye directamente en el crecimiento de los cultivos y en la sostenibilidad de los ecosistemas agrícolas. Dicha humedad no solo depende de las prácticas de irrigación y del manejo del suelo, sino que está estrechamente vinculada a diversas variables climáticas.

2.2. Técnicas de aprendizaje automático

Las técnicas de aprendizaje automático, como Random Forest Regressor, K-Nearest Neighbors, Gradient Boosting Regressor y Regresión Lineal Múltiple son herramientas poderosas para predecir valores en una variedad de contextos. Estos algoritmos pueden emplearse para modelar relaciones complejas entre variables y generar predicciones precisas sobre valores futuros. Los autores en [8] presentan la aplicación de un método de aprendizaje automático en específico Random Forest Regressor para generar pronósticos diarios precisos de generación de energía solar, haciendo uso de datos históricos de mediciones y datos meteorológicos de fuentes abiertas suministrados por servicios meteorológicos.

En [9], se propone un método que utiliza el algoritmo de K-Vecinos Más Cercanos (KNN) para evaluar la calidad del suelo y predecir los cultivos más adecuados. Este enfoque considera la temperatura y la calidad del suelo como variables de entrada para el algoritmo. El artículo [10] describe el uso de modelos de aprendizaje automático para predecir la evapotranspiración de referencia, facilitando así la planificación del riego.

Fig. 2. Valor promedio mensual del dataset de variables meteorológicas.

Se emplearon datos meteorológicos diarios, que incluyen la temperatura máxima y mínima, humedad relativa, radiación solar, temperatura del suelo y velocidad del viento. Los datos se procesaron utilizando técnicas de Regresión Lineal Múltiple, Random Forest Regressor y Gradient Boosting Regressor Los resultados indicaron que el modelo preprocesado con GBR superó a los otros modelos en la precisión de las predicciones de evapotranspiración de referencia.

En el estudio [2], el objetivo principal fue predecir la humedad diaria del suelo a nivel de cultivo utilizando información meteorológica a través de modelos de regresión lineal múltiple. Se concluyó que estos modelos, al incorporar variables meteorológicas, resultaron efectivos para estimar la humedad del suelo. Esto se debe a que la humedad tendió a replicar los patrones estacionales y a responder a las variaciones en las precipitaciones. Que les falta a los demás trabajos o que tiene este trabajo que no tengan los mismos.

2.3. Datasets

En [11] presentan un conjunto de datos obtenidos del monitoreo del contenido de agua en el suelo, así como datos complementarios, recolectados en una granja experimental de labranza cero de 37 hectáreas situada en el noroeste de los Estados Unidos. Las mediciones del contenido de agua se han realizado cada hora desde el año 2007 mediante sensores ECH2O-TE y 5TE distribuidos en 42 ubicaciones, abarcando cinco profundidades (0.3; 0.6; 0.9; 1.2 y 1.5 metros), sumando un total de 210 sensores en toda la granja agronómica RJ Cook. Este conjunto de datos se encuentra disponible en [12].

Este conjunto de datos cuenta con mediciones horarias y diarias del contenido de agua (en m^3/m^3) y la temperatura del suelo (en °C) en 42 ubicaciones y en cinco profundidades (0.3; 0,6; 0,9; 1,2 y 1,5 metros) desde el 20 de abril de 2007 hasta el 16 de junio de 2016. Los datos se encuentran en archivos .txt para cada ubicación.

Tabla 1. Variables encontradas en dataset de humedad del suelo, definiciones y valores nulos.

Variable	Definición	Valores nulos
H_30cm	Humedad a 30 cm	18806
H_60cm	Humedad a 60 cm	23701
H_90cm	Humedad a 90 cm	22323
H_120cm	Humedad a 120 cm	24540
H_150cm	Humedad a 150 cm	25577
T_30cm	Temperatura a 30 cm	18806
T_60cm	Temperatura a 60 cm	23705
T_90cm	Temperatura a 90 cm	22330
T_120cm	Temperatura a 120 cm	24540
T_150cm	Temperatura a 150 cm	25578
Total		229906

El sitio web meteostat.net, es una base de datos meteorológicos y climáticos que proporciona datos detallados de miles de estaciones meteorológicas y lugares de todo el mundo. Afortunadamente, cuenta con una estación en Pullman, muy cerca de R.J. Cook Agronomy Farm, donde se tomaron las mediciones de contenido de agua del suelo y datos auxiliares a diferentes profundidades [13]. El sitio web [13], nos da la oportunidad de obtener datos de diferentes formas, más sin embargo cuando se descarga en un periodo de 7 días (una semana) los datos obtenidos tienen una frecuencia de cada hora lo cual es similar al dataset [12].

3. Metodología

La metodología empleada en este estudio se muestra en la Fig. 1. En primer lugar, se consolida un conjunto de datos que incluye humedad del suelo y datos meteorológicos (ver Sección 3.1). Luego, se seleccionan las técnicas de aprendizaje automático a utilizar. A continuación, se entrenan los algoritmos seleccionados para estimar la humedad del suelo. Posteriormente, se realizan pruebas y se evalúa la eficacia de los algoritmos elegidos. Finalmente, se analizan los resultados obtenidos.

3.1. Consolidar dataset que contenga la humedad del suelo y los datos meteorológicos

Esta etapa consta de consolidar un dataset que contenga la humedad del suelo del dataset [12] y los datos meteorológicos obtenidos de [13]. Primeramente, en el apartado 3.1.1 se muestra el proceso de obtención del dataset de datos meteorológicos.

3.1.1. Dataset de datos meteorológicos

El proceso de obtención de los datos meteorológicos consta de seleccionar el periodo de medición de 7 días a partir del día 20/4/2007 dentro de la estación de Pullman en el repositorio de meteostat.net, para después descargar el archivo seleccionando el

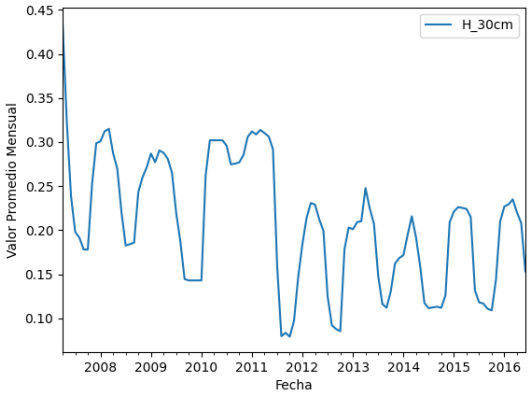


Fig. 1. Valores promedio mensuales de la humedad del suelo.

formato tipo CSV, de esta forma obtendremos archivos que contendrán la información de cada semana con frecuencia de cada hora. El proceso se repite hasta que llegamos a la fecha 16/6/2016, la cual compete al periodo del dataset [12] con el que se consolidara un nuevo dataset que contengan los datos de humedad del suelo y datos meteorológicos.

El primer reto fue la unión de todos los archivos descargados para la creación del dataset meteorológico, primero por año, y luego uno general que competerá a todo el periodo de prueba, para ello, generamos un código en Python que nos facilite la unión y ordenamiento de los datos de forma temporal.

En la Tabla 1 se muestra la información del dataset creado a partir de los datos meteorológicos, cuenta un número total de *80,283 registros* por variable y *12 variables*, se observa en la tabla que tanto las variables de snow, wpgt, tsun y coco, todos sus valores son nulos, por lo cual se eliminaran del dataset. Para las demás variables utilizaremos métodos para el llenado de estos valores nulos.

Existen varios métodos como: Imputación por media, mediana o moda, Regresión lineal o regresión múltiple, MICE (Multiple Imputation by Chained Equations), Matrix Factorization, Algoritmos de aprendizaje automático avanzados e Interpolación.

Dado que los datos están ordenados en el tiempo, es decir, tienen una estructura temporal, se puede utilizar métodos de interpolación para predecir los valores faltantes basados en los valores existentes.

El método de interpolación “nearest” o interpolación por vecino más cercano, es una forma de interpolación que se basa en la idea de que los valores cercanos en el tiempo (o en la secuencia) son más similares entre sí, por lo que el valor más cercano será una buena aproximación para el valor faltante. En la Fig. 2 se muestran las variables meteorológicas de este dataset.

3.1.2. Dataset de humedad del suelo

Como se mencionó anteriormente, este conjunto de datos incluye registros horarios y diarios de la humedad del suelo (expresada en m^3/m^3) y de la temperatura del suelo (en °C) en 42 ubicaciones diferentes y a cinco profundidades distintas (0.3; 0.6; 0.9; 1.2 y 1.5 metros). Estas mediciones se extienden desde el 20 de abril de 2007 hasta el 16 de junio de 2016. Los datos están almacenados en archivos de texto separados por ubicación. El primer desafío fue determinar cuál de estas ubicaciones ofrecía la mejor

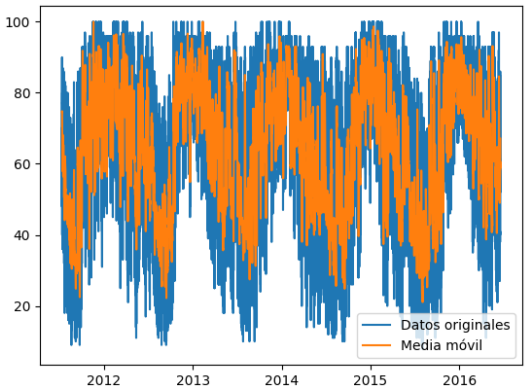


Fig. 2. Humedad relativa con y sin filtro.

calidad de datos, es decir, aquella que presentaba la menor cantidad de registros nulos. Esta selección fue crucial para asegurar la fiabilidad de los análisis subsiguientes.

Después de analizar los archivos de texto correspondientes, se determinó que el archivo CAF308.txt presentaba la menor cantidad de valores nulos en comparación con los archivos de las otras 42 ubicaciones de los sensores, específicamente en lo que respecta a las mediciones de humedad. En la Tabla 2 se detallan las variables del dataset de este archivo seleccionado para este estudio, como lo son la humedad del suelo y temperatura a diferentes profundidades, junto con la cantidad de valores nulos encontrados para cada variable.

Se observa que la medida de humedad del suelo a una profundidad de 30 cm es la que contiene menos valores nulos en comparación con las demás profundidades. El número total de valores de estas mediciones que se extienden desde el 20 de abril de 2007 hasta el 16 de junio de 2016 debería ser de *80283 registros*. Para abordar los valores faltantes, se empleó el método de interpolación "nearest" o interpolación por vecino más cercano. Al ser datos ordenados de forma temporal los valores cercanos en el tiempo tienden a ser más similares entre sí, lo que hace razonable suponer que el valor más próximo es una aproximación adecuada para el valor faltante.

En la Figura 3 se presenta el gráfico del valor promedio mensual de la humedad del suelo a una profundidad de 30 cm, que será el enfoque principal de este estudio. Se puede observar que abarca el período desde 2007 hasta 2016. Sin embargo, a simple vista se aprecia una tendencia más clara y significativa en los años comprendidos entre 2012 y 2016. Por lo tanto, para los análisis posteriores, nos centraremos en este intervalo de tiempo.

3.1.3. Dataset de humedad del suelo y datos meteorológicos

Después de adquirir los conjuntos de datos de humedad del suelo y datos meteorológicos, los combinamos en un único dataset. Luego, procedimos a crear una matriz de correlación entre las variables para explorar posibles relaciones. Dado que nuestra hipótesis sugería que la humedad relativa podría correlacionarse con la humedad del suelo, generamos un gráfico de la humedad relativa y aplicamos un filtro para suavizar el ruido. Utilizamos una media móvil con una ventana de tamaño 24, como se muestra en la Fig. 4.

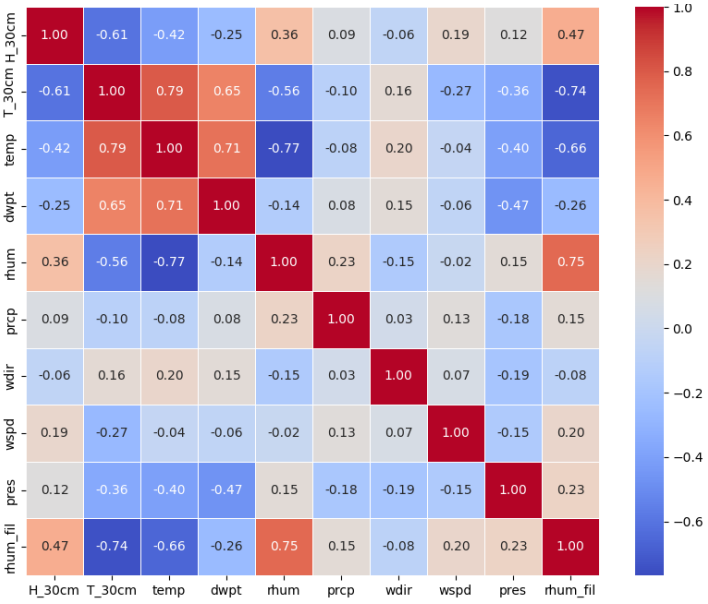


Fig. 5. Matriz de correlación de variables.

La ventana de tamaño 24 para la media móvil se seleccionó para coincidir con la periodicidad diaria de los datos recogidos cada hora, permitiendo cubrir un ciclo completo.

Este tamaño de ventana facilita el suavizado de las fluctuaciones diarias y resalta las tendencias claras en la humedad relativa, proporcionando una base sólida para analizar los efectos diarios en la humedad del suelo. En la Figura 3 se presenta la matriz de correlación entre las variables. Destaca que la correlación entre la Humedad del suelo (H_30cm) y la Humedad relativa (rhum) es de 0.36. Sin embargo, al aplicar el filtro de media móvil a la Humedad relativa, como se muestra en la Fig. 5, esta correlación aumenta a 0.47.

4. Resultados

Se realizó un análisis comparativo de cuatro técnicas de aprendizaje automático (Random Forest Regressor, Gradient Boosting Regressor, K-Nearest Neighbors y Regresión Lineal Múltiple) para generar nuevos valores de humedad del suelo para fechas futuras, utilizando variables meteorológicas. Las variables seleccionadas fueron temperatura (temp), punto de rocío (dwpt), humedad relativa (rhum), precipitación (prcp) y el mes correspondiente. La selección de estas variables se justifica por la siguiente razón:

La humedad relativa mostró una correlación más alta con la humedad del suelo, como se observa en la Fig. 6. Además, dado el comportamiento cíclico observado en la Figura 6, se decidió incluir el mes como característica para el entrenamiento de los modelos. Las otras variables complementarias fueron seleccionadas porque están disponibles para futuros trabajos, utilizando datos meteorológicos de la ciudad de

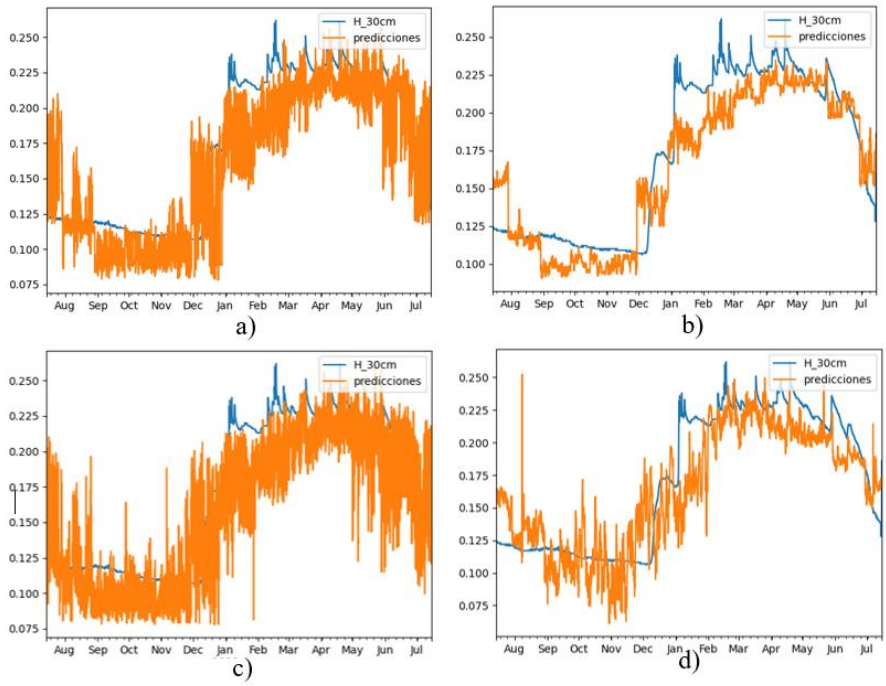


Fig. 3. Predicción de humedad del suelo de los diferentes modelos entrenados, a) Random Forest Regressor b) Gradient Boosting Regressor c) K-Nearest Neighbors y d) Regresión Lineal Múltiple.

Morelia, y serán utilizadas para generar valores sintéticos de humedad del suelo a partir del modelo entrenado

Para entrenar los modelos de aprendizaje automático, incluyendo Random Forest Regressor, Gradient Boosting Regressor, K-Nearest Neighbors y Regresión Lineal Múltiple, se utilizó Python junto con la biblioteca Scikit-learn. Scikit-learn es una herramienta de código abierto que ofrece una amplia variedad de algoritmos tanto supervisados como no supervisados para el aprendizaje automático. En este estudio, se optó por emplear los valores predeterminados de los hiperparámetros para cada modelo, lo que garantiza una configuración estándar y coherente durante el proceso de entrenamiento.

Para entrenar el modelo, se empleó el conjunto de datos presentado en la sección 3.3, que incluye información sobre la humedad del suelo y datos meteorológicos. Para el proceso de entrenamiento, se decidió utilizar el período de tiempo comprendido entre 2011 y 2015, reservando el intervalo de 2015 a 2016 para probar el modelo.

Durante el entrenamiento con los datos de 2011 a 2015, se asignó el 80% de los datos para entrenamiento y el 20% restante para pruebas. Después de entrenar los modelos, se evaluaron utilizando datos del intervalo de 2015 a 2016 para probar sus predicciones. Se observó que el modelo Gradient Boosting Regressor mostró una mayor precisión al ajustarse a los valores reales, mientras que el modelo K-Nearest Neighbors exhibió mayores variaciones con respecto a los valores reales de la humedad del suelo.

Tabla 3. Error cuadrático Medio y Error Absoluto Medio de los modelos entrenados.

Variable	Entrenamiento ECM	Dataset 2015-2016 ECM	Entrenamiento EAM	Dataset 2015-2016 EAM
Random Forest Regressor	**0.000401**	0.000736	**0.013026**	0.020937
Gradient Boosting Regressor	0.000574	**0.000460**	0.018599	**0.017459**
K-Nearest Neighbors	0.000525	0.000814	0.015645	0.022255
Regresión lineal múltiple	0.001033	0.000718	0.025296	0.021316

Esta diferencia se puede verificar en la Tabla 3, donde se presentan las métricas de error cuadrático medio (ECM) y error cuadrático absoluto (ECA). Se observa un menor error al utilizar el modelo Gradient Boosting Regressor para estimar la humedad del suelo con datos distintos a los utilizados durante el entrenamiento.

Una vez que obtuvimos el modelo mejor evaluado, Random Forest Regressor, procedimos a realizar predicciones de la humedad del suelo utilizando los datos meteorológicos de la estación meteorológica del Instituto Tecnológico de Morelia. Para esto, necesitábamos la fecha de la cual se extrajo el mes, así como las variables de temperatura (temp), punto de rocío (dwpt), humedad relativa (rhum) y precipitación (prcp). En la Figura 7 se muestran los valores generados por nuestro modelo para el intervalo de tiempo de enero de 2021 a mayo de 2023. Se observa un comportamiento lógico y coherente, en línea con lo esperado.

5. Discusión de los resultados

Después de revisar exhaustivamente los resultados obtenidos, queda claro que el uso de técnicas de aprendizaje automático ofrece un rendimiento prometedor en el ámbito de la agricultura de precisión para estimar variables de interés. Al analizar los resultados de los modelos entrenados, se destaca que el Gradient Boosting Regressor demostró ser el más efectivo; sin embargo, las otras técnicas también arrojaron resultados positivos.

Para el entrenamiento de nuestros modelos, se emplearon los hiperparámetros preconfigurados por la biblioteca Scikit-learn, lo que proporcionó resultados satisfactorios. No obstante, se reconoce la posibilidad de mejorar la estimación utilizando técnicas para obtener hiperparámetros más específicos según el problema en cuestión.

En cuanto a las variables de entrada seleccionadas para el modelo, se optó por aquellas consideradas como la mejor opción dadas las limitaciones de enfoque y recursos del proyecto. No obstante, los resultados abren la puerta a la exploración de diferentes variables y enfoques, ya que se logró una estimación exitosa de la humedad del suelo utilizando técnicas de aprendizaje automático. Dado que no contamos con sensores de humedad en todas las regiones de estudio, debido a los costos asociados, es valioso contar con técnicas que proporcionen estimaciones precisas de variables específicas.

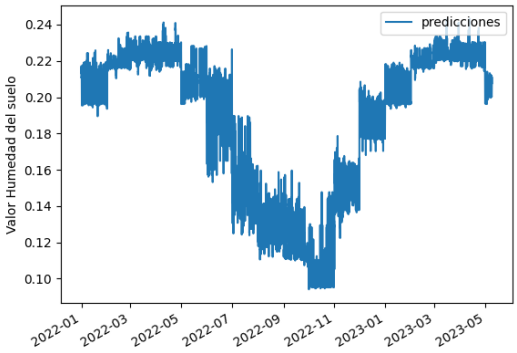


Fig. 4. Predicción de la humedad del suelo para dataset meteorológico del ITM.

Por esta razón, al tener solo datos meteorológicos y no mediciones directas de humedad del suelo, fue de interés realizar estimaciones basadas en modelos generados con aprendizaje automático a partir del comportamiento de dichas variables en diferentes ubicaciones. Este enfoque abre la oportunidad de experimentar con más variables o técnicas, así como de adquirir y tratar datos adicionales de interés en la agricultura. Además, una vez que se disponga de datos reales de medición de humedad del suelo en el área de Morelia, será posible compararlos con las estimaciones realizadas por el modelo implementado aquí.

6. Conclusiones

Este trabajo demuestra la aplicabilidad de las técnicas de aprendizaje automático para la estimación de la humedad del suelo, con potencial para futuras investigaciones y aplicaciones en el campo agrícola. Se exploraron y compararon diversas técnicas de aprendizaje automático para la estimación de la humedad del suelo en el contexto de la agricultura. A través del análisis de los resultados obtenidos, se pudo observar el potencial de estas técnicas para proporcionar estimaciones precisas y útiles en la planificación agrícola y la toma de decisiones.

Los modelos entrenados utilizando Random Forest Regressor, Gradient Boosting Regressor, K-Nearest Neighbors y Regresión Lineal Múltiple demostraron su eficacia para predecir la humedad del suelo utilizando variables meteorológicas como temperatura, punto de rocío, humedad relativa, precipitación y el mes correspondiente. Entre estos modelos, el Gradient Boosting Regressor se destacó por su menor error cuadrático medio y error absoluto medio, lo que sugiere su mayor capacidad predictiva en comparación con las otras técnicas evaluadas. Además, se observó que la inclusión del mes como característica en el entrenamiento de los modelos contribuyó significativamente a mejorar su rendimiento, lo que indica la importancia de considerar la variabilidad estacional en la estimación de la humedad del suelo.

El procesamiento y análisis de datos realizado en este estudio también proporcionó perspectivas importantes sobre la disponibilidad de información y la viabilidad de las técnicas de aprendizaje automático en entornos agrícolas donde los datos de sensores pueden ser limitados o costosos de adquirir.

Otra contribución este trabajo es la presentación y procesamiento del conjunto de datos creado por C. K. Gasch et al. Este conjunto de datos se ha transformado en un formato más accesible, facilitando su uso y análisis para futuras investigaciones en el campo de la estimación de la humedad del suelo y la agricultura de precisión. Al identificar y abordar los datos faltantes utilizando técnicas de interpolación de vecinos más cercanos y enriquecer el conjunto de datos con información meteorológica adicional, hemos creado una base sólida para análisis más detallados y comprensivos.

Agradecimientos. Los autores agradecen al Tecnológico Nacional de México por el apoyo brindado a través del proyecto 19476.24-P. Noel A. Zavala-Díaz agradece al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca de posgrado 2021-000018-02NACF-12060 brindada a través del Instituto Tecnológico de Morelia.

Referencias

1. Rasheed, M.W., Tang, J., Sarwar, A., Shah, S., Saddique, N., Khan, M.U., Imran-Khan, M., Nawaz, S., Shamshiri, R.R., Aziz, M., Sultan, M.: Soil Moisture Measuring Techniques and Factors Affecting the Moisture Dynamics: A Comprehensive Review. Sustainability, vol. 14, no. 18, pp. 11538 (2022). DOI: 10.3390/su141811538.
2. Palominos-Rizzo, T., Villatoro-Sánchez, M., Alvarado-Hernández, A., Cortés-Granados, V., Paguada-Pérez, D.: Estimación de la humedad del suelo mediante regresiones lineales múltiples en llano brenes, Costa Rica. Agronomía Mesoamericana, pp. 47872 (2022). DOI: 10.15517/am.v33i2.47872.
3. Lee, C., Sohn, E., Park, J., Dong, J.J.: Estimation of Soil Moisture using Deep Learning based on Satellite Data: A Case Study of South Korea. GIScience & Remote Sensing, vol. 56, no. 1, pp. 43–67 (2018). DOI: 10.1080/15481603.2018.1489943.
4. Ahmad, S., Kalra, A., Stephen, H.: Estimating Soil Moisture using Remote Sensing Data: A Machine Learning Approach. Advances in Water Resources, vol. 33, no. 1, pp. 69–80 (2010). DOI: 10.1016/j.advwatres.2009.10.008.
5. Prasad, R., Deo, C., Li, Y., Maraseni, T.: Soil Moisture Forecasting by a Hybrid Machine Learning Technique: ELM Integrated with Ensemble Empirical Mode Decomposition. Geoderma, vol. 330, pp. 136–161 (2018). DOI: 10.1016/j.geoderma.2018.05.035.
6. Wang, G., Hu, P., Lai, X., Xue, B., Fang, Q.: Root-Zone Soil Moisture Estimation based on Remote Sensing Data and Deep Learning. Environmental Research, vol. 212, pp. 113278 (2022). DOI: 10.1016/j.envres.2022.113278.
7. Kashyap, B., Kumar, R.: Sensing Methodologies in Agriculture for Soil Moisture and Nutrient Monitoring. IEEE Access, vol. 9, pp. 14095–14121 (2021). DOI: 10.1109/access.2021.3052478.
8. Khalyasmaa, A., Eroshenko, S.A., Chakravarthy, T.P., Gasi, V.G., Bollu, S.K.Y., Caire, R., Atluri, S.K.R., Karrolla, S.: Prediction of Solar Power Generation based on Random Forest Regressor Model. In: International Multi-Conference on Engineering, Computer and Information Sciences, pp. 0780–0785 (2019). DOI: 10.1109/sibircon48586.2019.8958063.
9. Gajula, A.K., Singamsetty, J., Dodda, V.C., Kuruguntla, L.: Prediction of Crop and Yield in Agriculture using Machine Learning Technique. In: 12th International Conference on Computing Communication and Networking Technologies, pp. 1–5 (2021). DOI: 10.1109/icccnt51525.2021.9579843.
10. Ponraj, A.S., Vigneswaran, T.: Daily Evapotranspiration Prediction using Gradient Boost Regression Model for Irrigation Planning. The Journal of Supercomputing, vol. 76, no. 8, pp. 5732–5744 (2019). DOI: 10.1007/s11227-019-02965-9.

11. Gasch, C.K., Brown, D.J., Campbell, C.S., Cobos, D.R., Brooks, E.S., Chahal, M., Poggio, M.: A Field-Scale Sensor Network Data Set for Monitoring and Modeling the Spatial and Temporal Variation of Soil Water Content in a Dryland Agricultural Field. Water Resources Research, vol. 53, no. 12, pp. 10878–10887 (2017). DOI: 10.1002/2017wr021307.
12. Gasch, C.K., Brown, D.J., Campbell, C.S., Cobos, D.R., Brooks, E.S., Chahal, M., Poggio, M.: A Field-Scale Sensor Network Data Set for Monitoring and Modeling the Spatial and Temporal Variation of Soil Water Content in a Dryland Agricultural Field. Water Resources Research, vol. 53, no. 12, pp. 10878–10887 (2017). DOI: 10.1002/2017wr021307.
13. Meteostat: Pullman/Sunshine. https://meteostat.net/es/station/KPUW0?t=2007-04-20/2007-04-27 (2024)

ISSN 1870-4069

Aplicación del escaneo 3D para la caracterización de lechugas en invernadero

Armando Figueroa-Martínez, Coral Martínez-Nolasco, Víctor M. Sámano-Ortega, José G. Zavala-Villalpando, Juan P. Aguilera-Álvarez

Tecnológico Nacional de México, Guanajuato, México

(M2403011, coral.martinez, victor.ortega, jg.zavala, juan.aguilera)@itcelaya.edu.mx

Resumen. En este artículo se explora la eficacia de la tecnología de escaneo 3D para dimensionar con precisión la lechuga en invernaderos. Se utilizó un escáner Revopoint para capturar de forma no destructiva datos tridimensionales de muestras de lechuga. Las mediciones lineales de puntos arbitrarios extraídos de los escaneos se compararon con mediciones físicas para validar la precisión. Además, se propuso un método novedoso para extraer áreas de superficie y medidas máximas a lo largo de los ejes de la lechuga escaneada se aporta una metodología que reduce la complejidad de uso y tiempo de puesta a punto. Los resultados indican que el escaneo 3D produce mediciones con una precisión promedio superior al 90% en comparación con las mediciones físicas. Este enfoque proporciona una alternativa confiable y precisa para dimensionar el tamaño de la lechuga en ambientes de invernadero, con implicaciones potenciales para la agricultura de precisión y la optimización de procesos agrícolas.

Palabras clave: Agricultura, escáner 3D, crecimiento, lechuga, caracterización, medición.

Application of 3D Scanning for Characterization of Lettuce in Greenhouses

Abstract. En this article, the effectiveness of 3D scanning technology for accurately sizing lettuce in greenhouses is explored. A Revopoint scanner was used to capture three-dimensional data from lettuce samples in a non-destructive manner. Linear measurements of arbitrary points extracted from the scans were compared with physical measurements to validate accuracy. Additionally, a novel method was proposed for extracting surface areas and maximum measurements along the axes of the scanned lettuce, providing a methodology that reduces complexity of use and setup time. The results indicate that 3D scanning produces measurements with an average accuracy exceeding 90% compared to physical measurements. This approach offers a reliable and precise alternative for sizing lettuce in greenhouse environments, with potential implications for precision agriculture and agricultural process optimization.

Keywords: Agriculture, 3D scanner, growth, lettuce, characterization, measure.

1. Introducción

Conforme pasa el tiempo, el mundo requiere una mayor cantidad de alimentos debido a ello, las técnicas en diversos sectores de producción han requerido mejoras y cambios. Dentro de las técnicas actuales, la agricultura protegida se encuentra entre los enfoques intensificados para la producción de alimentos, dado que es medible a través de diversas métricas, entre las cuales se encuentran insumos materiales, rendimientos por unidad de área, consumo de energía, emisiones de gases de efecto invernadero y costo [1]. También permite controlar de forma parcial o total el microclima y la protección de diferentes elementos ambientales, biológicos y climatológicos para mejorar la producción. Los cultivos de agricultura protegida generalmente logran un mayor rendimiento en comparación con cultivos convencionales [2]. Hoy en día, la agricultura protegida suele ser una base para diversos fines, por ejemplo, su uso en la investigación de la biología vegetal o con fines de comercialización.

Dentro de los cultivos más comunes están el tomate, la lechuga, el apio, el cilantro y el perejil [3]. El uso de este tipo de metodología para el cultivo ha sido acelerado; tan solo en 2020 se estima que la hidroponía en invernadero, que es una técnica representativa de agricultura protegida, tuvo un valor de 2.2 billones de dólares y su CAGR (Compound Annual Growth Rate) entre 2018 y 2021 fue de aproximadamente el 20.7% [4]. Sin embargo, la agricultura protegida aun presenta problemáticas en comparación con las técnicas de cultivo tradicional por ejemplo las lechugas cultivadas mediante técnicas de agricultura protegida poseen menor cantidad de antioxidantes esto se traduce en menor duración de forma óptima tras ser cortadas [5].

Por esta razón existe aún la necesidad de realizar contribuciones en el campo de la caracterización de los cultivos protegidos de manera que pueda reunirse más información y mejorar las técnicas de extracción de datos. En ámbito de la medición de crecimiento se tienen problemáticas con las técnicas actuales como el requerimiento de tiempo excesivo para la captura de datos [6]. Ante esta dificultad surgieron nuevos enfoques haciendo uso de nuevas tecnologías como la inteligencia artificial, en [6] se presenta un ejemplo de esta metodología. Algunos otros autores usan enfoques más convencionales como métodos lógicos que combinados con instrumentación como se describe en [7] también han logrado obtener buenos resultados.

Si bien estas herramientas permiten la identificación rápida y reducción del tiempo en la extracción de datos también aumentan en complejidad de conocimientos lo que dificulta la integración para especialistas de otras ramas distintas a la inteligencia artificial o procesamiento de imágenes. En el sentido del dimensionamiento/medición, el escaneo 3D ha demostrado ser una tecnología que entrega resultados con precisión y exactitud; es una herramienta esencial para los productores que necesitan una inspección dimensional precisa, imágenes virtuales, análisis e incluso fabricación de prototipos físicos [8].

El escaneo 3D aún tiene la posibilidad de explotar sus beneficios en áreas como la agricultura protegida, ya que las diferentes investigaciones que se han realizado suelen hacer uso de escaneo mediante cámaras de profundidad u otras tecnologías, y pocas veces se destaca el uso de algún escáner 3D comercial, abriendo así áreas de

Tabla 1. Ponderación de atributos.

	Precisión	**Velocidad de escaneo**	**Distancia de trabajo**	**Extracción de texturas**	**Precio**
Precisión	1/1	7/1	5/1	5/1	3/1
Velocidad de escaneo	1/7	1/1	3/1	1/5	1/5
Distancia de trabajo	1/5	1/3	1/1	1/3	1/5
Extracción de texturas	1/5	5/1	3/1	1/1	1/3
Precio	1/3	5/1	5/1	3/1	1/1

oportunidad como metodologías y análisis de parámetros para selección de escáner, comprobación de la fiabilidad de escáneres comerciales aplicados a cultivos y la virtualización de cultivos más realistas [9].

Por ello, el aporte de esta investigación es principalmente ejemplificar el uso de un escáner comercial para la caracterización de lechugas crecidas en invernadero, principalmente recolectando datos dimensionales y manteniendo una buena precisión y exactitud. A diferencia de otras investigaciones, no se utilizan sistemas basados en cámaras fotográficas que requieran un montaje y programación [10]. Además, se incluye la extracción de texturas con el escaneo 3D para una visualización más realista.

En comparativa con las técnicas usadas en otras investigaciones, el método propuesto también permite la portabilidad y la poca manipulación del cultivo al momento de realizar el escaneo. Además, es necesario resaltar que es una técnica no invasiva e "in situ", y que posibilita el envío de datos de manera inalámbrica haciéndose uso de baterías como fuente de alimentación.

Por otro lado, no existe la necesidad de un proceso de aprendizaje por parte del sistema, y el tiempo necesario para la captura es menor [11]. Debido a lo anterior el desarrollar esta propuesta permite una exploración inicial de métodos más eficientes y sencillos de utilizar en la caracterización de cultivos.

La caracterización de cultivos presentada es una alternativa importante dado existe una creciente demanda de alimentos y dificultades climáticas que impactan en las cosechas, requiriendo generar un conocimiento y mayor entendimiento de los cultivos para poder ejercer una mejora en el manejo de recursos o cuidado de los diferentes tipos de frutos y cultivos.

Así pues, puede concluirse que es relevante iniciar en la exploración del potencial del escaneo 3D al ser una herramienta que puede ayudar a entender mejor los cultivos y de esta manera de forma indirecta optimizar recursos agrícolas, mejorar la resiliencia ante condiciones climáticas adversas, aumentar la calidad y rendimiento de los cultivos, impulsar la investigación y desarrollo agrícola, y contribuir a la seguridad alimentaria en un contexto de creciente demanda y desafíos climáticos antes mencionada.

Finalmente, cabe destacar que dentro de los objetivos de la investigación están: proponer una alternativa de recolección y extracción de características dimensionales mediante el uso del escaneo 3D, reduciendo el tiempo de puesta en marcha del sistema y con mayor facilidad de uso. También se permite integrar modelos visualmente más

representativos de los cultivos reales para la virtualización y en formatos estandarizados que aporten a trabajos futuros en el ámbito de los gemelos digitales.

La estructura y contenido del artículo se divide en 5 secciones 1) En la introducción se presenta información sobre los cultivos en invernadero y la manera actual para caracterizarlos, principalmente en términos de dimensiones y formas, así como algunos trabajos relacionados y la diferencia con el presente en aporte. 2) Trabajos relacionados; se trata de manera más específica las tecnologías y métodos para la caracterización y dimensionamiento usados en otros trabajos afines. 3) Metodología: se expone principalmente la forma de selección del escáner utilizado, posteriormente se explica y muestra el diseño de pruebas realizadas. 4) Resultados: se muestran y concentran los datos obtenidos de diversos escaneos y pruebas efectuadas; con distintas finalidades.

2. Trabajos relacionados

Debido a que en el dimensionamiento de cultivos de lechuga se presentan diversas técnicas aplicadas, se realizó una exploración inicial. A continuación, se expone una recopilación de trabajos enfocados en esta área. Es fundamental destacar que estos trabajos no se limitan exclusivamente al ámbito de la agricultura protegida, sino que abordan el dimensionamiento y la representación tridimensional mediante diversas tecnologías. El propósito es llevar a cabo una primera aproximación a las herramientas disponibles en este campo. Las diversas investigaciones se agrupan acorde a sus herramientas o tecnologías características.

2.1. Uso del sensor kinect

De [12] se puede concluir que el artículo se centra en la medición de plantas utilizando el sensor Kinect como sistema de adquisición. La relevancia de este estudio radica en su enfoque en el proceso de calibración, ya que el sistema puede calibrarse de forma autónoma. Sin embargo, es importante destacar que este sistema, aunque útil, se basa en conceptos técnicos avanzados de álgebra lineal y cálculo vectorial. Esto implica que su replicación requiere una mayor inversión de tiempo en la curva de aprendizaje inicial. Además, se han reportado errores de hasta un 10% en las mediciones realizadas para verificar el sistema propuesto.

2.2. Uso de cámaras RealSense D415

Dentro de [13], se examina un huerto inteligente desde diversos aspectos económicos y morfológicos. En cuanto al dimensionamiento, destaca la capacidad de recopilar datos de manera autónoma en forma de nubes de puntos cada 15 minutos.

Se propone el uso de cámaras RealSense D415, las cuales emplean la técnica de visión estéreo para capturar imágenes de profundidad, RGB e IR. Todo el proceso de comunicación se realiza a través de computación en la nube utilizando la plataforma Azure, lo que implica la necesidad de acceso a internet, conocimientos de programación y manejo de plataformas en la nube. En sistemas con la misma base, se encuentra como limitante la incapacidad de generar nubes de puntos desde diferentes perspectivas debido a que las cámaras están fijadas en su lugar.

Tabla 2. Características escáner.

Atributo	Valor
Precisión	0.1 mm
Velocidad de escaneo	16 fps
Distancia de trabajo	400 - 1300 mm
Extraction de texturas	Si
Precio	799 dólares

Zhiyan et al. en [14] explora un objetivo y metodología distintos al concentrarse en una técnica individual de escaneo de cultivo de hojas de maíz. Para la reconstrucción del cultivo en 3D, se basan en capturar cada 45 grados de rotación para posteriormente usar un procesamiento de imágenes basado en el algoritmo de Otsu, que básicamente permite segmentar las imágenes. Una vez se realiza esto, se propone la fusión de imágenes por pares hasta lograr el modelo 3D. El autor reporta errores de hasta el 16% en las mediciones virtuales comparadas con las medidas manuales del cultivo físico.

2.3. Investigaciones basadas en técnicas de escaneo 3D

Dentro de las investigaciones que hacen uso de escáner 3D se encuentra [15]. En esta, se propone el uso de un escáner tipo Artec Space Spyder; no obstante, se limita al escaneo de la geometría, y el escaneo de textura o colores se realiza de manera independiente mediante captura de imágenes 2D. Posteriormente a esto, propone métodos de estimación de crecimiento. Los autores no presentan comparativas con mediciones físicas del objeto.

Las investigaciones basadas en técnicas de escaneo 3D han facilitado la recopilación de datos sobre diversas variables indirectas de los cultivos. Por ejemplo [16] en este trabajo se emplean cámaras de profundidad para realizar escaneos de suelo, centrándose en la clasificación de los escaneos según sus dimensiones de altura. Los datos obtenidos a través de estos escaneos son utilizados para estimar la erosión del suelo, lo que permite al investigador determinar su idoneidad para el cultivo. Es importante destacar que las pruebas se realizaron en pequeñas muestras de suelo y no en un entorno real.

Las distintas investigaciones involucran metodologías de puesta a punto tediosas para usuarios inexpertos requieren además en ocasiones una cantidad preliminar de datos para entrenamientos del sistema o el desarrollo de sistemas de automatización laboriosos y poco portables. Estas diversas problemáticas son reducidas con la propuesta realizada de caracterización mediante escáner 3D al ser un dispositivo de uso rápido, portable al poseer baterías, curva de aprendizaje menor comparada con otras propuestas. En términos de desventajas se encuentra pocas posibilidades de manejo de software del fabricante al no ser de código abierto, así como la sensibilidad ante malas iluminaciones para recoger correctamente la textura de los cultivos.

3. Metodología

El enfoque metodológico se orientó hacia la exploración de cultivos, utilizando lechugas como objeto de estudio y empleando la técnica de escaneo 3D a través de un

Fig. 1. Plataforma giratoria para escaneo.

escáner comercial. El principal propósito fue evaluar la fiabilidad del método propuesto y obtener mediciones precisas del área superficial, así como modelos 3D realistas que incluyan texturas. La metodología se divide en dos secciones principales. En primer lugar, se ejecutó una comparación entre dos modelos comerciales de escáneres, junto con la definición de los requisitos básicos para las pruebas planificadas.

La segunda fase de la metodología comprende la descripción detallada de las pruebas, dividida principalmente en dos procesos. El primer proceso consistió en la evaluación de la precisión del método mediante la comparación con mediciones físicas de las lechugas escaneadas. Posteriormente, se extraen mediciones automatizadas de áreas superficiales mediante programación de alto nivel, como un enfoque complementario.

3.1. Selección del scanner y aditamentos

Para la selección del escáner, se aplicó el método analítico jerárquico como marco de referencia, determinando la importancia de los parámetros mediante su ponderación (Ver Tabla 1) para seleccionar el parámetro clave.

Los atributos considerados para este propósito fueron: precisión, velocidad de escaneo, distancia de trabajo, capacidad de extracción de texturas y precio. Los resultados del análisis de la Tabla 1 se muestran en la ecuación 1. De ella se concluye que el atributo más determinante para la selección del escáner será la precisión, seguido del precio:

$$\begin{matrix}\text{Precisión}\\ \text{Velocidad de escaneo}\\ \text{Distancia de trabajo}\\ \text{Extracción de texturas}\\ \text{Precio}\end{matrix} = \begin{bmatrix}0.4734\\ 0.0741\\ 0.0522\\ 0.1462\\ 0.2541\end{bmatrix}. \quad (1)$$

Tras el análisis de los atributos, se consideraron dos modelos de escáner para la comparación: el Range 2 y el Pop 2, ya que ambos son escáneres manuales de bajo costo. La decisión se basó principalmente en la precisión; se observó que el Range 2

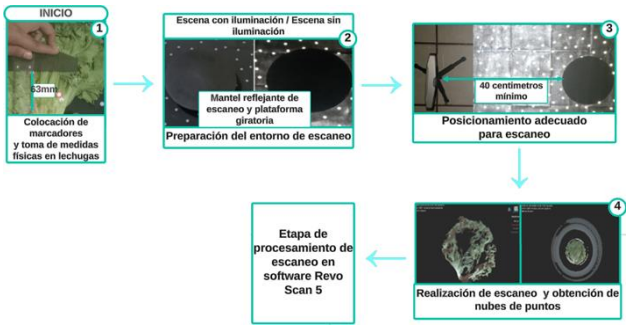


Fig. 1. Proceso de la etapa física de las pruebas.

Fig. 2. Proceso de la etapa de procesamiento de escaneos.

tiene una precisión de 0.1 mm [17], mientras que el Pop 2, según el fabricante, tiene una precisión ligeramente inferior de 0.15 mm.

Es relevante destacar que este parámetro, al ser evaluado en términos cuantitativos, es casi el doble de importante que el segundo parámetro (precio), tal como se observa en la ecuación 1. Los atributos del escáner seleccionado se concentran en la Tabla 2 [18]. Como siguiente paso, se requirió la incorporación de una plataforma giratoria (Ver Fig. 1). En este caso, la selección se centró en una plataforma que pudiera funcionar con un consumo mínimo de energía y de manera inalámbrica. La opción seleccionada tiene una velocidad de 7 rpm y un diámetro de 20 cm.

3.2. Diseño de pruebas

El diseño experimental se estructuró en dos fases distintas. Durante la primera etapa, se tomaron mediciones tanto físicas como virtuales con el propósito de evaluar la confiabilidad del escáner.

En la segunda fase, se elaboró un script destinado al análisis de las áreas superficiales de los escaneos, así como a la medición de las dimensiones máximas en los tres ejes cartesianos (x, y, z).

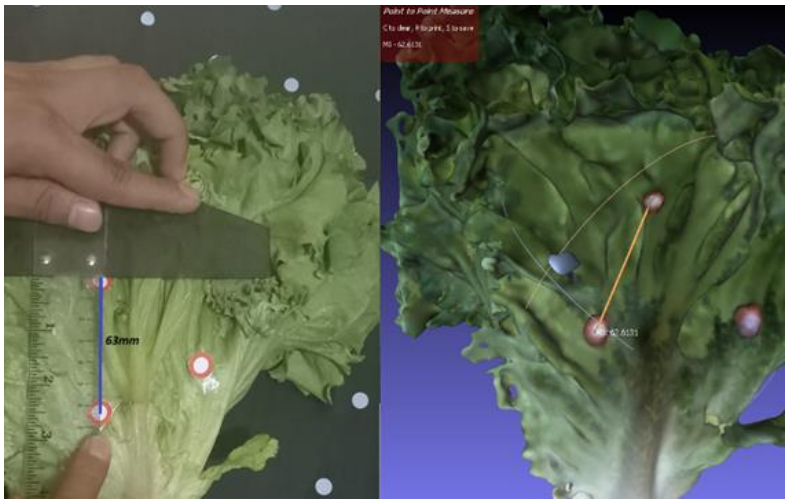


Fig. 4. Comparativa de mediciones físicas vs virtuales.

Pruebas de fiabilidad. Para la etapa de verificación de la fiabilidad, se realizaron escaneos de un conjunto de cinco lechugas. Este proceso se dividió en varios pasos: la etapa física y la etapa de procesamiento de los escaneos. El primer conjunto de pasos comenzó con la colocación de marcadores utilizados como referencia tanto para las mediciones físicas como para las virtuales.

Posteriormente, se procedió a preparar el entorno de escaneo, garantizando la ausencia de objetos que pudieran interferir y manteniendo una iluminación tenue para asegurar resultados óptimos. Además, se estableció una distancia mínima de aproximadamente 40 centímetros entre la plataforma giratoria y el escáner (ver Fig. 2).

La etapa de procesamiento de los escaneos se lleva a cabo en una serie de pasos estructurados para llegar desde la nube de puntos hasta los archivos obj y ply dichos archivos son usados comúnmente para almacenar información de figuras y geometrías 3D la diferencia radica en que los ply son más compactos (ver Fig. 3). La ejecución de los pasos se lleva a cabo en el software Revo Scan 5 dichos pasos se describen a continuación:

1. **Eliminación de puntos basura:** Esta etapa se enfoca en eliminar los datos que no son relevantes para el objeto de escaneo.
2. **Fusión de nube de puntos:** Esta operación permite combinar los datos del escaneo para generar una nube de puntos más precisa.
3. **Aislamiento de puntos:** El software detecta puntos aislados dentro de los conjuntos de puntos, lo que facilita la eliminación de datos que no pertenecen al modelo y no son visibles a simple vista.
4. **Detección de superposición**: Consiste en identificar puntos superpuestos y eliminarlos del modelo.
5. **Suavizado:** Esta operación tiene como objetivo eliminar el ruido y las imperfecciones locales del modelo.

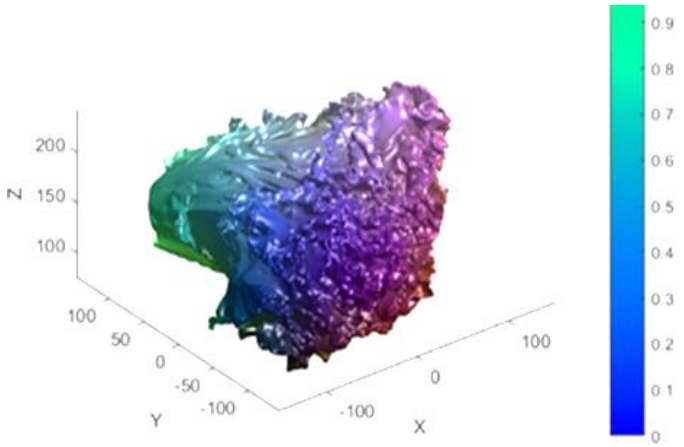


Fig. 5. Gráfico 3D de escaneo.

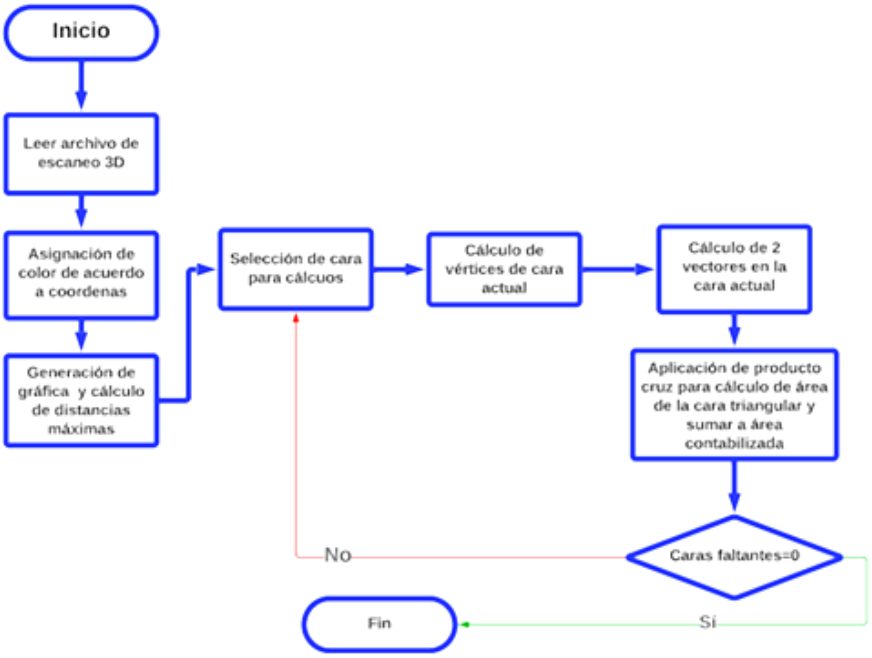


Fig. 6. Diagrama de flujo de estimación de área superficial

6. **Simplificación:** Se reduce el número de datos para mejorar el rendimiento de renderizado y la velocidad de procesamiento.

7. **Fusión de nubes de puntos:** Se refiere al proceso de combinar dos nubes de puntos procesadas en una sola para obtener un modelo mejorado.

8. **Construcción de la malla:** Esta etapa consiste la creación de una malla continua a partir de las nubes de puntos obtenidas.

Una vez finalizados los pasos de procesamiento del escaneo, el modelo fue exportado en formato ply y obj. Estos formatos de usaron posteriormente para realizar mediciones virtuales mediante el software MeshLab y compararlas con las mediciones físicas (Ver Fig. 4).

Medición de área superficial. La medición del área superficial se realizó utilizando los archivos obtenidos de los escaneos 3D. Esto implicó la estimación de áreas superficiales y su representación en un gráfico tridimensional, lo que facilitó la extracción de las medidas máximas en los diferentes ejes (Ver Fig. 5). El procedimiento para realizar estas mediciones se llevó a cabo mediante un script de MATLAB. Este script se encargó de asignar colores a los diferentes vértices y caras según las coordenadas en cada uno de los ejes, mientras realizaba el cálculo de las distancias máximas en los ejes coordenados. El siguiente proceso consiste en iterar en cada una de las caras y vértices para obtener una sumatoria y así calcular el valor del área superficial (Ver Fig. 6).

4. Resultados

Los resultados de las pruebas de fiabilidad comprenden 5 escaneos, de los cuales se extrajeron 3 medidas de referencia, las cuales fueron comparadas con mediciones virtuales (Ver Tabla 3). La mayoría de las mediciones realizadas presentan una exactitud superior al 90%.

Los valores de exactitud promedio para todos los escaneos se sitúan por encima del 95%. Es importante señalar que las medidas están expresadas en milímetros y fueron definidas como puntos de referencia de manera arbitraria. Los resultados del área superficial, junto con las distancias máximas entre puntos en cada eje, se presentan de manera concisa en la tabla 4.

No obstante, se observó una variación en los valores de las coordenadas dependiendo de la posición en la que se coloque el escaneo 3D al momento de la importación. Aunque los datos exhiben coherencia entre los valores del área y las medidas en los diferentes ejes, es esencial destacar la necesidad de aplicar métodos físicos en futuros trabajos para verificar estas mediciones de manera precisa.

Por último, también es visible que los resultados virtuales son más realistas en términos de texturas y colores al realizarse el escaneo 3D, como se muestra en la Figura 7.

5. Conclusiones y trabajos futuros

En este artículo se presenta una alternativa para la extracción de características de cultivos de lechugas, la cual demostró tener un acercamiento significativo a los valores reales en términos dimensionales. Con esto, se concluye que el escaneo 3D es una técnica que merece ser explorada en mayor profundidad, aplicada al estudio de la agricultura protegida. Además, se ejemplificó que es posible obtener estimaciones de áreas superficiales mediante este método.

Tabla 3. Resultados comparativos de mediciones reales y virtuales.

No. Lechuga	Medidas reales (mm)	Medidas virtuales (mm)	Exactitud %	Exactitud % Promedio
1	63	62	98.42	92.56
	62	66	93.55	
	56	64	85.72	
2	66	66.5	99.25	98.05
	107	108	99.07	
	48	50	95.84	
3	63	65	96.83	96.09
	59	63	93.23	
	47	45	95.75	
4	65	67	96.93	97.84
	61	60	98.37	
	56	55	98.22	
5	49	51	95.72	97.38
	60	59	98.34	
	104	106	98.08	

Tabla 4. Resultados de área superficiales y medidas máximas en ejes.

No. Lechuga	Área superficial cm^2	Valor máximo de grafica (x, y, z) cm
1	3465.5	30.5
		28.9
		16.6
2	4397.5	34.2
		38.5
		20.6
3	5303.1	36
		27.4
		23.5
4	3747.2	34.7
		25.12
		22.9
5	3643.3	30.5
		26.8
		15.9

Sin embargo, es crucial llevar a cabo más investigaciones en este campo para corroborar la repetibilidad de las pruebas y métodos propuestos, incluso utilizando diferentes equipos de escaneo y durante períodos más prolongados de la vida de los cultivos y en diversas etapas de su crecimiento. Es importante señalar que una vertiente del presente trabajo que queda pendiente de comprobación en términos de exactitud es la estimación del área superficial y las mediciones de longitudes máximas en los ejes. Estas últimas requieren métodos invasivos para poder ser realizadas de manera física, razón por la cual podrían plantearse en investigaciones a futuro.

Por otra parte, las ventajas que se demuestran en la metodología de este artículo están la capacidad de extracción de características de cultivos sin necesidad de una curva de

Fig. 7. Resultados de modelo virtualizado realista.

aprendizaje elevada, la recolección de texturas más acordes a los cultivos físicos, la poca infraestructura y equipo comparado con otras metodologías, y finalmente que es un procedimiento fiable en mediciones lineales y sin necesidad de ser invasivo para el cultivo a escanearse.

Siguiendo en el contexto de trabajos futuros, este artículo pretende ser un punto de partida para el desarrollo de gemelos digitales en formato 3D, enfocados en mejorar la gestión y generación de conocimientos de la agricultura protegida, principalmente de lechugas. Esto incluiría no solo el monitoreo en una etapa de crecimiento de los cultivos, sino también la caracterización del crecimiento desde etapas tempranas hasta la cosecha, permitiendo así desarrollar una base de información que contribuya a la mejora de la agricultura protegida.

Referencias

1. Villagrán, E., Romero-Perdomo, F., Numa-Vergel, S., Galindo-Pacheco, J.R., Salinas-Velandia, D.A.: Life Cycle Assessment in Protected Agriculture: Where are we Now, and Where Should we Go Next? Horticulturae, vol. 10, no. 1, pp. 15–49 (2023). DOI: 10.3390/horticulturae10010015.
2. Negra, C., Pratt, L., Manuel-Ortega, J., House, K., Qadir, U.: Protected Agriculture: Mexico the Climate Bonds Standard and Certification Scheme's Protected Agriculture Criteria for Mexico (2019)
3. López-Elías, J.: La producción hidropónica de cultivos. Idesia (Arica), vol. 36, no. 2, pp. 139–141 (2018). DOI: 10.4067/s0718-34292018005000801.
4. Sanchaya, S., Saritha, M., Bhaskar, M., Leul, T., Anushri, T., Subham, A., Anish, J.: Research on Hydroponics Farming. International Journal of Research Publication and Reviews, vol. 4, no. 4, pp. 825–832 (2023)
5. Lei, C., Engeseth, N.J.: Comparison of Growth Characteristics, Functional Qualities, and Texture of Hydroponically Grown and Soil-Grown Lettuce. LWT, vol. 150, pp. 111931 (2021). DOI: 10.1016/j.lwt.2021.111931.
6. Lei, C., Engeseth, N.J.: Comparison of Growth Characteristics, Functional Qualities, and Texture of Hydroponically Grown and Soil-Grown Lettuce. Lebensmittel-Wissenschaft und -Technologie, vol. 150, pp. 111931 (2021). DOI: 10.1016/j.lwt.2021.111931.

7. Wang, Y., Wu, M., Shen, Y.: Identifying the Growth Status of Hydroponic Lettuce based on YOLO-Efficientnet. Plants, vol. 13, no. 3, pp. 372–384 (2024). DOI: 10.3390/plants13030372.
8. Ma, Y., Zhang, Y., Jin, X., Li, X., Wang, H., Qi, C.: A Visual Method of Hydroponic Lettuces Height and Leaves Expansion Size Measurement for Intelligent Harvesting. Agronomy, vol. 13, no. 8, pp. 1996–2013 (2023). DOI: 10.3390/agronomy13081996.
9. Javaid, M., Haleem, A., Pratap-Singh, R., Suman, R.: Industrial Perspectives of 3D Scanning: Features, Roles and its Analytical Applications. Sensors International, vol. 2, pp. 100114 (2021). DOI: 10.1016/j.sintl.2021.100114.
10. Wada, K.C., Hayashi, A., Lee, U., Tanabata, T., Isobe, S., Itoh, H., Maeda, H., Fujisako, S., Kochi, N.: A Novel Method for Quantifying Plant Morphological Characteristics using Normal Vectors and Local Curvature Data Via 3D Modelling—A Case Study in Leaf Lettuce. Sensors, vol. 23, no. 15, pp. 6825 (2023). DOI: 10.3390/s23156825.
11. Xiang, L., Wang, D.: A Review of Three-Dimensional Vision Techniques in Food and Agriculture Applications. Smart Agricultural Technology, vol. 5, pp. 100259 (2023). DOI: 10.1016/j.atech.2023.100259.
12. Koyama, K.: Leaf Area Estimation by Photographing Leaves Sandwiched between Transparent Clear File Folder Sheets. Horticulturae, vol. 9, no. 6, pp. 709–728 (2023). DOI: 10.3390/horticulturae9060709.
13. Sun, G., Wang, X.: Three-Dimensional Point Cloud Reconstruction and Morphology Measurement Method for Greenhouse Plants based on the Kinect Sensor Self-Calibration. Agronomy, vol. 9, no. 10, pp. 596–618 (2019). DOI: 10.3390/agronomy9100596.
14. Petropoulou, A.S., van-Marrewijk, B., de-Zwart, F., Elings, A., Bijlaard, M., van-Daalen, T., Jansen, G., Hemming, S.: Lettuce Production in Intelligent Greenhouses—3D Imaging and Computer Vision for Plant Spacing Decisions. Sensors, vol. 23, no. 6, pp. 2929 (2023). DOI: 10.3390/s23062929.
15. Ma, Z., Wan, H., Gan, X.: Research on Crop 3D Model Reconstruction based on RGB-D Binocular Vision. Scientific Programming, vol. 2023, pp. 1–10 (2023). DOI: 10.1155/2023/5974981.
16. Shadrin, D., Somov, A., Podladchikova, T., Gerzer, R.: Pervasive Agriculture: Measuring and Predicting Plant Growth using Statistics and 2D/3D Imaging. In: Proceedings of the IEEE International Instrumentation and Measurement Technology Conference, pp. 1–6 (2018). DOI: 10.1109/i2mtc.2018.8409700.
17. Kříž, M., Linda, M., Svatoš, J., Hromasová, M.: Application of 3D Cameras in Agriculture when Evaluating the Quality of Soil Tillage. Research in Agricultural Engineering, vol. 62, no. 2, pp. 39–49 (2016). DOI: 10.17221/4/2014-rae.
18. REVOPOINT: RANGE 2 3D Scanner. http://global.revopoint3d.com/pages/handheld-3d-scanner-range2 (2024)
19. REVOPOINT: Revopoint POP 2. http://www.revopoint3d.com/pages/face-3d-scanner-pop2 (2024)

ISSN 1870-4069

Lógica difusa y el manifiesto ágil: Innovación en la medición de agilidad en el desarrollo de software

Sergio Octavio Rosales Aguayo[1,3], Pedro Damián Reyes[2], José Román Herrera Morales[2], Ricardo Acosta Díaz[2]

[1] Tecnológico Nacional de México, Jalisco, México

[2] Universidad de Colima, México

[3] Universidad Davinci, México

sergio.ra@cdguzman.tecnm.mx, {damian, rherrera, acosta}@ucol.mx

Resumen. Las metodologías ágiles son esenciales en el desarrollo de software; sin embargo, cuantificar la agilidad sigue siendo un desafío. Este estudio presenta un enfoque que utiliza la lógica difusa, reconocida por manejar la imprecisión y ambigüedad, para medir la agilidad en el desarrollo de aplicaciones. Se establece una matriz de correspondencia y una distribución de frecuencias que informa un marco de medición de agilidad basado en lógica difusa. La metodología propuesta se detalla en etapas secuenciales y se representa esquemáticamente con diagramas de actividad UML. El sistema de inferencia difuso, creado con Python, integra funciones de membresía predefinidas y 81 reglas. Además, se desarrolló un instrumento de medición aplicado a equipos de desarrollo frontend que provee datos para el sistema difuso, resultando en métricas de agilidad cuantificables. La eficacia de esta metodología se demuestra en estudios de caso reales, mostrando una alineación con la agilidad percibida y proporcionando una métrica objetiva y matizada de agilidad. Este estudio sienta las bases para futuras exploraciones y el refinamiento continuo de prácticas ágiles, proponiendo el modelo basado en lógica difusa como un estándar potencial para evaluar la agilidad en proyectos de software.

Palabras clave: Lógica difusa, desarrollo ágil, evaluación de la agilidad en desarrollo de software.

Fuzzy Logic and the Agile Manifesto: Innovation in Measuring Agility in Software Development

Abstract. Agile methodologies are essential in software development; however, quantifying agility remains a challenge. This study presents an approach that uses fuzzy logic, known for handling imprecision and ambiguity, to measure agility in application development. A correspondence matrix and frequency distribution

are established that inform an agility measurement framework based on fuzzy logic. The proposed methodology is detailed in sequential stages and represented schematically with UML activity diagrams. The fuzzy inference system, created with Python, integrates predefined membership functions and 81 rules. Additionally, a measurement instrument was developed applied to frontend development teams that provides data for the fuzzy system, resulting in quantifiable agility metrics. The effectiveness of this methodology is demonstrated in real case studies, showing alignment with perceived agility and providing an objective and nuanced metric of agility. This study lays the foundation for future exploration and continuous refinement of agile practices, proposing the fuzzy logic-based model as a potential standard for evaluating agility in software projects.

Keywords: Fuzzy logic, agile development, agility evaluation in software development.

1. Introducción

En el dinámico mundo del desarrollo de software, la agilidad es más que una metodología; es una necesidad vital para la supervivencia y el éxito. El Manifiesto por el Desarrollo Ágil de Software [1], ha inspirado a equipos de todo el mundo a adoptar prácticas que promueven la adaptabilidad y la respuesta eficiente al cambio [2]. Sin embargo, medir la agilidad de una manera que capture su esencia multifacética sigue siendo un desafío [3], la lógica difusa, con su capacidad intrínseca para manejar la imprecisión y la ambigüedad, se postula como una solución prometedora. Este estudio presenta un modelo innovador que fusiona la lógica difusa con los principios ágiles y cuatro metodologías populares para medir la agilidad en el desarrollo de aplicaciones front end.

2. Revisión de la literatura

2.1. Desarrollo de software

El desarrollo de software implica la creación, diseño, despliegue y soporte de aplicaciones y sistemas software. Este proceso puede adoptar diversas metodologías que estructuran, planifican y controlan el desarrollo de un sistema de información. Entre los modelos tradicionales se encuentran:

- Modelo Cascada: donde el proceso sigue secuencias lineales y fases estancas, como requisitos, diseño, implementación, verificación y mantenimiento [4].
- Modelo Espiral: que combina elementos iterativos con evaluaciones de riesgo en cada ciclo [5].
- El proceso de Desarrollo de Software Unificado: un enfoque iterativo que enfatiza la adaptabilidad y la entrega de software operativo en cada iteración [6].

Los modelos anteriores fueron considerados pesados en términos de documentación extensa y tiempos largos de análisis y diseño. En respuesta a las limitaciones de estos

Fig. 1. Metodología de trabajo.

modelos tradicionales, surge el Manifiesto por el Desarrollo Ágil de Software [1], que propone valores y principios orientados a la adaptabilidad y la entrega continua. Las metodologías ágiles como Scrum, Kanban y Extreme Programming (XP) se enfocan en la colaboración, el cliente como centro y la capacidad de adaptarse a cambios rápidos [7-8]. Finalmente, la medición de la agilidad en el desarrollo de software se ha comenzado a abordar mediante técnicas avanzadas como la inteligencia artificial, el aprendizaje automático y la lógica difusa, ofreciendo métodos más sofisticados y adaptativos para evaluar y mejorar las prácticas de desarrollo ágil [9-10].

2.2. Metodologías ágiles comunes

Por otro lado, las metodologías ágiles se fundamentan en una serie de principios y valores que promueven un enfoque de desarrollo de software más flexible y orientado a las personas, Meyer [11] menciona que las metodologías de mayor uso son: Scrum desarrollada por Jeff Sutherlan y Ken Schwaber [7], Lean Software desarrollada por Mary Poppendieck [12], Crystal desarrollada por Alistair Cockburn [13] y Extreme Programming (XP) desarrollada por Kent Beck [14].

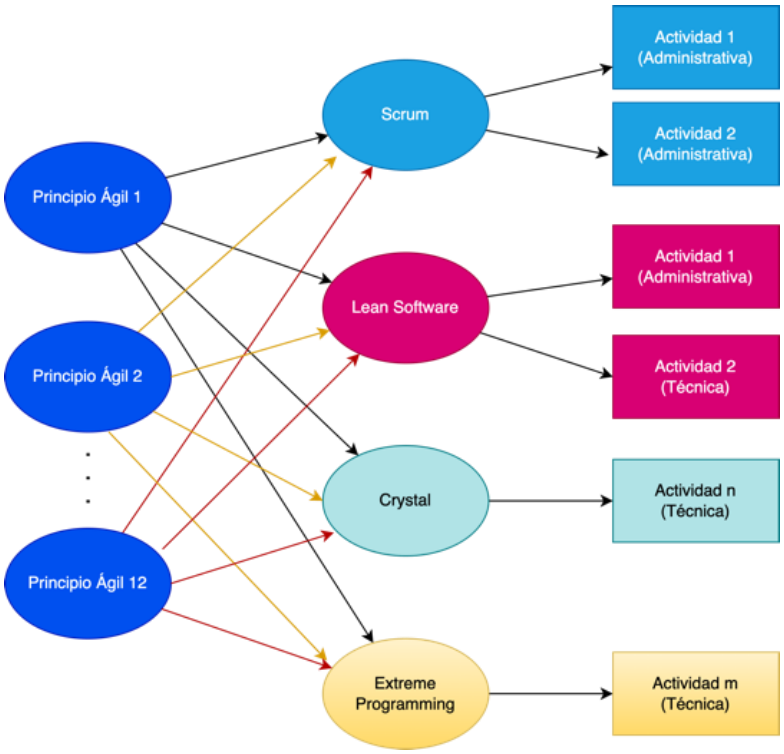


Fig. 1. Correspondencia entre principios ágiles, metodologías y actividades.

2.3. Lógica difusa

La lógica difusa es una rama de la Inteligencia Artificial que permite analizar información entre lo falso y lo verdadero [15]. El objetivo principal de la lógica difusa es crear un sistema basado en el comportamiento y pensamiento humano, con lo cual se pueden resolver problemas de ciencias actuariales, administración y gestión de empresas, química, ciencias de la tierra, ecología y ciencias ambientales, economía, ingeniería (civil, industrial, mecánica, nuclear, etc.), ergonomía, tecnología de la información, medicina, ciencias sociales, telecomunicaciones, gestión del tráfico [16]. La lógica difusa está basada en teoría de conjuntos difusos, la cual es una generalización de la teoría clásica de conjuntos [17].

2.4. Agilidad y lógica difusa

Los fundamentos de la agilidad y la lógica difusa tienen raíces profundas tanto en la teoría como en la práctica. El MDAS, ha sido un catalizador para la transformación de procesos de desarrollo de software.

La lógica difusa, propuesta por Lotfi Zadeh [18], ha evolucionado hasta convertirse en una herramienta poderosa para el modelado de decisiones en situaciones inciertas.

Tabla 1. Selección de principios, la actividad correspondiente y la metodología a la que pertenece.

Principio	Actividad	Metodología
Principio Ágil 1	Frequent Integration	Crystal Method
	Small Releases	Extreme Programming
Principio Ágil 2	Refactoring	Lean Software
	Frequent Delivery	Crystal Method
	TDD	Extreme Programming
Principio Ágil 3	Frequent Delivery	Crystal Method
	Small Releases	Extreme Programming
Principio Ágil 7	Frequent Delivery	Crystal Method
	Continuous Integration	Extreme Programming
Principio Ágil 9	Refactoring	Lean Software
	Refactoring	Extreme Programming
Principio Ágil 10	Simple Design	Extreme Programming

Tabla 2. Distribución de frecuencia de actividades ágiles de tipo técnico.

Actividad	Frecuencia	Porcentaje
Continuous Integration	2	15%
Frequent Delivery	5	38%
Refactoring	4	31%
Testing	1	8%
Simple Design	1	8%

Los trabajos de Beck y colaboradores [14] sobre prácticas de desarrollo ágil y de Ross sobre lógica difusa [19], permiten integrar ambas áreas del conocimiento.

3. Metodología

La metodología que se propone en este artículo se construyó a partir de una revisión exhaustiva de la literatura sobre el Manifiesto Ágil y cuatro metodologías ágiles prominentes. Se desarrolló una matriz de correspondencia para identificar y sintetizar los principios y metodologías que fundamentan la práctica ágil.

Posteriormente, se clasificaron las actividades de desarrollo en administrativas y técnicas para enfocar la evaluación de agilidad en aquellas actividades que directamente contribuyen a la entrega de valor.

Se elaboró una distribución de frecuencias para las actividades identificadas con el fin de determinar su relevancia en la práctica ágil actual. Con estos datos, se diseñó un marco de medición que emplea la lógica difusa para cuantificar la agilidad, permitiendo una evaluación más flexible y representativa de las prácticas ágiles en diferentes contextos. Esta metodología se representa de forma esquemática en la Fig. 1.

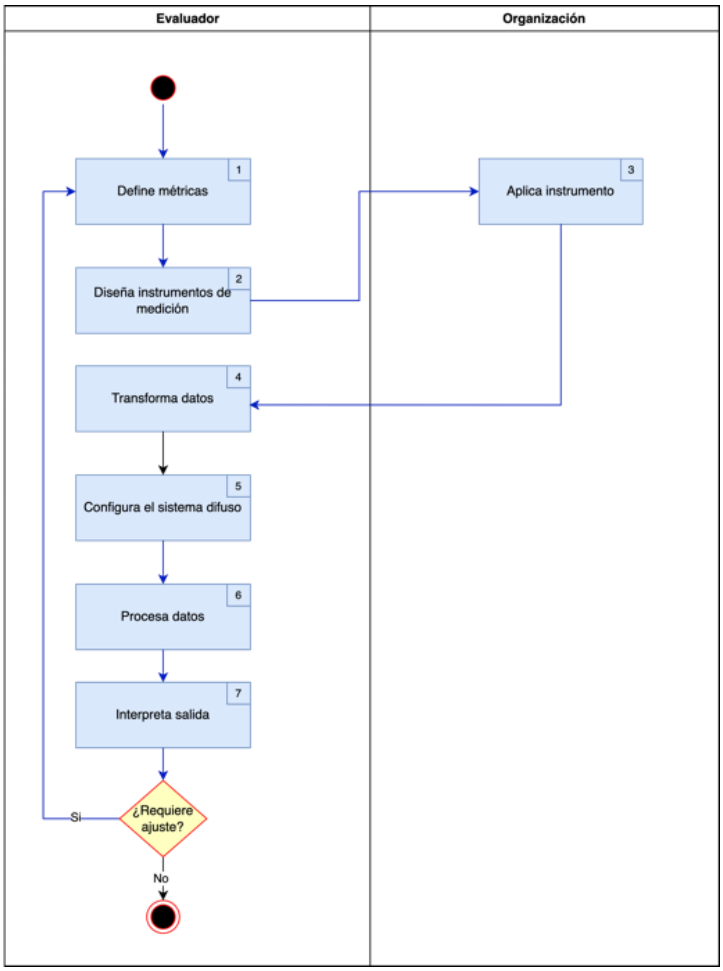


Fig. 3. Metodología para medir el nivel de agilidad de desarrollo.

La Fig. 2, muestra la relación entre los doce principios ágiles fundamentales, como se definen en el Manifiesto Ágil, y cuatro metodologías ágiles prominentes: Scrum, Lean Software, Crystal y Extreme Programming. Se distinguen dos categorías de actividades de desarrollo de software: administrativas y técnicas.

Las actividades administrativas son aquellas que se relacionan con la gestión del proyecto, mientras que las técnicas se refieren a las prácticas directamente involucradas en la creación del software. La Tabla 1 presenta una asociación detallada entre seis principios ágiles seleccionados, las actividades de desarrollo de software clave y las metodologías ágiles que típicamente las implementan.

Los datos se organizan en tres columnas: la primera lista los principios ágiles numerados; la segunda, las actividades de desarrollo de software relacionadas con ese principio; y la tercera, las metodologías ágiles que adoptan esas actividades. Esto proporciona un marco para comprender cómo las metodologías ágiles pueden ser

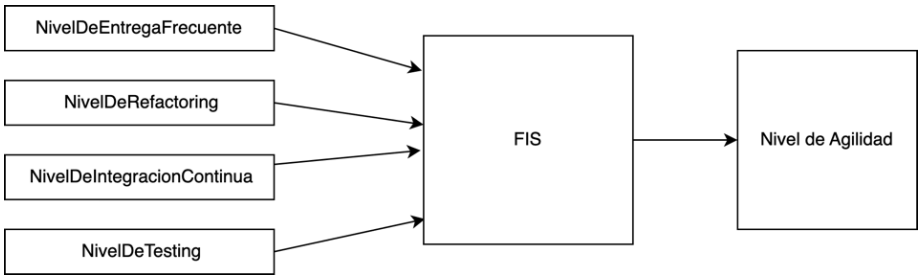


Fig. 2. Sistema de inferencia difuso con 4 variables de entrada.

Tabla 3. Variables de entrada.

Actividad	Variable	Relevancia	Términos lingüísticos
Frequent Delivery (Entrega Frecuente)	NivelDeEntregaFrecuente (NEF)	38%	Bajo (BNE)
			Medio (MNE)
			Alto (ANE)
Refactoring+Simple Design (Refactorización)	NivelDeRefactoring (NR)	39%	Bajo (BNR)
			Medio (MNR)
			Alto (ANR)
Continuous Integration (Integración Continua)	NivelDeIntegracionContinua (NIC)	15%	Bajo (BNIC)
			Medio (MNIC)
			Alto (ANIC)
Testing (Pruebas)	NivelDeTesting (NT)	8%	Bajo (BNT)
			Medio (MNT)
			Alto (ANT)

Tabla 4. Salida del Sistema Difuso.

Variable lingüística	Términos lingüísticos
Nivel de agilidad	Nivel de Agilidad BAJO (NAB)
	Nivel de Agilidad Medio (NAM)
	Nivel de Agilidad Alto (NAA)

aplicadas selectivamente para reforzar principios específicos en un entorno de desarrollo de software.

En referencia a la Tabla 1, se llevó a cabo un análisis para discernir y consolidar las actividades del desarrollo ágil.

Se observó la presencia de actividades recurrentes y otras que, pese a tener denominaciones distintas, compartían esencia y propósito. Esta evaluación culminó con la identificación de cinco actividades fundamentales que representan prácticas centrales en metodologías ágiles: 'Continuous Integration', 'Frequent Delivery', 'Refactoring', 'Testing', y 'Simple Design'. Estas actividades conforman la base para la posterior elaboración de la propuesta de la metodología de medición de agilidad.

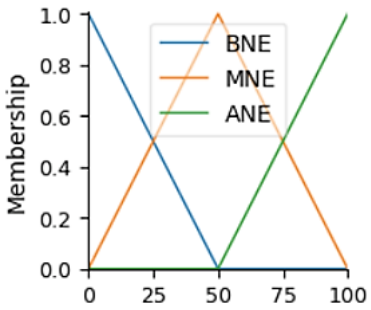


Fig. 5. Funciones de membresía para las variables de entrada.

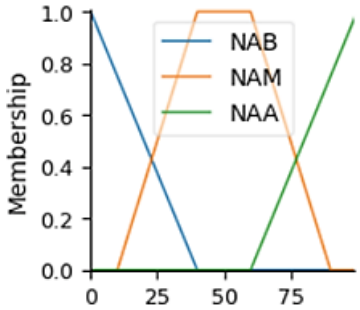


Fig. 6. Funciones de membresía de NivelDeAgilidad.

Tabla 5. Nivel de Agilidad Empresa de las empresas E001, E002, E003, E004, E005 y E006.

Variable Lingüística	**E001**	**E002**	**E003**	**E004**	**E005**	**E006**
Nivel de Entrega Frecuente	56	63	55	38	15	20
Nivel de Refactoring	34	65	65	46	11	28
Nivel de Integración Continua	80	73	60	60	16	18
Nivel de Testing	68	90	58	36	17	22
Nivel de Agilidad Calculado	**NAM (55.07)**	**NAM (57.07)**	**NAM (53,75)**	**NAM (49.07)**	**NAB (35.56)**	**NAB (39.83)**

La Tabla 2, muestra un análisis cuantitativo de la frecuencia con la que se implementan ciertas actividades esenciales dentro de las metodologías ágiles, según se ha derivado de la Tabla 1. Las actividades se enumeran en la columna izquierda, mientras que la columna central refleja el número de veces que cada actividad es referenciada o aplicada dentro del contexto de la investigación actual, la columna de la derecha representa el porcentaje en términos de relevancia.

Frequent Delivery encabeza la lista con un 38% en términos de relevancia, seguido de Refactoring con un 31%. A estos le siguen Continuous Integration con un 15% y Simple Design y Testing, ambos con un 8%. Por lo tanto, la conclusión que se extrae es que estas actividades deben ser consideradas como prioritarias en cualquier implementación de metodologías ágiles, dada su significativa influencia en la programación y en el alineamiento con los principios ágiles del MDAS.

4. Metodología de medición propuesta

Con estos antecedentes, se ha diseñado una metodología estructurada para la evaluación de la agilidad en el desarrollo de aplicaciones front end utilizando la lógica difusa. Esta metodología se describe a través de un Diagrama de Actividad de UML

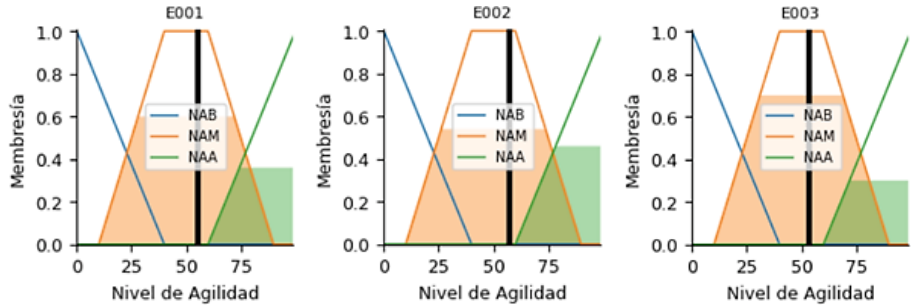


Fig. 3 Niveles de Agilidad de las empresas E001, E002 y E003.

representado en la Fig. 3. El diagrama ofrece una representación visual y sistemática del proceso de evaluación, mismo que se detalla a continuación.

1. **Define métricas**: Establecer claramente las métricas que definen, para este caso las métricas son: 'NivelDeEntregaContinua', 'NivelDeIntegraciónContinua', 'NivelDeRefactoring' y 'NivelDeTesting'.
2. **Diseña instrumentos de medición:** Se crearon cuatro instrumentos o herramientas para obtener valoraciones cualitativas (alto, medio, bajo) para cada métrica.
3. **Aplica instrumento:** Se aplican los instrumentos a equipos de desarrollo front end para recabar las valoraciones.
4. **Transforma datos:** Se traducen las valoraciones cualitativas a valores difusos utilizando funciones de membresía predefinidas.
5. **Configura el sistema difuso:** Se utiliza Python para configurar el sistema de inferencia de Mandani con las 81 reglas ya establecidas.
6. **Procesa datos:** Ingresar los valores difusos en el sistema de Mandani y realizar la inferencia difusa.
7. **Interpreta salida:** Decodifica la salida difusa del sistema para obtener un nivel de agilidad claro (bajo, medio, alto).
8. **Valida y Ajusta:** Contrasta los resultados con evaluaciones expertas de agilidad para validar la precisión del sistema difuso y realiza ajustes si es necesario.

Esta metodología provee una guía estructurada para la evaluación de la agilidad en proyectos de desarrollo front end, facilitando un enfoque sistemático y replicable.

5. Implementación del sistema difuso

La Tabla 2, muestra las actividades seleccionadas con su relevancia, mismas que fueron consideradas inicialmente para la construcción del sistema difuso. Sin embargo, con el fin de optimizar el sistema en mención, se propuso fusionar las actividades Refactoring y Simple Design. Beck y colaboradores [14], sostienen que la refactorización continua es un componente crítico en el sostenimiento de un diseño simple, proporcionando la flexibilidad necesaria para acomodar cambios en los

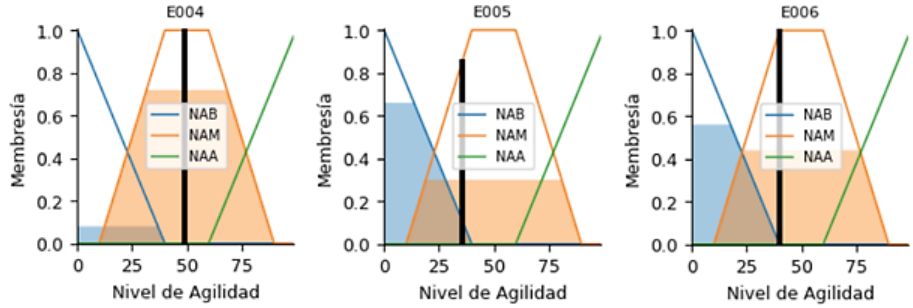


Fig. 4 Niveles de agilidad de las empresas E004, E005 y E006.

requisitos sin incrementar la complejidad del sistema. Para estas actividades, se definen las variables lingüísticas.

5.1. Definición de variables lingüísticas

Las variables lingüísticas son variables cuyos valores no son números sino términos o etiquetas lingüísticas en un lenguaje natural o artificial.

En la lógica difusa, permiten la manipulación de conceptos aproximados como "alto", "medio", o "bajo" y son utilizadas para describir tanto los antecedentes como los consecuentes de las reglas If-Then [20]. En este caso, las variables lingüísticas están asociadas con las actividades de agilidad en el desarrollo de aplicaciones. A cada actividad se le asocia una variable lingüística que describe el nivel de desempeño de la actividad correspondiente. El sistema está entonces definido por medio de 4 entradas y una salida conforme se muestra la Fig 4.

El esquema operativo presentado en la Fig. 4 incorpora un conjunto de reglas If-Then, fundamentales para la lógica difusa, que establecen una relación causal entre condiciones específicas (antecedentes) y resultados esperados (consecuentes). Mediante la aplicación de la inferencia difusa, el sistema traduce entradas con grados de verdad parciales en una salida difusa coherente, siguiendo un proceso de razonamiento aproximado que simula la capacidad humana de tomar decisiones en situaciones ambiguas o inciertas [19].

5.2. Definición de reglas If-Then

Para construir las reglas If-Then en un sistema de lógica difusa que maneja variables con tres grados de membresía, se emplea la Ecuación (1). Basándose en esta formulación, y utilizando el operador lógico 'y' (AND), se determina que un sistema que comprende 4 variables con tres grados de membresía cada una, se necesitan un total de 81 reglas If-Then. Este cálculo se detalla en la Ecuación (2):

$$\text{Núm. de reglas} = \text{Núm. de grados de membresía}^{\text{Núm. de variables}}, \tag{1}$$

$$\text{Núm de reglas} = 3^4 = 81. \tag{2}$$

Tabla 6 Años de Experiencia y Valor de Agilidad.

Empresa	Años de Experiencia	Valor De Agilidad	Nivel De Agilidad
E001	7	55.07	NAM
E002	15	57.08	NAM
E003	20	53.75	NAM
E004	7	49.07	NAM
E005	20	35.56	NAB
E006	4	39.83	NAB

Las variables lingüísticas con sus pesos y términos lingüísticos se definen como se muestra en la Tabla 3. De igual manera, la variable de salida denominada NivelDeAgilidad, tendrá los términos lingüísticos como se muestran en la Tabla 4.

5.3. Funciones de membresía

Las funciones de membresía son el núcleo de la lógica difusa, asignando valores cuantitativos a términos lingüísticos mediante grados de pertenencia, en contextos donde se prioriza la simplicidad y la claridad interpretativa, las funciones triangulares y trapezoidales son preferibles debido a su estructura lineal por partes y facilidad de cálculo [21]. En esta investigación, se optó por utilizar funciones de membresía triangulares para las variables de entrada, así como triangular y trapezoidal para la salida, coherentes con la necesidad de una modelación y computación eficiente y comprensible [19]. La Fig 5, exhibe una generalización de las funciones de membresía triangulares correspondientes a las variables de entrada las cuales fueron etiquetadas como se describe a continuación:

- 'NivelDeEntregaFrecuente'. Dichas funciones están categorizadas de la manera siguiente: 'BNE' representa un Bajo Nivel de Entrega, 'MNE' indica un Mediano Nivel de Entrega y 'ANE' denota un Alto Nivel de Entrega,
- 'NivelDeRefactoring'. Dichas funciones están categorizadas de la manera siguiente: 'BNR' representa un Bajo Nivel de Refactoring, 'MNR' indica un Mediano Nivel de Refactoring y 'ANR' denota un Alto Nivel de Refactoring.
- 'NivelDeIntegraciónContinua'. Dichas funciones están categorizadas de la manera siguiente: 'BNIC' representa un Bajo Nivel de Integración Continua, 'MNIC' indica un Mediano Nivel de Integración Continua y 'ANIC' denota un Alto Nivel de Integración Continua.
- 'NivelDeTesting'. Dichas funciones están categorizadas de la manera siguiente: 'BNT' representa un Bajo Nivel de Testing, 'MNT' indica un Mediano Nivel de Testing y 'ANT' denota un Alto Nivel de Testing.

La Fig. 6, exhibe las funciones de membresía correspondientes a la variable 'NivelDeAgilidad'. Dichas funciones están categorizadas de la manera siguiente: 'BNA' representa un Bajo Nivel de Agilidad, 'MNA' indica un Mediano Nivel de Agilidad y 'ANA' denota un Alto Nivel de Agilidad. El uso de la función trapezoidal es para dar mayor pertenencia a NAM. En la construcción del sistema difuso, se integró el nivel de

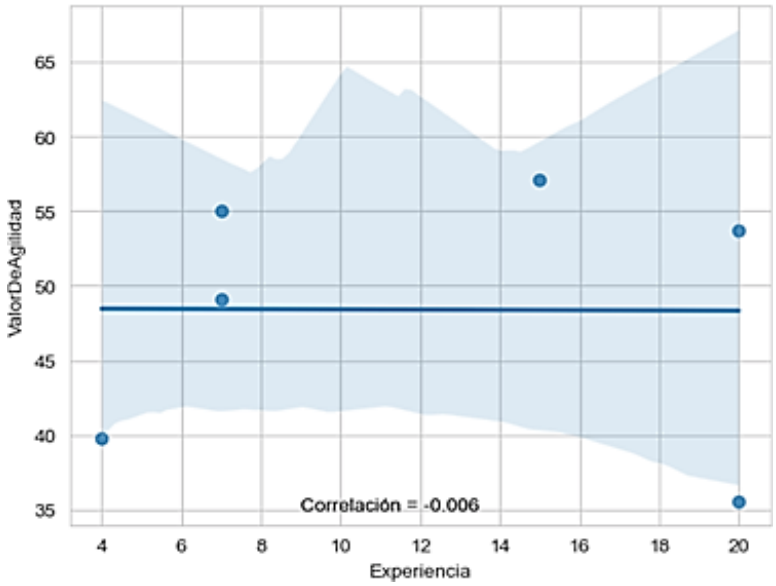


Fig. 5 Correlación de años de experiencia y nivel de agilidad.

agilidad dentro de las reglas If-Then, dichas reglas constituyen la estructura de las relaciones causales en sistemas de lógica difusa.

Estas reglas articulan los vínculos entre variables difusas, delineando cómo las condiciones (antecedentes) se mapean a las consecuencias (consecuentes) a través de la inferencia difusa. La utilización de estas reglas proporciona un mecanismo para deducir salidas difusas a partir de datos de entrada difusos, reflejando la complejidad y la naturaleza graduada de los procesos del mundo real [19].

5.4. Relevancia y ponderación de las actividades ágiles

Se calculó la relevancia de cada actividad ágil basándose en su porcentaje de importancia, tal como se detalla en la Tabla 4.

Este cálculo se efectuó mediante la aplicación de la fórmula **¡Error! No se encuentra el origen de la referencia.**, la cual fue diseñada para ponderar adecuadamente la contribución de cada actividad al nivel general de agilidad, las siglas se encuentran definidas en la Tabla 3. La formulación de las reglas *If-Then* refleja esta ponderación, permitiendo así que el sistema difuso evalúe con precisión el nivel de agilidad en el contexto del proyecto de software en estudio:

$$\text{Nivel De Agilidad} = NEF \times 0.38 + NR \times 0.39 + NIC \times 0.15 + NT \times 0.08. \quad (3)$$

Se desarrolló un sistema de inferencia difuso utilizando Python y el módulo Scikit-Fuzzy [22], una biblioteca especializada en algoritmos para lógica difusa. Se integraron y programaron las 81 reglas definidas, permitiendo una ejecución precisa del modelo de evaluación.

5.5. Instrumento de medición de las variables de entrada

Para obtener los valores de entrada del Sistema Difuso, se diseñó un cuestionario que fue contestado por los líderes de proyectos de desarrollo. Este instrumento incluye 10 preguntas para cada variable de entrada, las cuales son 'NivelDeEntregaFrecuente', 'NivelDeRefactoring', 'NivelDeIntegraciónContinua' y 'NivelDeTesting', cada pregunta con tres opciones de respuesta que reflejan la intensidad o frecuencia de las prácticas ágiles: '3' para un bajo nivel de implementación, '5' para un nivel medio, y '10' para un nivel alto. Las respuestas se suman, resultando en un puntaje total que conforma la entrada del sistema difuso.

6. Resultados y discusión

La aplicación de la metodología a casos de estudio reales revela su eficacia en capturar niveles de agilidad. La Tabla 5, muestra los resultados de la metodología aplicada a 6 empresas que se dedican al desarrollo de aplicaciones y utilizan metodologías ágiles.

De igual manera, las Fig. 7 y 8, representan de forma gráfica el resultado de la medición de la agilidad de las empresas E001, E002, E003, E004, E005 y E006 respectivamente.

En la representación gráfica de la medición de agilidad de las figuras Fig. 7 y Fig. 8, se destaca una barra vertical negra que indica el nivel de agilidad obtenido, evaluado en una escala de 0 a 100. Este valor se calcula utilizando el método del centroide para las áreas bajo las funciones de membresía.

Visualmente, a la izquierda de la gráfica, se presenta una función de membresía triangular para simbolizar baja agilidad; en el centro, una función trapezoidal denota agilidad media; y a la derecha, otra función triangular indica alta agilidad.

Además, el gráfico muestra una correspondencia cromática en la que el área coloreada aumenta conforme lo hace la relevancia del nivel de agilidad indicado. Esto permite una interpretación intuitiva de hasta qué punto el nivel calculado refleja una agilidad significativa en el contexto evaluado.

Además de las mediciones obtenidas, se incorporó en el instrumento de medición el registro de los años de experiencia de cada empresa evaluada. Este procedimiento se diseñó para investigar la posible correlación entre la experiencia acumulada de la empresa y su nivel de agilidad. Los datos se presentan detalladamente en la Tabla 6.

Aunque el conjunto de datos actual es limitado, se efectuó un análisis preliminar de correlación entre los años de experiencia de las empresas y sus niveles de agilidad medidos, resultando en un coeficiente de correlación de -0.006.

Este valor sugiere que no existe una relación significativa entre los años de experiencia y el nivel de agilidad alcanzado. La Figura 9, muestra el gráfico correspondiente. Se reconoce la necesidad de una base de datos más amplia para obtener resultados estadísticamente significativos.

7. Conclusiones

El modelo propuesto demuestra ser una herramienta prometedora para medir la agilidad en el desarrollo de software. Su capacidad para integrar la riqueza de la percepción humana con la rigurosidad de la evaluación cuantitativa aporta una comprensión más profunda de la agilidad práctica. Este enfoque ofrece un nuevo horizonte para investigaciones futuras y un marco para la mejora continua de las prácticas ágiles en la industria del software.

La adaptabilidad del sistema asegura que su aplicación puede extenderse a diversos entornos de desarrollo, haciendo de este modelo un candidato para la evaluación estándar de la agilidad en proyectos de software. Adicionalmente, este estudio no solo amplía el entendimiento actual sobre la medición de la agilidad, sino que también sienta las bases para el desarrollo de una metodología que permita evaluar el nivel de agilidad desde una perspectiva administrativa, excluyendo aspectos técnicos del desarrollo de software.

Este enfoque innovador posibilita la inclusión de organizaciones no dedicadas a este sector en la evaluación de agilidad de sus procesos administrativos. Al hacerlo, se amplía el alcance aplicable de las métricas de agilidad, facilitando su adopción en una variedad de contextos organizacionales.

Referencias

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifiesto por el desarrollo ágil de software http://agilemanifesto.org/iso/es/manifesto.html (2001)
2. Hohl, P., Klünder, J., van-Bennekum, A., Lockard, R., Gifford, J., Münch, J., Stupperich, M., Schneider, K.: Back to the Future: Origins and Directions of the "Agile Manifesto" – Views of the Originators. Journal of Software Engineering Research and Development, vol. 6, no. 1 (2018). DOI: 10.1186/s40411-018-0059-z.
3. Escobar-Sarmiento, V., Linares-Vasquez, M.: A Model for Measuring Agility in Small and Medium Software Development Enterprises. In: Proceedings of the XXXVIII Conferencia Latinoamericana en Informatica, pp. 1–10 (2012). DOI: 10.1109/clei.2012.6427226.
4. Royce, W.W.: Managing the Development of Large Software Systems: Concepts and Techniques. In: Proceedings of the 9th International Conference on Software Engineering, pp. 328–338 (1987)
5. Fox, A., Patterson, D.A.: Engineering Software as a Service: An Agile Approach using Cloud Computing. Strawberry Canyon LLC (2021)
6. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley Professional (1998)
7. Schwaber, K., Sutherland, J.: Scrum Guide V7 (2020)
8. Jeffries, R., Anderson, A., Hendrickson, C.: Extreme Programming Installed. Addison-Wesley Professional (2000)
9. Algarni, A., Magel, K.: Applying Software Design Metrics to Developer Story: A Supervised Machine Learning Analysis. In: IEEE First International Conference on Cognitive Machine Intelligence, pp. 156–159 (2019). DOI: 10.1109/cogmi48466. 2019.00030.
10. Rai, A.K., Agarwal, S., Kumar, A.: A Novel Approach for Agile Software Development Methodology Selection Using Fuzzy Inference System. In: International Conference on

Smart Systems and Inventive Technology, pp. 518–526 (2018). DOI: 10.1109/icssit.2018.8748767.
11. Meyer, B.: Agile: The Good, the Hype and the Ugly. Springer International Publishing (2014). DOI: 10.1007/978-3-319-05155-0.
12. Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit: An Agile Toolkit. Addison Wesley (2003)
13. Cockburn, A.: Crystal Clear a Human-Powered Methodology for Small Teams. Addison-Wesley Professional (2004)
14. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. Addison-Wesley (2004)
15. Ponce-Cruz, P.: Inteligencia Artificial con Aplicacion a la Ingeniería. Alfaomega (2010)
16. Voskoglou, M.: Fuzzy Sets, Fuzzy Logic and their Applications. MDPI Books (2020). DOI: 10.3390/books978-3-03928-521-1.
17. Williams, J.K.: Introduction to Fuzzy Logic. Massachusetts Institute of Technology, pp. 127–151 (2013). DOI: 10.1007/978-1-4020-9119-3_6.
18. Zadeh, L.: Fuzzy sets. Information and Control, vol. 8, no. 3, pp. 338–353 (1965). DOI: 10.1016/s0019-9958(65)90241-x.
19. Ross, T.J.: Fuzzy Logic with Engineering Applications. Wiley (2010). DOI: 10.1002/9781119994374.
20. Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. In: IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 1, pp. 28–44 (1973). DOI: 10.1109/tsmc.1973.5408575.
21. Klir, G.J., Yuan, B.: Fuzzy Sets, and Fuzzy Logic: Theory and Applications. Pearson College Div (1995)
22. The Python Software Foundation: Scikit-Fuzzy. http://pypi.org/project/scikit-fuzzy/ (2024)

ISSN 1870-4069

Técnicas de inteligencia artificial para la detección e identificación del daño en el área foliar y radicular de un cultivo de fabáceas bajo la técnica de aeroponía

Jessica A. Araujo Rodríguez, Norma V. Ramírez Pérez, José A. Padilla Medina, Alejandro I. Barranco Gutiérrez, Micael G. Bravo Sánchez

Instituto Tecnológico de Celaya, Guanajuato, México

D2203008@itcelaya.edu.mx

Resumen. En el trabajo presentado se establece el monitoreo de variables y toma de imágenes en IR y RGB del cultivo de Phaseolus Vulgaris L. Var. Opus bajo la técnica de aeroponía, para la futura determinación de daño provocado por estrés biótico y abiótico encontrado en el área foliar y radicular mediante el uso del software MATLAB, con la finalidad de establecer una estrategia de seguimiento evitando la pérdida del cultivo y aumentado la calidad y cantidad de producción procedente de la gestión adecuada de los nutrientes suministrados y el nulo o mínimo uso de pesticidas derivado diversas enfermedades propagadas por bacterias, hongos o virus que afecten a las diversas partes presentes en la planta perteneciente al cultivo. Las imágenes recuperadas y analizadas forman parte del trabajo de tesis de doctorado y en trabajos futuros se utilizarán para realizar una clasificación y predicción de daños provocados por enfermedades presentes en el cultivo y el comportamiento de diversos cultivos aplicados bajo la técnica de aeroponía respectivamente, utilizando redes neuronales convolucionales.

Palabras clave: Aeroponía, IoT, procesamiento de imágenes, agricultura de precisión.

Artificial Intelligence Techniques for Detection an Identification of Damage in the Leaf and Root Area Fabaceae Crop on Aeroponics Technique

Abstract. This work presented the monitoring of variables and IR and RGB images of the crop of Phaseolus Vulgaris L. Var. Opus on aeroponics technique for future determination of damage caused by biotic and abiotic stress found in the leaf and root area by using MATLAB software, in order to establish a monitoring strategy to avoid crop loss and increase the quality and quantity of production from the proper management of nutrients supplied and minimal use of pesticides derived from diseases spread by bacteria, fungi or viruses that affect

the various parts present in the plant belonging to the crop. The recovered and analyzed images are part of the doctoral thesis work and in future work will be used to perform a classification and prediction of damage caused by diseases present in the crop and the behavior of various crops applied on the aeroponics technique respectively, using convolutional neural networks.

Keywords: Aeroponics, IoT, image processing, precision agriculture.

1. Introducción

En los últimos años se ha visto un decremento en el uso de suelo como consecuencia de la erosión por las malas prácticas en la limpieza del suelo y el incremento de la población en las zonas urbanas, de igual forma la detección tardía de plagas y daños en el cultivo establecen una pérdida a la hora de establecer la producción agrícola. La aplicación de diversas tecnologías que permiten la detección oportuna de los daños encontrados en el área foliar y la predicción del comportamiento de diversos cultivos bajo diversas condiciones climáticas y diversas aportaciones de macro y micro nutrientes generan un amplio beneficio para determinar la aplicación de cultivo apropiado para sembrar y maximizar el rendimiento de acuerdo con los parámetros definidos por el programa computacional.

De igual forma la aplicación de tecnologías interconectadas que cumple funciones tales como la adquisición de imágenes del cultivo, la activación de extractores dentro de una invernadero, la toma de temperatura mediante sensores y la comunicación a un dispositivo embebido que envíe los datos a un almacenamiento en la nube, son útiles para la consecución efectiva de proceso de germinación, crecimiento y reproducción de un cultivo, minimizando las pérdidas dado a la corrección oportuna de los diversos problemas presentes en el cultivo. Se estima que para el año 2050 la producción de alimentos se verá superada por la demanda, dado al incremento exponencial de la población, por esta problemática se requiere una rápida producción de los alimentos básicos que puedan atender esta demanda disminuyendo la problemática de hambruna prevista por la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) para dentro de 25 años.

Por lo anterior, se da pie al término denominado agricultura de precisión, estrategia que establece el uso de tecnologías dentro de la agricultura que permiten mejorar la calidad y cantidad de producción previniendo corrigiendo las praxis que comprometen al cultivo, aplicando la robótica, el Internet de las cosas (IoT por sus siglas en inglés), la inteligencia artificial (IA por sus siglas), entre otras; generando y procesando información continua respecto a diversas variables involucradas en la producción de cultivos [1], como la humedad, la temperatura ambiente y temperatura dentro de la cámara de crecimiento, uso de agua y pesticidas, crecimiento vegetativo, cambio de temperatura foliar para la detección de estrés biótico y abiótico.

Generando de tal forma, un ahorro significativo de recurso hídrico y disminuyendo la contaminación por el uso mínimo o nulo de pesticidas a lo largo del ciclo fenológico del cultivo. De la misma manera la detección y diagnóstico oportuno derivado de estrés biótico o abiótico en las plantas evita en mayor medida la disminución de la producción agrícola derivado de la perdida de cultivos [2], gracias a las tecnologías involucradas

en el censado, adquisición, análisis y procesamiento de imágenes que permiten el funcionamiento adecuado del sistema de valoración del estado del cultivo.

Es por ello que en este artículo se expondrá una experimentación del cultivo de Phaseolus Vulgaris L. Var. Opus mediante una técnica sin suelo denominada aeroponía, así como el procesamiento de las imágenes capturadas del área foliar y radicular del cultivo, además, se explorarán los trabajos futuros que involucran en mayor medida a las variables obtenidas mediante tecnologías IoT y se propondrá la aplicación de redes artificiales convolucionales para identificar y clasificar la causa del daño presente en el área radicular y foliar de las plantas.

2. Materiales y métodos

2.1. Entorno de desarrollo

MATLAB es un entorno de desarrollo de programación matemático que cuenta con herramientas denominadas TOOLBOX que permiten trabajar con diferentes áreas como: visión artificial, aprendizaje automático, entre otras.

2.2. Dispositivos IoT

El Internet de las Cosas (IoT por sus siglas en inglés) es una de las tantas tecnologías emergentes que establecen el uso de diversos dispositivos inteligentes y conectados a internet que permite censar y registrar la información del entorno o proceso, dentro de una base de datos, para después accionar diversos elementos dependiendo de las decisiones registradas por el intermediario, el cual puede ser determinado por un sistema experto o un experto humano para las modificaciones o acciones necesarias dentro del entorno de estudio.

Para el trabajo presentado se determina el uso de indicadores de temperatura y humedad relativa que se encuentran conectados a wifi y que transmiten la información registrada a una Base de Datos MySQL. De la misma forma, se registra información del estado del recurso hídrico utilizado para fertirrigar el cultivo, como la concentración de pH, partículas por millón de los nutrientes diluidos y conductividad eléctrica (μs/cm). Este procesamiento y evaluación de información que permite respaldar la gestión de los recursos permitiendo la estabilidad del ciclo de cultivo establece lo que se denomina agricultura de precisión [3-4].

2.3. Visión artificial

La visión artificial es una rama de la IA que en conjunto con otros elementos es capaz de extraer información relevante de las imágenes presentadas. Es un campo amplio de aplicación en la agricultura de precisión, ya que ha permitido la segmentación de enfermedades en las hojas, detección de malezas, reconstrucción tridimensional de frutos y detección de los mismos [5], las imágenes obtenidas en el proceso de seguimiento de cultivo se podrán evaluar en una red neuronal convolucional (CNN por sus siglas en inglés) para determinar el tipo de afección que compromete las diversas etapas que conforman al ciclo fenológico del cultivo en cuestión.

Las CNN son ideales para el trabajo de procesamiento y análisis de imágenes dado al reconocimiento de patrones a partir de píxeles [6-8], con ello se podrá generar la determinación del tipo de estrés presentado en el área foliar y radicular del cultivo.

2.4. Aprendizaje automático

El aprendizaje automático es una técnica computacional que forma parte de la Inteligencia Artificial (IA), se basa en algoritmos que generan un autoaprendizaje según la experiencia, realizando un entrenamiento mediante patrones de entrada y patrones esperados de salida, consecuentemente el sistema será capaz de determinar sus propias respuestas dado los patrones de entrada, sin necesidad de supervisión humana; las entradas dadas, se proporcionan a través de un conjunto de datos que son separados en dos subconjuntos, entrenamiento y prueba; la salida del sistema se determina por medio de una predicción o clasificación generado por el subconjunto de entrenamiento [9]. Está técnica es utilizada en diversos campos como en la gestión de energía y procesos de control de edificios [10], detección del daño estructural [11], identificación de carcinoma mamaría [12], predicción de la supervivencia de los receptores de trasplantes [13], así como diversas aplicaciones en la agricultura [14-17], entre muchas otras aportaciones importantes al desarrollo social, industrial y económico.

2.5. Aeroponía

La aeroponía es una técnica de cultivo en la cual el suministro del recurso hídrico se realiza dispersando a manera de nebulización agua con nutrientes diluidos directamente sobre el área radicular del cultivo, generando un ambiente óptimo sin necesidad de involucrar sustrato o suelo para el desarrollo de las plantas [18-19].

A diferencia de las técnicas más conocidas como el sustrato, la hidroponía y suelo, esta técnica conserva a la planta suspendida en el aire manteniendo el área radicular de las plantas en una canastilla; el microambiente generado dentro de la cámara de crecimiento permite mantener la humedad necesaria para la supervivencia del cultivo, el suministro de recurso hídrico se realiza por aspersiones durante algunos segundos que son esenciales para proveer los nutrientes suficientes para alimentar e hidratar a cada una de las plantas, así mismo, existe un tiempo de espera en donde no se suministra agua al área radicular, sin embargo, esto no implica en mayor medida la pérdida de cultivo por estrés hídrico.

Los tiempos de espera durante el suministro de agua generan un ahorro considerable del recurso, aproximadamente un 97% a comparación con la técnica tradicional y un 90% comparado con la hidroponía [20], lo que concluye ser una técnica más sostenible dentro de la agricultura.

2.6. Estrés biótico y abiótico

Las plantas son organismos que de manera frecuente se encuentran expuestas a diversos elementos que producen estrés biótico y abiótico ya que forman parte de su entorno; cuando se habla del estrés abiótico este incluye la salinidad del agua (impide la correcta absorción de los nutrientes), temperaturas extremas (por encima o debajo de lo apropiado para el desarrollo del cultivo), escasez de nutrientes, la incidencia de

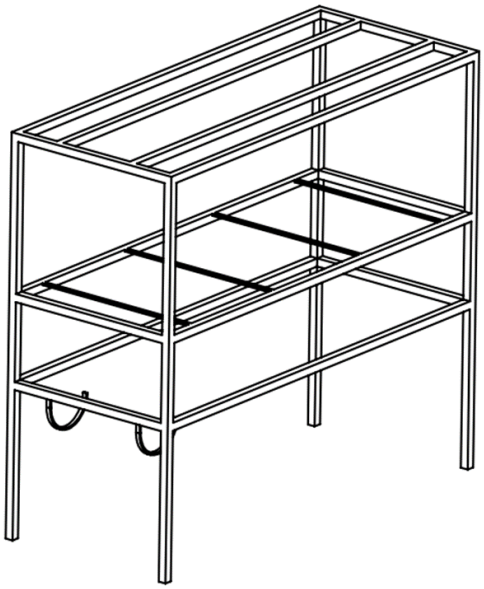


Fig. 1. Diseño de la estructura aeropónica utilizada en el cultivo de experimentación.

radiación solar sobre la planta, estrés hídrico y la presencia de metales pesados cuando se tiene un cultivo bajo la técnica en suelo; en cuanto al estrés biótico, hace referencia a todos los organismos como las plagas de insectos, bacterias, protistas, virus, nemátodos y hongos [21-22].

3. Diseño experimental

3.1. Funcionamiento de la cámara aeropónica

En el presente estudio se cultivó Phaseolus Vulgaris L. Var. Opus sobre una estructura aeropónica en un invernadero de cristal sin luz artificial con extractores rudimentarios de uso convencional. La estructura aeropónica está compuesta principalmente por acero al carbón ptr calibre 14, incluyendo refuerzos perimetrales, formando un prisma rectangular hueco, con una inclinación de 1 cm para generar una corriente de agua hacia uno de los lados de la estructura (ver Fig. 1), haciendo posible el retorno del suministro de agua a un recipiente secundario que a su vez retorna el agua a un recipiente principal que contiene la mayor parte de recurso hídrico con solución nutritiva.

En la parte media de la estructura se tienen tres conductos de material PVC seccionados transversalmente que se encuentran debajo de cada fila de plantas, su principal función es recabar el agua que el área radicular no requiere para su hidratación, regresando el suministro restante al recipiente principal. Para la nebulización del recurso hídrico se tiene un conjunto de válvulas adheridas a tubos de PCV que corren por debajo del área radicular de las plantas, esta aspersión se genera

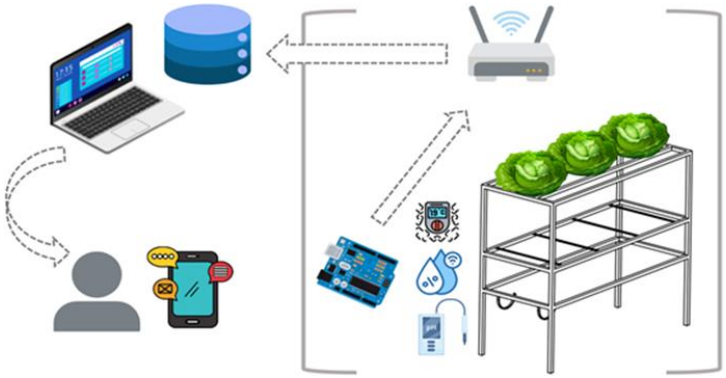


Fig. 2. Implementación de IoT en el seguimiento del ciclo fenológico del cultivo.

gracias al uso de una bomba sumergible que genera la precisión suficiente para disipar el agua por los tubos de PVC hasta las boquillas, generando una fertirrigación en forma de nebulización. La fertirrigación se realiza a manera de nebulización para evitar lastimar el área radicular de la planta y a su vez promoviendo una absorción homogénea.

Dentro de la cámara de crecimiento se genera un microambiente similar a las condiciones presentadas cuando el área radicular se encuentra en sustrato o suelo, estableciendo un entorno obscuro, húmedo y caliente que incentiva su desarrollo apropiado, además, las paredes que conforman el sistema aeropónico están recubiertas por plástico negro calibre 600 que evita la filtración de la radiación solar que pueden ser causante de quemaduras en las raíces. Mediante el uso de temporizadores se establece un tiempo de espera de aspersión y un tiempo (segundos) de fertirrigación, de esta manera el proceso de hidratación se realiza de manera automática por el sistema.

3.2. Cultivo experimental

El cultivo propuesto se realizó con Phaseolus Vulgaris L. Var. Opus es una especie perteneciente a la familia de las Fabáceas, es mejor conocida en Latinoamérica como frijol ejotero o bien denominado coloquialmente como ejote, siendo uno de los alimentos que aporta mayor concentración de fibra. La elección del cultivo experimental se estableció por la adaptabilidad del cultivo a altas temperaturas presentadas dentro del invernadero propuesto. El ejote consta de dos fases, vegetativa y reproductiva, la primera se encuentra establecida por los siguientes elementos: germinación (denominada V0), emergencia (V1), brote de hojas primarias (V2), primera hoja trifoliada (V3) y tercera hoja trifoliada (V4); la segunda fase consta de: prefloración (R5), floración (R6), formación de vainas (R7), llenado de vaina (R8) y maduración (R9).

3.3. Trasplante a la cámara de crecimiento

La germinación del cultivo se realizó en una esponja especial que no contiene ningún tipo de sustrato que se pueda diluir y mezclarse con el agua, esta esponja se denomina

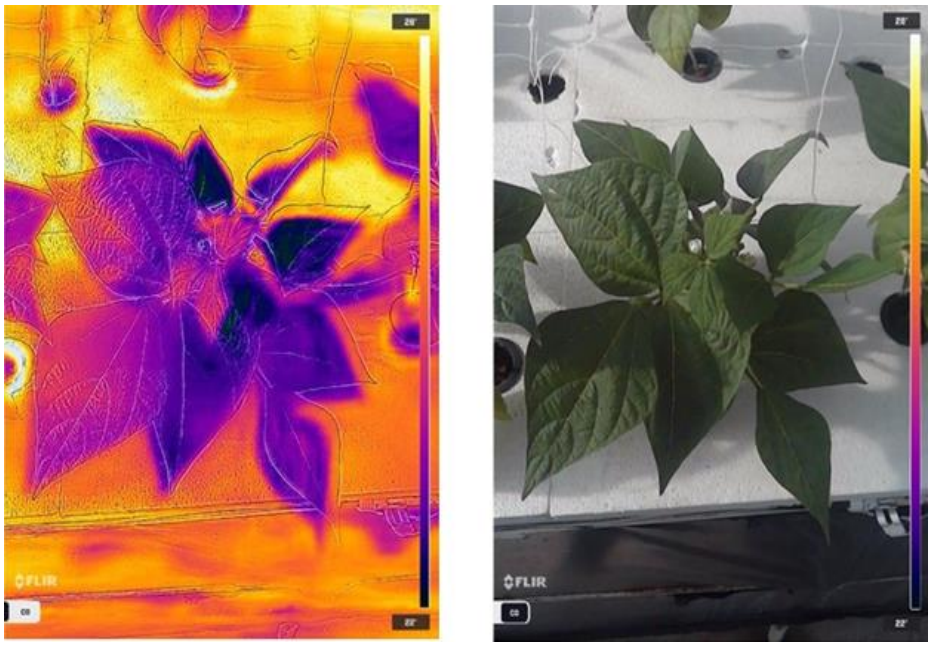


Fig. 3. Imágenes en IR y RGB de una planta del cultivo de Phaseolus Vulgaris L. Var Opus en un ambiente no controlado.

"esponja agrícola" o "fomi agrícola" el cual promueve un ambiente necesario para la germinación de la plántula; de manera inicial funciona como soporte de la semilla, ya que esta queda totalmente recubierta.

La esponja retiene la humedad durante un periodo amplio de tiempo lo que evita suministrar agua de manera frecuente, de igual manera, consta de un cuerpo poroso que permite que las raíces pasen a través de estos poros dejando de forma libre el crecimiento del área radicular. Cuando la plántula obtiene un tamaño aproximado de 15 centímetros, se procede con el trasplante de cada plántula a las canastillas hidropónicas para colocarlas en la estructura aeropónica.

3.4. Monitoreo de variables

Para la interpretación del análisis de las imágenes se requiere involucrar los diversos elementos que se encuentran implicados directa o indirectamente en el ciclo fenológico del cultivo, en los cuales se considera la medición continua de la temperatura ambiente, la humedad relativa de dos áreas (dentro del invernadero y dentro de la cámara de crecimiento aeropónica), intensidad lumínica, pH, partículas por millón contenidas en el recurso hídrico y conductividad eléctrica (µs/cm), la consideración de estas variables se debe principalmente a la correlación existente entre el comportamiento variables registradas y a la temperatura promedio calculada en la fotografía en IR del área foliar y radicular de las plantas.

A su vez, se realiza el registro de las diversas variables mediante sensores de temperatura y humedad relativa de dos zonas específicas ubicadas dentro del

Fig. 4. Filtrado de imagen mediante el TOOLBOX color thresholder para determinar el área foliar de la planta.

invernadero y de la cámara aeropónica, estos se encuentran en una placa Arduino con conexión wifi, los datos son transmitidos a una base de datos creada en el Sistema gestor MySQL.

De igual forma, se toman las medidas de las variables que involucran al suministro de solución nutritiva diluida en el agua y pH, las cuales de igual forma se envían a la base de datos. Además, de manera paralela se realiza el registro del consumo de recurso hídrico por aspersión, clasificándolo en útil y retornado, así como el total utilizado por cada fase dentro del ciclo fenológico (fase vegetativa y fase reproductiva). Durante el proceso de adquisición de valores de las variables descritas anteriormente obtenidas por los diversos sensores conectados a una placa Arduino, se realiza una comparación continua de los valores obtenidos con los rangos óptimos, en caso de salir del rango, se envía una alerta al dispositivo móvil (ver Fig. 2).

3.5. Captura de imágenes

Para el seguimiento del comportamiento en el ciclo fenológico del cultivo de Phaseolus Vulgaris L. Var. Opus se realizaron tomas de fotografías aéreas en IR y RGB mediante una cámara FLIR ONE Pro 3ra generación que se encuentra sujetada en una estructura metálica en la parte superior del área foliar de la planta para la toma de imágenes, con un periodo de 3 fotografías por semana para la observación continua del proceso fenológico y la aparición de estrés biótico o abiótico.

En cuanto a la toma de las imágenes del área radicular se realiza mediante una estructura montada en la pared lateral de la estructura lo que permite desplazar la cámara adecuadamente por capturar cada una de las plantas. Las imágenes adquiridas se realizan en un ambiente no controlado (ver Fig. 3), por tal motivo, se requiere disminuir el ruido presente en cada una de ellas, esto es posible gracias a las herramientas que contiene el software MATLAB.

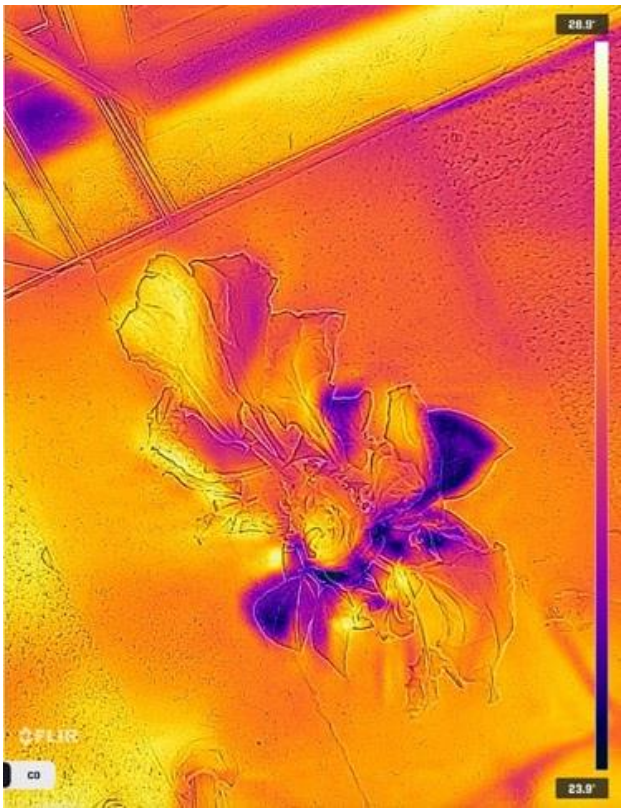


Fig. 5. Reducción de área foliar como consecuencia estrés hídrico.

3.6. Análisis y procesamiento de imágenes

FLIR ONE Pro 3ra generación es una cámara térmica de clasificación profesional para Smartphones, las cual permite detectar diferencias de temperatura en las imágenes de 70mK (miliKelvins) y las cuales pueden llegar hasta 400 °C, las imágenes constan de una dimensión de 1080×1440 píxeles y 72 ppp (píxeles por pulgada).

Ambas imágenes la térmica y RGB son tranferidas al software MATLAB para su posterior análisis, en ambos casos considerando el punto de referencia para determinar la cantidad de píxeles que forman un centímetro cuadrado.

De tal manera, las imágenes RGB contienen una escala de color de 0 a 255, de esta forma, es posible determinar, para la imagen RGB el área foliar, ignorando aquello que no conforma a la planta; en cuanto a la imagen térmica, se obtiene el promedio de temperatura encontrado en el área foliar visible desde la vista aérea realizando un conteo de píxeles. Las imágenes en RGB e IR se cargan al software de procesamiento en el entorno MATLAB, el procesamiento se realiza mediante un código con extensión .m que contiene un anexo de script generado en el TOOLBOX Color Thresholder, el programa general contiene la carga automática del conjunto de imágenes tomadas en un día determinado.

Fig. 6. Tablas de la base de datos utilizada para el almacenamiento de la información monitoreada.

Este programa realiza la segmentación de las imágenes mediante los umbrales de color determinados, de esta forma es posible segmentarla al trabajar con el filtrado mediante la matiz, saturación y brillo (HSV) considerando únicamente aquellos colores verdes (al tratar área foliar) o blanquizco-amarillo (cuando se trata el área radicular) encontrados en la imagen, generando una mascara de segmentación binaria para una imagen de color con la que se podrán trabajar el resto de imágenes para determinar el área, la temperatura promedio encontrada en el área foliar y radicular, así como la posibilidad de identificar el tipo de estrés presentado.

Al realizar el filtro mencionado, se omite todo aquello que no forma parte de la planta (ver Fig. 4), estableciendo una matriz que contiene valores de 0 y 1, la cual a su vez se contrapone sobre la imagen térmica para determinar únicamente los valores encontrados en el mapa de calor correspondiente al área seleccionada de la imagen, logrando el cálculo de la temperatura registrada de cada uno de los pixeles, de tal manera, que es posible determinar el promedio de temperatura presente en el área de estudio. Así pues, los puntos más altos de calor capturados en la imagen se pueden determinar en el último paso, permitiendo evaluar las áreas afectadas en busca de posible plaga presente en el cultivo; o bien, si las zonas concentradas de calor se enfocan en la totalidad del área foliar o radicular, se puede establecer un análisis de identificación de estrés abiótico.

4. Resultados preliminares

El presente trabajo es una demostración del avance de la experimentación realizada en la investigación del segundo año del Doctorado en Ciencias de la Ingeniería en donde se aplican conocimientos multidisciplinarios que permiten orientar un cultivo bajo la técnica de aeroponía hacia la agricultura de precisión fomentando el uso adecuado de los recursos involucrados en el desarrollo apropiado de diversos cultivos para obtener un mejor rendimiento, considerando las diversas afecciones negativas que se pueden presentar durante el ciclo fenológico

Actualmente el sistema de procesamiento de imágenes programado es capaz de calcular el área observable de la vista aérea del cultivo, así como la determinación del porcentaje de daño presente en las hojas, de igual forma, se determina la temperatura promedio del área foliar; a su vez, se realiza el cálculo del área radicular y la

temperatura promedio, esto gracias al apoyo de la herramienta del software MATLAB con ayuda del TOOLBOX Color Thresholder.

De acuerdo a la literatura es posible determinar en qué punto una planta comienza a presentar estrés, haciendo más evidente la presencia de estrés hídrico dado a la disminución del área foliar (ver Fig. 5) [23-24]. Los resultados de los análisis realizados y las imágenes originales y procesadas, son guardados en la base de datos perteneciente al estudio, de igual manera, se generan documentos con extensión .mat que contienen todas las imágenes originales y procesadas, además de los cálculos obtenidos, así mismo, la comparación de resultados entre los cálculados y los aproximados manualmente se establece mediante el área de una figura irregular, esta información es almacenada en una Base de Datos en el Sistema Gestor MySQL (ver Fig. 6) al igual la toma de variables adquiridas por los dispositivos IoT.

5. Trabajos futuros

Se pretende realizar una mejora en el procesamiento de las imágenes obtenidas mediante la cámara FLIR ONE Pro 3ra generación, ya que el ambiente en donde se toma la fotografía no es controlable debido a la imposibilidad de desplazar a las plantas a un ambiente con la misma iluminación evitando la existencia de ruido.

Se recabaron todas las tomas aéreas de las imágenes registradas por la cámara FLIR ONE Pro 3ra generación para analizar los estados de estrés biótico y abiótico presentes en el ciclo fenológico del cultivo de Phaseolus Vulgaris L.Var Opus para que en un futuro mediante una CNN se gestionen los recursos involucrados en el desarrollo de las plantas y mejorar la toma de decisiones en cuanto a la prevención y corrección de enfermedades presentes en el área foliar y radicular debido a plagas derivadas de hongos, virus o bacterias, que pudieran afectar el rendimiento o generar la pérdida total o parcial del cultivo.

Se tiene un avance significativo proporcionado por la adquisición en tiempo real de las variables involucradas en el proceso de crecimiento y desarrollo del cultivo y a su vez por el almacenamiento de la base de conocimiento que alimentará al sistema que realizará la gestión de los recursos y a las fotografías que servirán de referencia para el entrenamiento y prueba de la CNN incluyendo bases de datos externas que permitirán generar un resultado con un rango de error mínimo para esclarecer los pasos a seguir en la consecución exitosa de diversos cultivos que se pretender generar, ampliando la aplicabilidad del sistema.

Finalmente se diseñará e implementará una aplicación web que permita observar el comportamiento en tiempo real de las variables medidas y del estado del cultivo considerando las imágenes tomadas, de igual forma visualizar los históricos de comportamiento de las plantas durante el ciclo fenológico y el consumo de nutrientes generado en cada fase.

6. Conclusiones

El monitoreo continuo del ciclo fenológico de todo cultivo en cualquier técnica, es de suma importancia para asegurar el aprovechamiento de los recursos y la mejora del

rendimiento, de tal forma que se obtenga un alto beneficio al utilizar las diversas tecnologías que se proporcionan en la agricultura de precisión, esta comunicación entre dispositivos permite la valoración continua del cultivo precisando la supervivencia y aumentado el rendimiento. Dado al panorama de crecimiento poblacional y la disminución de suelo cultivable para los próximos años, se insta la importancia de la aplicación de IoT y el Aprendizaje Automático, así como la consideración de técnicas de cultivo sostenibles que no involucren indispensablemente suelo para la producción de alimentos seguros y asequibles para la población, de igual forma la producción a gran escala conlleva al uso desmesurado de recursos y tiempo en el cuidado de los cultivos, por lo que se considera apropiado la aplicación de tecnologías capaces de gestionar los recursos adecuadamente y de mejorar la toma de decisiones para el control de infecciones, minimizando el impacto ambiental y aumentando la disponibilidad de alimentos frescos.

Referencias

1. Abioye, E.A., Hensel, O., Esau, T.J., Elijah, O., Abidin, M.S.Z., Ayobami, A.S., Yerima, O., Nasirahmadi, A.: Precision Irrigation Management using Machine Learning and Digital Farming Solutions. AgriEngineering, vol. 4, no. 1, pp. 70–103 (2022). DOI: 10.3390/agrieng ineering4010006.
2. Alam, M., Alam, M.S., Roman, M., Tufail, M., Khan, M.U., Khan, M.T.: Real-time Machine-learning based Crop/Weed Detection and Classification for Variable-rate Spraying in Precision Agriculture. In: Proceedings of the 7th International Conference on Electrical and Electronics Engineering, pp. 273–280 (2020). DOI: 10.1109/iceee49618. 2020.9102505.
3. Alanne, Kari, Sierla, S.: An Overview of Machine Learning Applications for smart Buildings. Sustainable Cities and Society, vol. 76, pp. 103445 (2022). DOI: 10.1016/ j.scs.2021.103445.
4. Aslan, M.F., Durdu, A., Sabanci, K., Ropelewska, E., Gültekin, S.S.: A Comprehensive Survey of the Recent Studies with UAV for Precision Agriculture in Open Fields and Greenhouses. Applied Sciences, vol. 12, no. 3, pp. 1047 (2022). DOI: 10.3390/ app12031047.
5. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D.J.: A Review of Vibration-based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and Deep Learning Applications. Mechanical Systems and Signal Processing, vol. 147, pp. 107077 (2021). DOI: 10.1016/j.ymssp.2020.107077.
6. Berger, K., Machwitz, M., Kycko, M., Kefauver, S.C., Van-Wittenberghe, S., Gerhards, M., Verrelst, J., Atzberger, C., van-der-Tol, C., Damm, A., Rascher, U., Herrmann, I, Paz, V. S., Fahrner, S., Pieruschka, R., Prikaziuk, E., Buchaillot, M.L., Halabuk, A., Celesti, M., Koren, G., et. al.: Multi-sensor Spectral Synergies for Crop Stress Detection and Monitoring in the Optical Domain: A Review. Remote Sensing of Environment, vol. 280, pp. 113198 (2022). DOI: 10.1016/j.rse.2022.113198.
7. Chugh, G., Kumar, S., Singh, N.: Survey on Machine Learning and Deep Learning Applications in Breast Cancer Diagnosis. Cognitive Computation, vol. 13, no. 6, pp. 1451–1470 (2021). DOI: 10.1007/s12559-020-09813-6.

8. Cravero, A., Sepúlveda, S.: Use and Adaptations of Machine Learning in Big Data-Applications in Real Cases in Agriculture. Electronics, vol. 10, no. 5, pp. 552 (2021). DOI: 10.3390/electronics10050552.
9. Ferrag, M.A., Shu, L., Friha, O., Yang, X.: Cyber Security Intrusion Detection for Agriculture 4.0: Machine Learning-based Solutions, Datasets, and Future Directions. IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 3, pp. 407–436 (2022). DOI: 10.1109/jas.2021.1004344.
10. Gotlieb, N., Azhie, A., Sharma, D., Spann, A., Suo, N., Tran, J., Orchanian-Cheff, A., Wang, B., Goldenberg, A., Chassé, M., Cardinal, H., Cohen, J.P., Lodi, A., Dieude, M., Bhat, M.: The promise of Machine Learning Applications in Solid Organ Transplantation. NPJ Digital Medicine, vol. 5, no. 1 (2022). DOI: 10.1038/s41746-022-00637-2.
11. Hidayatuloh, A., Nursalman, M., Nugraha, E.: Identification of Tomato Plant Diseases by Leaf Image using Squeezenet Model. In: Proceedings of the International Conference on Information Technology Systems and Innovation, pp. 199–204 (2018). DOI: 10.1109/icitsi.2018.8696087.
12. Kateb, F.A., Monowar, M.M., Hamid, M. Abdul, O., Abu Q., Mridha, M.F.: Fruitdet: Attentive Feature Aggregation for Real-Time Fruit Detection in Orchards. Agronomy, vol. 11, no. 12, pp. 2440 (2021). DOI: 10.3390/agronomy 11122440.
13. Khan, M.M., Akram, M.T., Janke, R., Qadri, R.W.K., Al-Sadi, A.M., Farooque, A.A.: Urban Horticulture for Food Secure Cities through and Beyond COVID-19. Sustainability, vol. 12, no. 22, pp. 9592 (2020). DOI: 10.3390/su12229592.
14. Lakhiar, I.A., Gao, J., Syed, T.N., Chandio, F.A., Tunio, M.H., Ahmad, F., Solangi, K.A.: Overview of the Aeroponic Agriculture – An Emerging Technology for Global Food Security. International Journal of Agricultural and Biological Engineering, vol. 13, no. 1, pp 1-10 (2020). DOI: 10.25165/j.ijabe.20201301.5156.
15. Lee, S.H., Goëau, H., Bonnet, P., Joly, A.: New Perspectives on Plant Disease Characterization based on Deep Learning. Computers and Electronics in Agriculture, vol. 170, pp. 105220 (2020). DOI: 10.1016/j.compag.2020.105220.
16. Li, Q., Li, X., Tang, B., Gu, M.: Growth Responses and Root Characteristics of Lettuce Grown in Aeroponics, Hydroponics, and Substrate Culture. Horticulturae, vol. 4, no. 4, pp. 35 (2018). DOI: 10.3390/horticulturae4040035.
17. Li, Y., Nie, J., Chao, X.: Do we Really Need Deep CNN for Plant Diseases Identification? Computers and Electronics in Agriculture, vol. 178, pp. 105803 (2020). DOI: 10.1016/j.compag.2020.105803.
18. Miragaia, R., Chávez, F., Díaz, J., Vivas, A., Prieto, M.H., Moñino, M.J.: Plum Ripeness Analysis in Real Environments using Deep Learning with Convolutional Neural Networks. Agronomy, vol. 11, no. 11, pp. 2353 (2021). DOI: 10.3390/agronomy11112353.
19. Moso, J.C., Cormier, S., de-Runz, C., Fouchal, H., Wandeto, J.M.: Anomaly Detection on Data Streams for Smart Agriculture. Agriculture, vol. 11, no. 11, pp. 1083 (2021). DOI: 10.3390/agriculture11111083.
20. Ouhami, M., Hafiane, A., Es-Saady, Y., El-Hajji, M., Canals, R.: Computer Vision, IOT and Data Fusion for Crop Disease Detection using Machine Learning: A Survey and Ongoing Research. Remote Sensing, vol. 13, no. 13, pp. 2486 (2021). DOI: 10.3390/rs13132486.
21. Sharma, N., Sharma, R., Jindal, N.: Machine Learning and Deep Learning Applications-A Vision. Global Transitions Proceedings, vol. 2, no. 1, pp. 24–28 (2021). DOI: 10.1016/j.gltp.2021.01.004.
22. Shurtleff, M.C., Pelczar, M.J., Kelman, A., Pelczar, R.M.: Plant disease. Britannica. http://www.britannica.com/science/plant-disease (2024)

23. Habib, N., Ali, Q., Ali, S., Javed, M.T., Zulqurnain-Haider, M., Perveen, R., Shahid, M.R., Rizwan, M., Abdel-Daim, M.M., Elkelish, A., Bin-Jumah, M.: Use of Nitric Oxide and Hydrogen Peroxide for Better Yield of Wheat (Triticum Aestivum l.) under Water Deficit Conditions: Growth, Osmoregulation, and Antioxidative Defense Mechanism. Plants, vol. 9, no. 2, pp. 285 (2020). DOI: 10.3390/plants9020285.
24. Sah, R.P., Chakraborty, M., Prasad, K., Pandit, M., Tudu, V.K., Chakravarty, M.K., Narayan, S.C., Rana, M., Moharana, D.: Impact of Water Deficit Stress in Maize: Phenology and Yield Components. Scientific Reports, vol. 10, no. 1 (2020). DOI: 10.1038/s41598-020-59689-7.

ISSN 1870-4069

PLN con Transformers para detección de toxicidad: construcción y evaluación de corpus para la plataforma MisProfesores.com

María Lucía Barrón Estrada, Ramón Zatarain Cabada,
Ramón Alberto Camacho Sapien, Víctor Manuel Bátiz Beltrán

Instituto Tecnológico de Culiacán, Posgrado e Investigación,
México

{lucia.be, ramon.zc, ramon.cs, victor.bb}@culiacan.tecnm.mx

Resumen. El crecimiento de las redes sociales como medios de comunicación ha permitido una interacción más rápida y directa entre los usuarios, pero por otra parte presenta desafíos, como el riesgo de difusión de discursos de odio. Detectar estas publicaciones dañinas tempranamente es crucial. Este artículo presenta una metodología para crear un corpus único de comentarios en español obtenidos de la plataforma MisProfesores.com, abarcando todas las entidades federativas de México. Este proceso resultó en un conjunto de datos de 18,000 muestras no etiquetadas y 853 muestras etiquetadas manualmente. Además de describir el proceso de construcción del corpus, se exponen los resultados de la evaluación de varios modelos entrenados con estos datos, así como su comparación con trabajos previos para la detección de toxicidad existentes en el estado del arte, evidenciando la importancia del desarrollo de corpus en español para tareas específicas.

Palabras clave: Análisis de sentimientos, aprendizaje profundo, BERT, corpus, toxicidad, transformers.

NLP with Transformers for Toxicity Detection: Corpus Construction and Evaluation for MisProfesores.com Platform

Abstract. The growth of social networks as a means of communication has enabled faster and more direct interaction between users, but also presents challenges, such as the risk of spreading hate speech. Detecting these harmful publications early is crucial. This paper presents a methodology to create a unique corpus of Spanish-language comments obtained from the MisProfesores.com platform, covering all Mexican states. This process resulted in a dataset of 18,000 unlabeled samples and 853 manually labeled samples. In addition to describing the corpus construction process, the results of the evaluation of several models trained with these data are presented, as well as their comparison with previous works for toxicity detection existing in the state of the art, evidencing the importance of corpus development in Spanish for specific tasks.

Keywords: Sentiment analysis, deep learning, BERT, corpus, toxicity, transformers.

1. Introducción

El análisis de sentimientos es una subárea del procesamiento de lenguaje natural (PNL) que se enfoca en la identificación automática y la categorización de emociones y sentimientos expresadas dentro de un texto [1]. Este proceso es aplicable a diferentes sectores de la sociedad. Por ejemplo, los medios digitales de comunicación como las redes sociales, en dónde el cambio y expresión de ideas son una actividad recurrente, requieren un monitoreo constante para garantizar la integridad de los usuarios. Esto ha convertido a dichos medios en un sector importante donde aplicar el análisis de sentimientos. Como resultado, el análisis de sentimientos aumentó su popularidad entre las comunidades de investigación en los años recientes [2].

En plataformas como MisProfesores (www.misprofesores.com) en dónde el idioma español es el predominante, usar modelos de aprendizaje para aplicar un análisis de sentimientos representa una tarea complicada debido a la complejidad y diversidad de la lengua. El desarrollo de modelos de aprendizaje automático efectivos para el análisis de sentimientos en textos en español está limitado por los corpus en español existentes. En este artículo, revisaremos el proceso de construcción de un corpus en español a partir de la extracción de los comentarios publicados por usuarios de nacionalidad mexicana en la plataforma MisProfesores.

Este artículo está organizado de la siguiente manera. En la sección 2 se hace un repaso de los trabajos relacionados que hacen referencia a la construcción de corpus en español. En la sección 3, describimos el proceso para la recolección y procesamiento de los datos. La sección 4 muestra la metodología empleada, incluyendo el proceso de construcción de datos, así como los algoritmos y modelos de aprendizaje automático usados para las pruebas. Los resultados de las pruebas están en la sección 5. Y por último en la sección 6 presentamos nuestras conclusiones.

2. Trabajos relacionados

La detección de toxicidad en comentarios de Internet se ha consolidado como un área de interés creciente dentro del campo del PLN. Diversas plataformas que permiten calificar y dejar reseñas sobre docentes se han convertido en focos significativos para el análisis de sentimientos, dado que acumulan una gran variedad de comentarios. Un ejemplo notable de investigación en este ámbito es el estudio realizado por Arceo-Gomez and Campos-Vazquez [3] en donde se llevó a cabo un análisis estadístico exhaustivo sobre aproximadamente 600,000 evaluaciones.

Este estudio no solo proporciona perspectivas sobre las interacciones en dichas plataformas, sino que también destaca la presencia de estereotipos de género, subrayando la importancia de las técnicas de PLN para identificar y mitigar sesgos implícitos. Por otra parte, en el trabajo presentado por Kolhatkar et al. [4] se describe el desarrollo de un corpus considerable a partir de comentarios en inglés, recopilados de sitios web de noticias, que incluye cerca de 500,000 muestras.

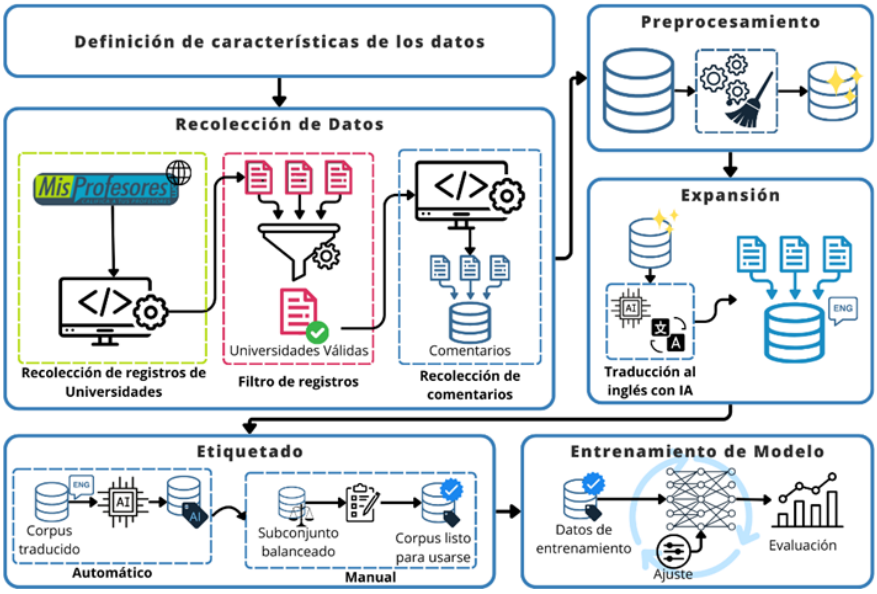


Fig. 1. Diagrama representativo para la metodología.

Este estudio es particularmente relevante porque no solo desarrollaron un corpus amplio, sino que también etiquetaron un subconjunto de aproximadamente 1000 muestras, enfocándose en la clasificación de la toxicidad de los comentarios.

Este enfoque ofrece una base sólida para futuras investigaciones sobre herramientas automatizadas de moderación de contenido. En relación con el uso y la eficacia de los modelos de inteligencia artificial (IA), Nabiilah et al. [5] aportan una valiosa comparativa entre modelos preentrenados en distintos corpus. Los hallazgos indican que aquellos modelos entrenados con corpus en idiomas específicos obtienen mejores resultados en tareas de clasificación de toxicidad en el mismo idioma.

Esto subraya la importancia de considerar las variaciones lingüísticas y culturales al diseñar y entrenar modelos de IA para la moderación de contenido. En contraste con los trabajos previamente mencionados, este estudio presenta un enfoque que amalgama los procedimientos y resultados revisados anteriormente. Aquí, se introduce una metodología integral que abarca áreas como la minería de opiniones, la construcción de conjuntos de datos y el entrenamiento de modelos.

3. Metodología

En la sección de metodología, se detalla un proceso comprensivo que abarca desde la definición inicial de las características deseables del corpus hasta el proceso de entrenamiento de distintos modelos usando un corpus etiquetado manualmente desarrollado en este trabajo. En la Fig.1 se presenta un diagrama que ilustra cada etapa de la metodología.

Fig. 2. Reseña extraída de la plataforma MisProfesores.

3.1. Definición de características de los datos

Como se observa en la Fig. 1, el primer paso para la construcción del corpus es definir las características de los datos a extraer.

Se estableció una escala geográfica a nivel nacional como la más adecuada, dado que está limitada por el alcance de la plataforma MisProfesores. Se seleccionó cada estado del país como puntos de interés para la extracción de los datos de dicha plataforma. Esto aseguró abarcar la diversidad lingüística de México. Posteriormente, se definieron las características de las instituciones académicas de las cuales se extrajeron los datos. Dichas instituciones cumplen con las siguientes características: ser universidad pública, escolarizada y con la mayor relevancia a nivel estatal según la cantidad de matrículas registradas. Lo anterior se realizó con base en los datos de la ANUEIS 2023 [6]. Los datos de interés para su obtención fueron las reseñas realizadas por los alumnos hacía los profesores de las instituciones previamente seleccionadas. Cada reseña en la plataforma está compuesta por un comentario/opinión del alumno, la materia, la calificación obtenida con el docente en la materia, la fecha, entre otros (Véase Fig. 2).

3.2. Recolección de datos

Mediante técnicas de extracción de textos de sitios Web, comúnmente conocidas como Web Scrapping se extrajeron los datos del sitio web MisProfesores. Para ello, se utilizaron herramientas como: Selenium y BeautifulSoup, ambas bibliotecas de Python usadas para extraer datos generados de manera dinámica y estática respectivamente. Debido a la flexibilidad que ofrece la plataforma MisProfesores para registrar profesores e instituciones educativas, se implementó un filtro en el algoritmo de extracción. Este filtro únicamente capturaba los registros de las universidades más destacadas en los resultados de la plataforma, como se muestra en la Fig. 3. La selección del filtro se basó en el número de registros de profesores asociados a cada institución educativa.

3.3. Preprocesamiento de los datos

Para el comentario de cada reseña, el texto obtenido fue sometido a un proceso en donde fueron eliminados signos de puntuación innecesarios. Por ejemplo, signos de exclamación o interrogación repetidos al iniciar o terminar una oración. Con este procedimiento, fue reducido el ruido presente en el corpus. También fueron descartadas las reseñas que presentaban un comentario:

Escuela	Ciudad	Estado	Num. de Profs.
Universidad Autónoma de Sinaloa	Culiacán	Sinaloa	117
Universidad autónomas de sinaloa	Culiacán	Sinaloa	1
Universidad Autónoma de Sinaloa	Angostura	Sinaloa	0
UNIVERSIDAD AUTONOMA DE SINALOA	LOS MOCHIS	SINALOA	6
Universidad autónoma de Sinaloa	Guamuchil	Sinaloa	1
Universidad Autónoma de Sinaloa	Culiacan	Sinaloa	16
Universidad Autónoma de Sinaloa	Mazatlán	Sinaloa	34

Fig. 3. Resultados de búsqueda para "Universidad Autónoma de Sinaloa" remarcado en rojo el registro más relevante en base al número de profesores registrados.

Fig. 4. Muestra de reseña con comentario inválido.

- Que solo estuviese compuesto por espacios en blanco o completamente vacío.
- Compuesto solo por caracteres especiales y/o números.
- En espera de revisión o bloqueados por la misma plataforma MisProfesores.

En la Fig. 4 se muestra un ejemplo de una reseña con un comentario inválido, en este el comentario se encuentra en espera de revisión por la plataforma. Posterior a este procedimiento de filtración de los datos, se obtuvo un corpus de 18,000 muestras dónde cada muestra contiene los datos mencionados anteriormente en el apartado 3.1.

3.4. Expansión del conjunto de datos

Para enriquecer el conjunto de datos se aplicó una traducción del español al inglés a cada comentario del corpus. Por la cantidad de muestras presentes en el corpus, la traducción se realizó de manera automática mediante modelos basados en Transformers. Como resultado, se obtuvo una versión en inglés del corpus, la cual amplió los casos de uso de este. Dicha versión en inglés nos sirvió para la etapa de etiquetado automático.

3.5. Etiquetado del conjunto de datos

Por la magnitud del corpus obtenido, se realizó un etiquetado automático a los comentarios en su versión en inglés utilizando varios modelos clasificadores basados

Tabla 1. Comparación entre distintas arquitecturas de modelos usando el Corpus MisProfesores.

Modelo	Exactitud	Recall	F1
EvoMSA BoW	0.8479	0.8212	0.8067
EvoMSA BoW + Text Representation	0.8596	0.8522	0.8262
LSTM	0.7134	0.6714	0.6761
mBERT base	0.8011	0.8011	0.8027
XLM RoBERTa base	0.8245	0.8245	0.8233
BETO base	0.9649	0.9649	0.9645

en Transformers entrenados para analizar sentimientos y clasificar toxicidad en texto. Este procedimiento permite obtener resultados preliminares y conocer el estado del balance de los datos. Con base en este primer etiquetado automático, se extrajo un subconjunto balanceado de muestras tóxicas y no tóxicas. Posteriormente, profesores de nuestra institución lo etiquetaron manualmente.

Para llevar a cabo el etiquetado manual, se establecieron diversos criterios a considerar. Por ejemplo, se instruyó al equipo a leer minuciosamente cada comentario con el fin de identificar posibles expresiones sarcásticas. Además, se enfatizó la importancia de no basar exclusivamente la etiqueta del comentario en palabras altisonantes, sino evaluar el contexto en el que se utilizan. Como resultado, se obtuvo un corpus etiquetado manualmente conformado por 853 muestras. Cada muestra cuenta con un texto en español presente en el comentario de la reseña original y una etiqueta binaria. En la etiqueta binaria, el número "1" representa que el texto de la muestra es tóxico y el número "0" significa la ausencia de toxicidad en el texto.

3.6. Entrenamiento de modelo para clasificación de texto

Para desarrollar un modelo capaz de clasificar comentarios en la plataforma MisProfesores como tóxicos o no tóxicos, se procedió al entrenamiento con el subconjunto de 853 muestras etiquetadas manualmente. Dado el enfoque del estudio en español, se optó por utilizar modelos de aprendizaje automático (ML) como máquinas de soporte vectorial y aprendizaje profundo (DL) como modelos neuronales con arquitectura LSTM y modelos basados en arquitecturas de Transformers. Todos los modelos fueron entrenados a 10 épocas, un tamaño de lote de 16 y una caída de peso de 0.01. Definiendo tales hiperparámetros, aseguramos el entrenamiento justo entre los modelos. Los procesos de entrenamiento y evaluación se realizaron en la nube, usando un mismo hardware para cada modelo.

3.7. Experimentos y resultados

Usando el subconjunto de comentarios y sus correspondientes etiquetas, mencionado previamente, se entrenaron distintos modelos de ML y DL. Los modelos de ML utilizados fueron bolsa de palabras y una combinación de bolsa de palabras con representación de texto ambos modelos construidos con apoyo de la biblioteca EvoMSA [7].

En cuanto a los modelos de DL, se entrenó una red LSTM básica, conocida por su efectividad en tareas de procesamiento de lenguaje natural. También se entrenaron 3

Tabla 1. Resultados de distintos modelos de clasificación de toxicidad en texto.

Modelo	Exactitud	Recall	F1
BETO-MP	0.9649	0.9649	0.9645
TextDetox XLMR	0.7894	0.7894	0.7942
dehateBERT	0.6783	0.6783	0.6031

modelos basados en Transformers. El primero que fue mBERT, es un modelo basado en BERT [8] preentrenado con un gran corpus en distintos idiomas. El segundo modelo fue XLM-RoBERTa, un modelo variante de RoBERTa [9] preentrenado en un corpus multilenguaje.

El tercer modelo utilizado fue BETO [10], que, aunque también comparte su arquitectura con BERT, este fue preentrenado con un corpus exclusivamente en español. La tabla 1 nos muestra que el mejor modelo para esta tarea fue BETO base, obteniendo resultados sobresalientas en cada métrica, por lo que se tomó este como referencia para las siguientes comparaciones. A partir de este punto y para efectos prácticos, nos referiremos al modelo seleccionado BETO, como BETO-MP.

Posterior a la selección del modelo, se comparó este con otros modelos que forman parte del estado del arte en clasificación binaria de toxicidad. El primero, un modelo XLM-RoBERTa entrenado con un corpus presentado en [11] para la tarea de clasificación binaria de toxicidad el cual es una compilación de diferentes corpus en varios lenguajes, incluyendo el español. El segundo, "dehateBERT" [12] un modelo basado en BERT multilenguaje el cual fue entrenado con un corpus en español de toxicidad. Los modelos se evaluaron con el 20% restante de muestras para pruebas.

Los resultados de la evaluación de cada modelo se presentan en la Tabla 2. Como se muestra en la tabla anterior, el modelo BETO-MP obtuvo resultados superiores con respecto a los otros modelos presentados en otros trabajos en la tarea de clasificación de comentarios tóxicos en la plataforma MisProfesores.

4. Conclusiones

Este estudio representa un avance significativo en la construcción de corpus en español para el análisis de sentimientos, específicamente en el contexto de comentarios en la plataforma MisProfesores. A través de un meticuloso proceso que incluyó la recolección, preprocesamiento, limpieza, y etiquetado de datos, se desarrolló un corpus único que refleja la diversidad y complejidad del español hablado en México. Este corpus no solo es relevante por su tamaño, con 18,000 muestras no etiquetadas y 853 muestras etiquetadas manualmente (Ambos corpus disponibles en el sitio[1]), sino también por su enfoque en capturar la riqueza lingüística y cultural específica de este contexto.

Los resultados de nuestro modelo superan otros modelos entrenados con otros corpus para la detección de toxicidad, evidenciando la importancia de la construcción y especialización de un corpus para esta tarea específica con el fin de mejorar la precisión en la detección de comentarios tóxicos en plataformas sociales académicas. Estos

[1] catalabs.mx/datasets/misprofesores/

hallazgos no solo contribuyen al campo académico del análisis de sentimientos y la IA, sino que también ofrecen aplicaciones prácticas para plataformas educativas en línea, ayudando a crear ambientes de aprendizaje más seguros y positivos.

Al detectar y manejar de manera proactiva los discursos de odio, se puede fomentar un intercambio de ideas más respetuoso y constructivo, crucial para el desarrollo educativo y social. Finalmente, este estudio subraya la necesidad de continuar expandiendo y refinando los corpus en idiomas distintos al inglés, adaptándolos a contextos específicos para mejorar la efectividad de los modelos de análisis de sentimientos. En el trabajo futuro se recomienda aumentar la cantidad de muestras etiquetadas manualmente y explorar otras plataformas y contextos, con el objetivo de desarrollar modelos más robustos y versátiles. La construcción de estos recursos no solo beneficia la investigación académica, también tienen un impacto directo en la sociedad, promoviendo entornos digitales más inclusivos y respetuosos.

Referencias

1. Tan, K.L., Lee, C.P., and Lim, K.M.: A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. Applied Sciences, vol. 13, no. 7, pp. 4550 (2023). DOI: 10.3390/ app13074550.
2. Wankhade, M., Rao, A.C.S., and Kulkarni, C.: A Survey on Sentiment Analysis Methods, Applications, and Challenges. Artificial Intelligence Review, vol. 55, no. 7, pp. 5731–5780 (2022). DOI: 10.1007/s10462-022-10144-1.
3. Arceo-Gomez, E.O., Campos-Vazquez, R.M.: Gender Stereotypes: The Case of misprofesores.com in Mexico. Economics of Education Review, vol. 72, pp. 55–65 (2019). DOI: 10.1016/j.econedurev.2019.05.007.
4. Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., and Taboada, M.: The Sfu Opinion and Comments Corpus: A Corpus for the Analysis of Online News Comments. Corpus Pragmatics, vol. 4, no. 2, pp. 155–190 (2020). DOI: 10.1007/s41701-019-00065-w.
5. Nabiilah, G.Z., Prasetyo, S.Y., Izdihar, Z.N., and Girsang, A.S.: Bert Base Model for Toxic Comment Analysis on Indonesian Social Media. Procedia Computer Science, vol. 216, pp. 714–721 (2023). DOI: 10.1016/j.procs.2022.12.188.
6. Asociación Nacional de Universidades e Instituciones de Educación Superior: Información estadística de educación superior, anuarios estadísticos de educación superior www.anuies.mx/informacion-y-servicios/informacion-estadistica-de-educacion-superior/ anuario-estadistico-de-educacion-superior (2024).
7. Graff, M., Miranda-Jimenez, S., Tellez, E.S., and Moctezuma, D.: Evomsa: A Multilingual Evolutionary Approach for Sentiment Analysis. In: IEEE Computational Intelligence Magazine, vol. 15, no. 1, pp. 76–88 (2020). DOI: 10.1109/mci.2019.2954668.
8. Devlin, J., Chang, M., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019). DOI: 10.18653/v1/n19-1423.
9. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V.: Unsupervised Cross-lingual Representation Learning at Scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2020). DOI: 10.18653/v1/2020.acl-main.747.
10. Cañete, J., Chaperon, G., Fuentes, R., Ho, J., Kang, H., and Pérez, J.: Spanish Pretrained BERT Model and Evaluation Data (2023). DOI: 10.48550/arXiv.2308.02976.
11. PAN: Multilingual Text Detoxification (TextDetox). http://pan.webis.de/clef24/pan24-web/text-detoxification.html#task (2024).

12. Aluru, S.S., Mathew, B., Saha, P., and Mukherjee, A.: Deep Learning Models for Multilingual Hate Speech Detection. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 1–16 (2020). DOI: 10.48550/arXiv.2004.06465.

ISSN 1870-4069

Máquinas de soporte vectorial enpredicción de falla cardíaca

María Dolores Torres Soto[1], Aurora Torres Soto[2]

[1] Universidad Autónoma de Aguascalientes,
Sistemas de Información,
México

[2] Universidad Autónoma de Aguascalientes,
Ciencias de la Computación,
México

{dolores.torres, aurora.torres}@edu.uaa.mx

Resumen. En este trabajo, se presenta un clasificador de falla cardíaca basado en máquinas de soporte vectorial cuyo objetivo fue que utilizando datos clínicos lograra una exactitud por encima del 80%. Se configuró un diseño de experimentos factorial para encontrar los mejores parámetros para la máquina de soporte vectorial haciendo modificaciones en el kernel y el parámetro C. Los resultados obtenidos superaron las expectativas, pues el modelo resultante presenta una exactitud superior al 95% y una precisión de 94.44%. En este trabajo, fue usada la base de datos "Predicción de Falla Cardiaca" del repositorio Kaggle. La falla cardíaca es una patología crónica y degenerativa, por lo que un diagnóstico temprano puede ser la diferencia entre la vida y la muerte. Contar con una herramienta de aprendizaje supervisado de diagnóstico temprano con una buena exactitud, es un paso importante hacia la prevención.

Palabras clave: Falla cardíaca, máquinas de soporte vectorial, aprendizaje supervisado, aprendizaje automático

Support Vector Machines in Heart Failure Prediction

Abstract. In this work, a heart failure classifier based on support vector machines is presented whose objective was to achieve an accuracy above 80% using clinical data. A factorial design of experiments was configured to find the best parameters for the support vector machine by making modifications to the kernel and parameter C. The results obtained exceeded expectations, since the resulting model has an accuracy greater than 95% and precision of 94.44%. In this work, the "Heart Failure Prediction" database from the Kaggle repository was used. Heart failure is a chronic and degenerative pathology, so early diagnosis can be the difference between life and death. Having a supervised learning tool for early diagnosis with good accuracy is an important step towards prevention.

Keywords: Heart failure, support vector machines, supervised learning, machine learning.

1. Introducción

En este trabajo, se presenta un clasificador de "Falla Cardíaca" basado en máquinas de soporte vectorial. El clasificador, pronostica esta patología médica mediante datos clínicos con una exactitud por encima del 95%. En el ámbito mundial, la falla cardíaca es una condición médica importante que afecta a millones de personas y representa una carga significativa para los sistemas de salud.

Según la Organización Mundial de la Salud (OMS), se estima que más de 26 millones de personas en todo el mundo viven con falla cardíaca en alguna medida, y se prevé que esta cifra aumente debido al envejecimiento de la población y la mayor incidencia de factores de riesgo cardiovascular [1]. Una detección temprana de este tipo de patología puede ser la diferencia entre la vida y la muerte, pues cuando una falla cardíaca comienza, el organismo compensa las consecuencias por cierto tiempo, hasta el punto donde esto ya no le es posible [2].

Si se detecta de manera temprana, el paciente puede comenzar con su tratamiento y prolongar su vida de manera considerable. El objetivo principal de este trabajo, consiste en diseñar, implementar y afinar (en términos de la selección de los mejores parámetros) un algoritmo Support Vector Machine (SVM) para el pronóstico de falla cardíaca que tenga una certeza de por lo menos el 80%, basado en datos clínicos con la intención de ser aplicado a grandes conjuntos de personas para lograr una identificación temprana de esta patología.

La falla cardíaca es el tema de interés de este trabajo porque se trata de una de las enfermedades que cobra una mayor cantidad de muertes alrededor del mundo. Por tratase muchas veces de una patología asintomática, crónica y degenerativa, es fácil que el paciente que la padece no lo sepa y no busque ayuda médica.

Por lo anterior, el contar con una herramienta confiable de detección temprana, puede ser de gran utilidad para combatirla a tiempo. En las siguientes secciones de este documento, se podrá revisar la justificación, algunos conceptos de interés, la metodología utilizada, los experimentos realizados, los resultados y las conclusiones de la investigación.

1.1. Justificación

A nivel mundial, las cardiopatías, son desde hace 20 años la causa principal de mortalidad en todo el mundo. Actualmente provocan más muertes que nunca. El número de muertes debidas a las cardiopatías ha aumentado desde 2000 en más de 2 millones de personas, hasta llegar a casi 9 millones de personas en 2019 y continúan con esta tendencia. Las cardiopatías representan en 2019, el 16% del total de muertes debidas a todas las causas médicas. Más de la mitad de los dos millones de muertes adicionales han ocurrido en la región del pacífico occidental de la OMS. [1].

En México, cerca de 220 mil personas fallecieron por enfermedades cardiovasculares en 2021, de las cuales 177 mil fueron por infarto al miocardio. Estas muertes pueden ser prevenibles al evitar o controlar los factores de riesgo como el

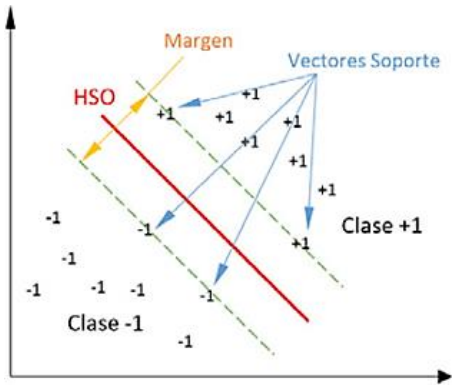


Fig. 1. Máquina de soporte vectorial [7].

tabaquismo, presión arterial alta, colesterol elevado y diabetes no controlada [3]. La prevención es una opción excelente para este tipo de patologías pues se trata de problemas de salud crónicos y degenerativos.

2. Conceptos de interés

2.1. Falla cardíaca

La insuficiencia cardíaca (IC) o falla cardíaca (FC), significa que el corazón no puede bombear suficiente sangre rica en oxígeno para abastecer a los órganos, músculos y tejidos del organismo [4]. El tener insuficiencia cardíaca no significa que el corazón se haya detenido o esté a punto de dejar de latir. Pero sin suficiente flujo de sangre, es posible que los órganos no funcionen bien, lo que puede causar problemas de salud graves [2].

A veces, se utiliza la palabra "falla" para describir la etapa final de la enfermedad, mientras que "insuficiencia" se usa de manera más general para describir la condición en sus diferentes etapas y grados de severidad. Por ejemplo, se puede hablar de "insuficiencia cardíaca congestiva" para referirse a la condición en la que el corazón tiene dificultad para bombear sangre y se produce una acumulación de líquido en los tejidos:

- Cuando el corazón empieza a fallar, el organismo lo detecta inmediatamente y pone en marcha ciertos mecanismos compensatorios, por lo que muchos pacientes no llegan a percibir los síntomas anormales que manifiesta el corazón. No obstante, los mecanismos compensatorios sólo son eficaces durante cierto tiempo, por lo que llega un momento en el que el organismo no puede compensar más el fallo en el bombeo del corazón.
- El síndrome de insuficiencia cardíaca es un problema sanitario de primer orden. Las causas que lo generan son muy variadas, pero en el mundo occidental las más frecuentes son las derivadas de la enfermedad isquémica, diabetes, hipertensión y miocardiopatía de diversos orígenes [5].

- La causa más común de IC es la enfermedad coronaria, como la angina de pecho y, especialmente, el infarto de miocardio. Otra causa habitual es la hipertensión arterial [4].
- Hay dos tipos de insuficiencia cardíaca:
- Insuficiencia cardíaca sistólica. Se produce cuando disminuye la capacidad de contracción del corazón. No se empuja con suficiente fuerza a la sangre hacia el resto del cuerpo y queda en la cavidad cardíaca. A causa de esto, la sangre no puede entrar en el corazón y queda acumulada en los pulmones. Es lo que se denomina congestión pulmonar.
- Insuficiencia cardíaca diastólica. El corazón no recibe la suficiente cantidad de sangre porque tiene problemas para distenderse. Esto produce acumulación de fluidos en pies, tobillos y piernas. Algunos pacientes pueden tener también congestión pulmonar [4].

2.2. Máquinas de soporte vectorial

Una Máquina de Soporte Vectorial (SVM por las siglas de Support Vector Machine), es un conjunto de algoritmos de aprendizaje supervisado desarrollado por Vladimir Vapnik y sus colaboradores en los laboratorios AT&T. Una SVM mapea los puntos de entrada a un espacio de características de una dimensión mayor a la que representa el problema original, (i.e., si los puntos de entrada están en $\Re^2$, entonces son mapeados a $\Re^3$) [6], y la SVM encuentra un hiperplano (HSO) que separe los objetos que pertenecen a clases distintas y que maximice el margen de separación entre clases, como se aprecia en la Fig. 1.

Una máquina de soporte vectorial puede modificar su comportamiento de acuerdo con la modificación de ciertos ajustes de componentes de la misma. Entre los principales parámetros que pueden configurarse en una SVM se encuentran: el kernel, el parámetro C y el parámetro gamma (γ) [8]. El Kernel permite determinar el tipo de transformación que se aplica a los datos de entrada para separar las diferentes clases. Los tipos comunes de kernel incluyen el lineal, polinómico y radial (RBF). El kernel lineal se utiliza para datos linealmente separables, mientras que los kernels polinómicos y radiales son útiles para datos de carácter no lineal [8].

El parámetro *C* controla el número y severidad de las violaciones del margen (y del hiperplano) que se toleran en el proceso de ajuste. Si $C = \infty$, no se permite ninguna violación del margen y por lo tanto, el resultado es equivalente al clasificador de margen maximal (teniendo en cuenta que esta solución solo es posible si las clases son perfectamente separables). Cuando C se aproxima a cero, los errores se penalizan menos y más observaciones pueden estar en el lado incorrecto del margen o incluso del hiperplano. C es, a fin de cuentas, el hiperparámetro encargado de controlar el balance entre bias (sesgo) y la varianza del modelo.

El bias es el sesgo del modelo, es un parámetro adicional en una SVM que permite desplazar la frontera de decisión para mejorar la capacidad de ajuste del modelo a los datos de entrenamiento. El proceso de optimización de una SVM tiene la peculiaridad de que solo las observaciones que se encuentran justo en el margen o que lo violan influyen sobre el hiperplano de separación.

A estas observaciones se denomina vectores soporte y son las que definen el clasificador obtenido. Esta es la razón por la que el parámetro C controla el balance entre bias y varianza. Cuando el valor de C es pequeño, el margen es más ancho, y más observaciones violan el margen, convirtiéndose en vectores soporte. El hiperplano está, por lo tanto, sustentado por más observaciones, lo que aumenta el bias pero reduce la varianza. Cuando mayor es el valor de C, menor el margen, menos observaciones son vectores soporte y el clasificador resultante tiene menor bias pero mayor varianza. [8]. El lector interesado, podrá encontrar mayor detalle en [9, 10].

Por otro lado, gamma (γ) es un parámetro que se aplica específicamente en kernels no lineales como el RBF. Gamma controla la influencia de cada punto de entrenamiento en la definición de la frontera de decisión. Un valor pequeño de gamma indica un alcance más amplio de influencia y un modelo más suave, mientras que un valor alto de gamma da como resultado un modelo más ajustado a los datos de entrenamiento. En esta sección, se ha enfatizado la manera teórica en la que una máquina de soporte vectorial responde ante cambios en los parámetros kernel, C *y* gamma porque estos parámetros son los que ajustamos para el mejor funcionamiento del clasificador desarrollado como podrá ser revisado en el apartado de "Experimentos" de este documento.

3. Metodología

En este espacio, el lector podrá revisar la base de datos utilizada para la realización de este estudio, definición operacional y conceptual de las características, el preprocesamiento realizado y el detalle de la metodología empleada. A continuación, se presenta el conjunto de datos empleado para la prueba de los algoritmos.

3.1. Conjunto de datos

El conjunto de datos empleado para la investigación se denomina: "Predicción de Falla Cardíaca" [11]. Recuperada del repositorio Kaggle. Se trata de una base de datos proporcionada por Diego Vergara en 2022, que contiene información de 1190 casos de pacientes de Cleveland, Suiza, Hungría, Long Beach entre otros y de los cuáles hay 272 duplicados, por lo que el conjunto final utilizado es de 918 casos.

De éstos, el 80% (734) forman el conjunto de entrenamiento y el 20% (184) se usan para validación. De los 918 casos, (508) el 55,34% son pacientes con falla cardíaca y (410) el 44.66% son sanos. Por supuesto, un criterio de exclusión fue la repetición del caso. En el conjunto resultante, se tienen pacientes con o sin falla cardíaca sin repetición. Las características de los casos, su definición conceptual y operacional respectivamente son:

1. Edad. Edad del paciente en años.
2. Género. Género del paciente [M: Masculino, F: Femenino] . Valores 0-1 en ese orden.
3. Tipo de Dolor de Pecho. [TA: Angina Típica, ATA: Angina Atípica, NAP: Dolor no anginal, ASY: Asintomático]. Valores de 0–3 en ese orden.
4. Presión Arterial en Reposo. Se mide en milímetros de mercurio [mm Hg]

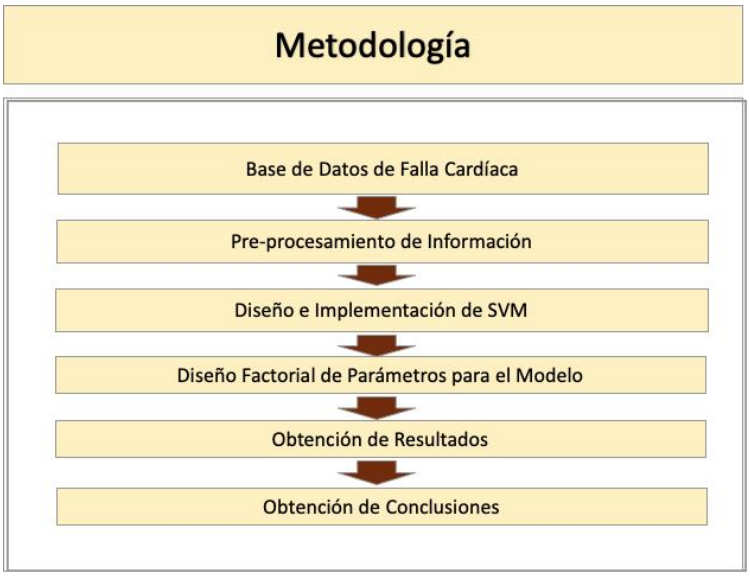


Fig. 2. Metodología.

5. Colesterol. Nivel de colesterol sérico [mm/dl]
6. Glucemia en Ayunas. Se presenta como: [1: Si FastingBS > 120 mg/dl, 0: de lo contrario]
7. Resultados de Electrocardiograma en Reposo. Puede ser: [Normal: Normal, ST: anormalidad en onda ST-T (inversión en onda T y/o elevación o depresión de la onda ST > 0.05 mV), LVH: que muestra hipertrofia. ventricular izquierda probable o definitiva según los criterios de Estes]. Valores 0-3 en ese orden.
8. Frecuencia cardíaca máxima. [Valor numérico entre 60 y 202].
9. Angina inducida por ejercicio. Puede ser: [Y: Si, N: No]. 0 y 1 en ese orden.
10. Oldpeak ó depresión del segmento ST en un ECG especialmente durante el ejercicio = ST [Valor numérico].
11. Pendiente del segmento ST de ejercicio máximo. Puede ser: [Up: ascendente, Flat: plana, Down: descendente]. Valores 0-2 en ese orden.
12. Enfermedad Cardíaca: Es la clase, y puede ser: [1: enfermedad cardiaca, 0: Normal].

3.2. Metodología utilizada

En la fig.2, se presenta el detalle de la metodología general de esta investigación. El lector podrá ver que la primera fase consiste en la obtención de la base de datos, que fue descargada del repositorio Kaggle [11].

Posteriormente, se prepararon los conjuntos de entrenamiento y validación con un 80% y un 20% respectivamente. Como tercera etapa, una máquina de soporte vectorial fue diseñada e implementada en lenguaje Python 3.7 con apoyo de las librerías Pandas

Tabla 1. Niveles de parámetros para experimentación.

Parámetro	Valor
Kernel	Lineal
	Poly
	RBF
C	1.2
	4.0
	6.0
Gamma	0.09

y sklearn. Como cuarta etapa de la metodología y con la intención de contar con un mejor desempeño para la máquina de soporte vectorial, se trabajó en un diseño factorial de 30 réplicas para cada grupo de combinación de parámetros. Contando con este diseño factorial, se comenzó a hacer el análisis de resultados para completar la fase cinco. Finalmente, como última etapa, se obtienen las principales conclusiones del estudio y se considera el trabajo futuro.

4. Experimentación

En esta sección, se presentan los detalles del experimento factorial diseñado para seleccionar los mejores parámetros de la máquina de soporte vectorial.

4.1. Experimentos con la máquina de soporte vectorial

Como primer conjunto de experimentos, se corrieron pruebas empíricas en la máquina de soporte vectorial para conocer de manera aproximada los valores de los parámetros a configurar. En este sentido, se trabajó con 3 parámetros: Kernel, C y gamma. Recordemos que el parámetro C, en una SVM controla la penalización por errores en la clasificación. Un valor de C bajo permite una mayor flexibilidad en la selección del hiperplano de separación, lo que puede resultar en un margen más amplio, pero también en una mayor tolerancia a errores de clasificación en el conjunto de entrenamiento.

Un valor del parámetro C bajo puede llevar a un modelo con alta capacidad de generalización, pero mayor susceptibilidad al sobreajuste, mientras que un valor alto para C, puede resultar en un modelo con menor capacidad de generalización, pero mayor capacidad para ajustarse a los datos de entrenamiento.

Por otro lado, un valor alto de C impone una mayor penalización por errores, lo que conduce a un hiperplano de separación más ajustado y posiblemente más sensible a ruido en los datos. En cuanto al kernel seleccionado, éste permite la transformación de un problema a un espacio *n*-dimensional por encima del espacio de datos original para lograr la separación lineal de las clases.

El kernel lineal es adecuado para conjuntos de datos linealmente separables, mientras que los kernels polinómicos y radiales son más adecuados para conjuntos de datos no lineales. Es conocido que el kernel radial (RBF) es especialmente versátil y

Tabla 2. Grupos de parámetros para experimentación.

Grupo	Kernel	C	Gamma
1-3	RBF Lineal Poly	1.2	0.09
4-6	RBF Lineal Poly	4.0	0.09
7-9	RBF Lineal Poly	6.0	0.09

puede manejar relaciones no lineales de manera efectiva [8]. Con respecto del parámetro gamma, recordemos que este parámetro controla la influencia de cada punto de entrenamiento en la definición de la frontera de decisión.

Un valor pequeño de gamma indica un alcance más amplio de influencia y un modelo más suave, mientras que un valor alto de gamma da como resultado un modelo más ajustado a los datos de entrenamiento. Por defecto, gamma es $1/n$ establecido por scikit-sklearn [8], donde n es el número de característica que se utilizan para la clasificación.

Tomando en consideración el efecto teórico de los parámetros y el resultado deseado del modelo clasificador, los valores para los parámetros mostrados en la Tabla 1, fueron seleccionados después de un análisis empírico inicial: Como se trata de dos parámetros con tres niveles de valor para cada uno, se crearon 9 grupos de parámetros para la experimentación como puede observarse en la Tabla 2.

Aunque se probaron 2 valores de gamma, (0.009 y 0.09), se observó que, en todos los casos, los mejores resultados corresponden a 0.09, que es el valor recomendado por Scikit-learn. Por tal motivo, esta parte se descartó de los experimentos y se fijó gamma en 0.09. Se desarrollaron 30 réplicas con cada uno de los grupos de parámetros y los resultados fueron analizados con el software estadístico SPSS versión 20.

Como puede verse en la tabla 2. El grupo 1 tiene en el mismo orden de aparición de la tabla los parámetros: kernel = rbf, $C = 1.2$, Gamma $= 0.09$, el segundo grupo tiene: Kernel = lineal, $C = 1.2$, Gamma $= 0.09$, el grupo 3 tiene kernel = poly, $C = 1.2$ y Gamma $= 0.09$. De esta misma manera el grupo 4 tiene kernel = rbf, $C = 4$ y Gamma $= 0.09$, el grupo5 tiene kernel = lineal, $C = 4$ y Gamma $= 0.09$, el grupo 6 tiene kernel = poly, $C = 4$ y Gamma $= 0.09$, ocurriendo en el mismo orden para los grupos 7, 8 y 9 con $C = 6.0$ y kernels rbf, lineal y poli respectivamente.

Las variables de desempeño consideradas para el algoritmo fueron: exactitud del modelo en validación (E), tiempo en segundos (T), verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN).

Utilizando estas características de desempeño, se establecieron los mejores valores de los parámetros y se procedió a la obtención de resultados y conclusiones de la investigación.

Tabla 3. Resultados de la experimentación.

Grupo	Exactitud Modelo	Tiempo (seg)	Verdaderos Positivos (VP)	Verdaderos Negativos (VN)	Falsos Positivos (FP)	Falsos Negativos (FN)
1	0.955434783	0.027	103.7	72.1	6.1	2.1
2	0.851086957	4.379	87.5	69.1	13.3	14.1
3	0.736956522	0.0233	83.3	52.3	20.6	27.8
4	0.744565217	0.0281	76.3	60.7	24.2	22.8
5	0.873369565	18.013	91.6	69.1	11.8	11.5
6	0.741847826	0.0275	81.9	54.6	23	24.5
7	0.745652174	0.0272	78.6	58.6	23.5	23.3
8	0.858695652	25.894	89.6	68.4	12.5	13.5
9	0.759782609	0.0316	81.7	58.1	19.1	25.1

5. Resultados

Como puede observarse en la Tabla 3. Los resultados apuntan a la combinación de parámetros correspondiente al grupo 1, como la mejor en términos de exactitud del modelo.

En cuanto al tiempo en segundos, también se encuentra como una de las mejores combinaciones de parámetros y respecto de verdaderos positivos, verdaderos negativos, falsos positivos (error tipo I) y falsos negativos (error tipo II), es la mejor tanto empírica como estadísticamente. Cada grupo corresponde con los primeros parámetros mostrados en la Tabla 2.

En el caso del grupo 1, usa un kernel $= RBF$, $C = 1.2$ y gamma $= 0.09$. Los grupos se conforman en el mismo orden de aparición de los parámetros en la Tabla 2.

El único criterio de desempeño del modelo que presentó una distribución normal y homocedástica fue el tiempo, la exactitud, VP, VN, FP, FN no cumplen con los criterios de normalidad y homocedasticidad, por tal motivo, se realizó la prueba de Kruskall Wallis con el fin de encontrar si las diferencias en los grupos de parámetros eran estadísticamente significativas.

Los resultados estadísticos mostraron que, en efecto, existe diferencia entre los grupos de combinaciones variando el kernel y el parámetro C. Como puede verse en la Tabla 3, el grupo 1 de parámetros tuvo los mejores resultados para la máquina de soporte vectorial.

Así pues, los mejores resultados promedio se obtuvieron con un kernel *RBF* y una $C = 1.2$ logrando una exactitud del modelo en validación del 95.54%, una precisión del 94.44%, un tiempo en segundos de 0.027, 103.7 VP, 72.1 VN, un error tipo I de 6.1 y un error tipo II de 2.1.

De estos resultados podemos concluir que, en efecto, vale la pena dedicar tiempo a la configuración de los parámetros de cualquier algoritmo de aprendizaje. Cabe mencionar que, con el mejor modelo, se realizó una validación cruzada de 5 pliegues obteniendo un promedio de exactitud 95.02%. Los valores de los 5 pliegues fueron: 94.3, 94.2, 95.4, 95.7 y 95.5 respectivamente.

6. Conclusiones y discusión

Aunque inicialmente se buscaba diseñar e implementar un clasificador de falla cardíaca que alcanzara una exactitud del 80%, este objetivo se mejoró por más de 15 puntos porcentuales alcanzando una exactitud de más del 95%. El ejercicio de diseño y desarrollo de un clasificador de falla cardíaca con datos reales de pacientes de Cleveland, Suiza, Hungría, Long Beach entre otros, es un paso importante que permite ver los resultados de un buen ajuste de parámetros en algoritmos de aprendizaje automático como es el caso de las máquinas de soporte vectorial. Es importante enfatizar que, para este problema, el kernel que mejores resultados proporciona es el rbf y el que peores resultados tuvo fue el poly. Por otro lado, un valor de C=1.2 proporcionó buenos resultados en contraposición con los demás valores probados (4.0 y 6.0).

La falla cardíaca es una patología crónica y degenerativa, por lo que un diagnóstico temprano puede ser la diferencia entre la vida y la muerte. Contar con una herramienta de aprendizaje supervisado de diagnóstico temprano con una buena exactitud (mayor al 95%), es un paso importante hacia la prevención de este tipo de enfermedades. Los resultados de la experimentación empírica presentaban una exactitud promedio del 80% que fue superada por mucho en los experimentos finales. El ajuste de parámetros puede significar la diferencia entre contar con un modelo adecuado y tener un excelente modelo para el pronóstico de una de las enfermedades que a nivel mundial representa la principal causa de muerte en el mundo y en México.

7. Trabajo futuro

Con los resultados obtenidos en esta investigación, se pretende entrenar una red neuronal artificial, someterla a un diseño factorial para escoger los mejores parámetros y contrastar sus resultados contra la máquina de soporte vectorial. Posteriormente, se ejecutarán ambos clasificadores con datos reales mexicanos obtenidos en colaboración con médicos expertos del sector salud del estado de Aguascalientes, México. Con estos modelos adaptados para la población mexicana, se llevarán a cabo aplicaciones masivas en el sector salud del Estado de Aguascalientes.

Referencias

1. WHO: Organización Mundial de la Salud: La OMS revela las principales causas de muerte y discapacidad en el mundo: 2000-2019. www.who.int/es/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019 (2020)
2. American Heart Association: ¿Qué es la insuficiencia cardiaca? www.heart.org/en/health-topics/heart-failure/what-is-heart-failure (2016).
3. Gobierno Federal: Comunicado de Salud. Secretaría de Salud (2021)
4. Colomer-Carretero, M.: Insuficiencia cardíaca tratamiento con agentes sensibilizadores del calcio. Offarm Farmacia y Sociedad, vol. 25, no. 7, pp. 87–87 (2006)
5. Imizcoz, M.Á.: Insuficiencia cardíaca: Definición, fisiopatología y cambios estructurales. Cirugía Cardiovascular, vol. 15, no. 1, pp. 15–20 (2008). DOI: 10.1016/s1134-0096(08) 70220-1.

6. Betancourt, G.: Las máquinas de soporte vectorial (SVMs). Scientia et Technica, vol. XI, no. 27, pp. 67–72 (2005)
7. González, R., Barrientos, A., Toapanta, M., del-Cerro, J.: Aplicación de las máquinas de soporte vectorial (SVM) al diagnóstico clínico de la enfermedad de párkinson y el temblor esencial. Revista Iberoamericana de Automática e Informática Industrial (RIAI), vol. 14, no. 4, pp. 394–405 (2017). DOI: 10.1016/j.riai.2017.07.005.
8. Scikit-learn: sklearn.svm.SVC. Scikitlearn. http://scikit-learn.org/0.15/modules/generated/sklearn.svm.SVC.html (2019)
9. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer New York (2000). DOI: 10.1007/978-1-4757-3264-1.
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000). DOI: 10.1017/CBO 9780511801389.
11. Vergara, D.: Predicción de falla cardiaca. Kaggle. http://kaggle.com/competitions/prediccin-de-falla-cardiaca (2022)

ISSN 1870-4069

Desarrollo de un modelo digital didáctico de lechugas aeropónicas

Raul O. Herrera-Arroyo[1], Juan J. Martínez-Nolasco[1], José E. Botello-Álvarez[1], Mauro Santoyo-Mora[1], Ricardo Yánez-López[2]

[1] Tecnológico Nacional de México, Guanajuato, México

[2] Instituto Tecnológico de Roque, Guanajuato, México

{M2303045, juan.martinez, enrique.botello, mauro.santoyo}@itcelaya.edu.mx, ricardo.yl@roque.tecnm.mx

Resumen. El aumento proyectado de la población mundial en conjunto con los desafíos del cambio climático, presentan nuevos retos para la seguridad alimentaria, especialmente en el sector agrícola. Ante esta situación se vuelve de suma importancia el desarrollar tecnologías avanzadas como cultivo sin suelo y la agricultura indoor. Estas técnicas permiten un uso más eficiente de los recursos al suministrar los nutrientes directamente a las plantas, reduciendo significativamente el desperdicio de agua y nutrientes. Los gemelos digitales surgen como una herramienta crucial en la búsqueda de soluciones para los desafíos de diversas áreas, en este caso es la agricultura, ya que permiten simular y analizar el comportamiento de los sistemas agrícolas en entornos cerrados. Un modelo digital es una representación cibernética de lo real, creada manualmente y desconectada de la realidad, ya que, a diferencia de un gemelo digital, en este solo se le introducen datos de manera manual. El objetivo de este trabajo es desarrollar un modelo digital de una lechuga aeropónica como herramienta didáctica para una mejor comprensión de cómo afectan las variables ambientales al crecimiento de las plantas en ambientes controlados. Los resultados muestran un modelo digital funcional que permite a los usuarios explorar las afectaciones de las variables ambientales como la temperatura y humedad relativa al crecimiento de las plantas.

Palabras clave: Modelo digital, cultivos indoor, temperatura, humedad relativa.

Development of an Educational Digital Model of Aeroponic Lettuces

Abstract. The projected increase in the world population, along with the challenges of climate change, presents new challenges for food security, especially in the agricultural sector. Given this situation, it becomes of paramount

importance to develop advanced technologies such as soilless cultivation and indoor agriculture. These techniques allow for a more efficient use of resources by supplying nutrients directly to the plants, significantly reducing water and nutrient waste. Digital twins emerge as a crucial tool in the search for solutions to challenges in various areas, in this case, agriculture, as they enable the simulation and analysis of the behavior of agricultural systems in closed environments. A digital model is a cybernetic representation of reality, created manually and disconnected from reality since, unlike a digital twin, only manual data input is provided. The objective of this work is to develop a digital model of aeroponic lettuce as a didactic tool for a better understanding of how environmental variables affect plant growth in controlled environments. The results show a functional digital model that allows users to explore the effects of environmental variables such as temperature and relative humidity on plant growth.

Keywords: Digital model, indoor crops, temperature, relative humidity.

1. Introducción

Para el año 2030, se espera que la población mundial alcance aproximadamente los 8.5 mil millones de personas, aumentando 1.18 mil millones más durante las próximas dos décadas, acercándose a los 9.7 mil millones de personas para el año 2050 [1], aunado al cambio climático, uno de los escenarios de preocupación global más grandes, es en el sector agrícola, ya que la producción suficiente y el suministro de alimentos se ven afectados [2-3]. Debido a estas circunstancias críticas se ha vuelto esencial desarrollar tecnologías y técnicas avanzadas para superar esta situación como lo son los cultivos sin suelo y la agricultura interior [4].

Los cultivos sin suelo, son sistemas de producción agrícola donde las plantas son cultivadas sin necesidad de tierra, en su lugar se utilizan soluciones nutritivas con todos los nutrientes necesarios para el correcto crecimiento de las plantas, un ejemplo de esta técnica es la hidroponía, en la cual, los nutrientes necesarios son suministrados a través de agua de riego y son absorbidos directamente desde las raíces, la solución no absorbida es recirculada, evitando desperdicios de agua y nutrientes [5].

La aeroponía es un método de cultivos sin suelo, donde las raíces de las plantas se suspenden en el aire, los nutrientes son suministrados por la atomización de solución nutritiva directamente a las raíces, logrando un crecimiento rápido y saludable [6]. La agricultura interior se refiere a las prácticas de cultivar plantas en ambientes cerrados, normalmente se utilizan sistemas de iluminación artificial, control de temperatura, humedad relativa y nutrientes, generando un ambiente óptimo para el desarrollo de los cultivos, con lo que se puede obtener mayor producción por unidad de área comparándola con agricultura en invernaderos [7].

La generación de este entorno controlado, junto al suministro de los nutrientes disueltos en el agua, conlleva un desafío relacionado con el consumo de energía, al ser un sistema automatizado [8]. Un Gemelo digital, es el equivalente digital de un objeto de la vida real, del cual refleja su comportamiento y estados a lo largo de su vida en el espacio virtual, permitiendo simular cambios en sus contrapartes físicas agregando análisis de datos avanzados como machine learning y modelos de predicción [9]. Un modelo digital es un reflejo de la realidad que se crea de manera manual y funciona

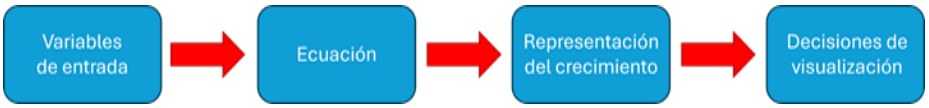


Fig. 1. Diagrama de la metodología propuesta.

Tabla 1. Valores mínimos, ideales y máximos de temperatura y humedad relativa para el crecimiento de lechugas.

Parámetros	Valores
Temperatura	Mínima: 7º C Ideal: 18–22º C Máxima: 28º C
Humedad relativa	Mínima: 40% Ideal: 70–80% Máxima: 85%

offline, es decir, el modelo digital no cambia cuando la realidad lo hace, en contrario, un gemelo digital, cambia cuando la realidad lo hace [10].

El objetivo de este trabajo fue el desarrollar un modelo digital de una lechuga aeropónica que pueda ser utilizado para el entendimiento temprano de cómo afectan las variables ambientales al crecimiento de una planta en un ambiente controlado. Cabe mencionar que los usos de los modelos digitales son muy diversos, siendo popular su uso en el sector energético, en el sector salud, logística, etc.

Con el desarrollo de este modelo digital didáctico, se contribuye a las nuevas aplicaciones de los modelos digitales y los beneficios que pueden aportar en el área de las tecnologías de la agricultura. En la sección 2 se describen algunos trabajos relacionados con el mostrado en este artículo, donde se observan las diferentes aplicaciones que tienen los modelos digitales. En la sección 3, se muestra la metodología seguida para la elaboración del modelo digital. En la sección 4 se muestran los resultados obtenidos hasta el momento. Finalmente, en la sección 5, se muestran las conclusiones a las que se llegaron, así como el trabajo a futuro.

2. Trabajos relacionados

Howard et al [11], presentan los primeros avances en el desarrollo de un gemelo digital para la optimización energética de los invernaderos, se utilizó la plataforma de simulación IA AnyLogic para la generación del modelo de simulación que abarque los pasos de producción desde la entrega de las plántulas hasta el envío de las plantas finales. Utilizaron entrevistas para conseguir información sobre los patrones de comportamiento de los operadores, lo cual será complementado con diversos parámetros para obtener una simulación más precisa del flujo de producción.

El objetivo de su gemelo digital es poder proporcionar un análisis de la velocidad de producción en invernaderos, realizar un análisis de la eficiencia energética y el cálculo de los plazos esperados de producción. Mokhtar et al [12], aplicaron tres algoritmos de

(a) (b)

Fig. 2. Vista gráfica de la plántula (a), vista gráfica de la lechuga (b).

(a) (b) (c) (d)

Fig. 3. Estados utilizados de la lechuga en el modelo digital. En estado óptimo (a), estado saludable (b), estado poco saludable (c), estado seco (d).

machine learning a tres diferentes técnicas de cultivos hidropónicos (NTF, aeroponía de torre, aeroponía piramidal).

Se evaluaron diferentes combinaciones de variables de entrada como número de hojas, consumo de agua, peso seco, longitud del tallo y diámetro del tallo. El objetivo de este trabajo es la predicción del rendimiento de la lechuga obtenida (peso fresco). Reyes et al [13], desarrollan un gemelo digital de los lechos de cultivo de un sistema de acuaponía con el objetivo de monitorear parámetros como el pH, electro conductividad, temperatura del agua, temperatura ambiente e intensidad de la luz, predecir con el uso de inteligencia artificial y machine learning la tasa de crecimiento y el peso fresco de los cultivos en crecimiento, ayudando al usuario a tomar mejores decisiones para obtener un entorno acuapónico saludable.

Jeong et al [14], realizaron un gemelo digital de un establo para cerdos. Las instalaciones virtuales desarrolladas reflejan las instalaciones porcinas reales junto con sus componentes como el interior, exterior, dispositivos de detección y los cerdos. El objetivo de este trabajo es obtener una disminución del consumo energético dentro del establo, instrumentado con capacidad de mantener la temperatura estable por medio de bombas de calor, y sistemas de paneles radiantes, por lo que se llevaron a cabo múltiples simulaciones para la gestión de la granja en el entorno digital.

En el desarrollo de este trabajo se hizo uso del software de simulación EnergyPlus para la simulación de la calefacción, utilizando la función de auto dimensionamiento incluida en el software. Utilizando el método de control propuesto por el gemelo digital, obtuvieron una reducción del 26.80% del consumo energético dentro de las instalaciones.

Tabla 2. Valor del FCTH necesario para mostrar cada uno de los estados en pantalla.

Estado mostrado en pantalla	FCTH
Estado optimo	FCTH ≥ 0.9
Estado saludable	0.9 > FCTH > 0.5
Estado poco saludable	FCTH ≤ 0.5
Estado seco	FCTH ≤ 0.5 (en crecimiento máximo)

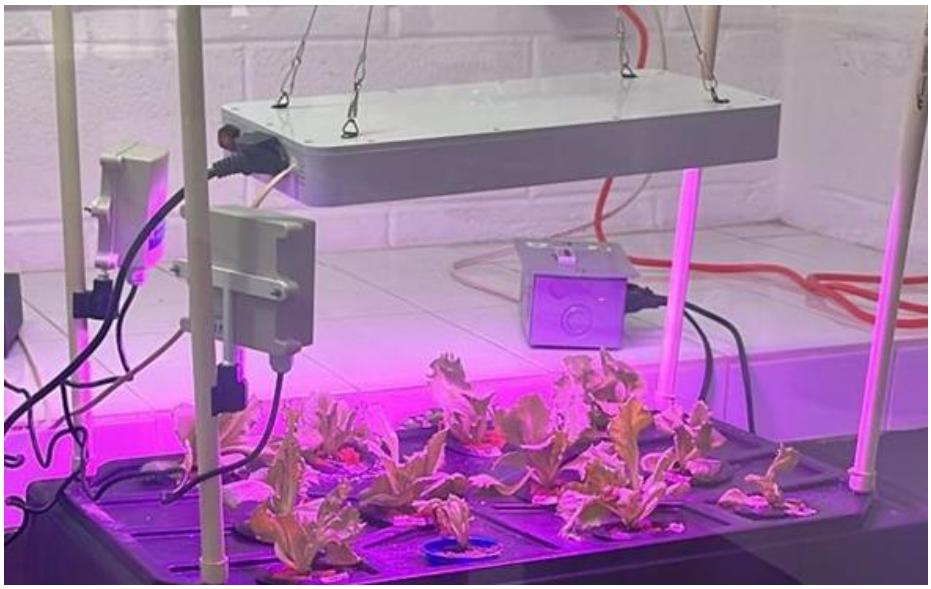


Fig. 4. Experimento realizado en cultivos interior para la obtención de valores óptimos de variables ambientales.

3. Metodología

Se propone el desarrollo de un modelo digital de tres lechugas aeropónicas, utilizando el motor gráfico Unity. Este modelo permitirá a los usuarios ingresar la temperatura y la humedad deseada, y observar cómo es afectado el crecimiento de las lechugas, facilitando la comprensión temprana de cómo estas variables influyen en el desarrollo de cultivos en interiores. En la metodología desarrollada en este proyecto de investigación están involucradas varias etapas, que, en conjunto, logran el objetivo en común, en este caso es el tener un modelo digital de una lechuga aeropónica.

El desarrollo del modelo digital se realizó utilizando el motor gráfico de Unity, ya que cuenta con la facilidad de tener una amplia documentación y el uso del lenguaje de programación C#. El modelo digital está basado en un prototipo de una cámara de crecimiento vegetal que se encuentra en el Tecnológico Nacional de México en Celaya. En la Fig. 1 se observa un diagrama de bloques con las etapas de la metodología propuesta.

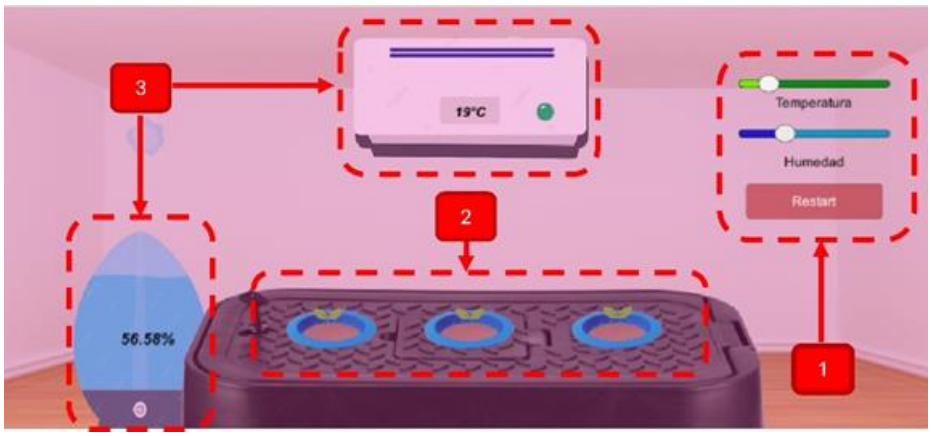


Fig. 4. Secciones del Modelo digital: Entrada de datos (1), visualización del estado de las lechugas (2), visualización numérica de los valores ingresados (3).

3.1. Variables de entrada

Por medio de revisión bibliográfica, se encontró que existen cuatro variables principales que infieren en el crecimiento de una planta: temperatura, humedad relativa en el ambiente, nivel del luminosidad y nivel de dióxido de carbono en el ambiente.

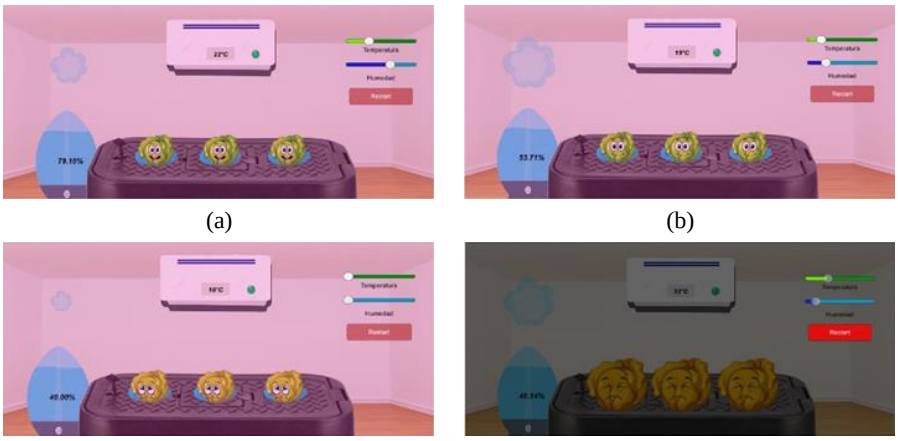
Debido a que en el modelo físico de la cámara de crecimiento no se encuentra controlado el nivel de dióxido de carbono, esta variable fue descartada al momento de realizar el modelo digital. El nivel de luminosidad presente en la cámara de crecimiento se mantuvo constante a 216 µmol/m2s con un fotoperiodo de 12 hrs, esta acción fue representada en el modelo digital por medio de una luz intermitente. Para la obtención de los valores de temperatura y humedad relativa ingresados por el usuario, se utilizaron deslizadores en la pantalla, con la finalidad de que el modelo digital sea didáctico, teniendo estos como limites la temperatura y la humedad relativa mínima y máxima que el prototipo físico puede representar.

3.2. Ecuaciones

En la tabla 1, se presentan los rangos de valores de las variables: temperatura y humedad relativa, para los cuales la lechuga tiene un correcto crecimiento. Los valores registrados fueron encontrados en literatura y comprobados con experimentación. Mediante software se realizaron ajustes lineales y parabólicos a los datos experimentales recopilados, se obtuvieron las ecuaciones que comparan las variables de temperatura y humedad relativa contra un factor decrecimiento de la lechuga, siendo un cero el factor mínimo de crecimiento y uno el factor máximo esperado:

$$\text{Factor de crecimiento por temperatura } = \frac{(\text{temperatura} - 26)^2}{-84} + 1.19, \quad (1)$$

$$\text{Factor de crecimiento por temperatura } = -0.07 \times \text{temperatura} + 2.5, \quad (2)$$



(a) (b)

(c) (d)

Fig. 5. Modelo digital en funcionamiento mostrando lechugas en estado óptimo (a), estado saludable (b), estado poco saludable (c), estado seco (d).

$$\text{Factor de crecimiento por humedad } = \frac{(\text{humedad} - 70)^2}{-800} + 1.13, \tag{3}$$

$$\text{Factor de crecimiento por humedad} = -0.05 \times \text{humedad} + 5. \tag{4}$$

La ecuación 1 se utiliza cuando la temperatura ingresada por el usuario es menor o igual a 22º C, mientras que la ecuación 2 se emplea cuando la temperatura supera ese valor, de igual manera la ecuación 3 se aplica cuando la humedad relativa ingresada es menor o igual a 80%, mientras que la ecuación 4 se utiliza cuando este valor se ve superado.

3.3. Representación del crecimiento

Para la representación del crecimiento de la planta, ver Fig. 2, se emplearon 2 vistas gráficas principales. Se utilizó la velocidad de fotogramas para aumentar el tamaño de la vista gráfica, realizando una multiplicación de los factores de crecimiento de temperatura y humedad relativa seleccionados por el usuario y sumándolo al vector de escala en cada fotograma. Como se ve en la Fig. 2, la primera vista gráfica utilizada representa la plántula o parte inicial de la vida de la planta, cuando la altura de la plántula llega a un límite máximo, la vista gráfica se deshabilita y se habilita la segunda vista gráfica que simula la forma de una lechuga desarrollada. El crecimiento de la lechuga se detiene cuando iguala o supera el límite máximo.

3.4. Decisiones de visualización

Debido a que busca que el modelo digital sirva como modelo didáctico para el entendimiento temprano de cómo afectan las variables ambientales en el crecimiento de una planta en un ambiente controlado, se utilizan modificaciones a la representación

gráfica de la lechuga desarrollada, en conjunto con expresiones faciales, para denotar el estado de esta. Como se observa en la Fig. 3, se presentan cuatro estados de salud de la lechuga, el primero de ellos representa a una lechuga en estado óptimo (a), la cual experimentará un crecimiento excepcional. El estado (b) simboliza una lechuga saludable, la cual crecerá normalmente.

El tercer estado expresa una lechuga en un estado poco saludable (c), ya que no cuenta con las condiciones para desarrollarse, por lo cual su crecimiento será muy poco. Finalmente, el ultimo estado, o estado seco (d), se mostrará en pantalla cuando el crecimiento máximo de la lechuga sea alcanzado en el estado poco saludable, representando que la lechuga se ha secado. Como se observa en la tabla 2, los estados se muestran en pantalla dependiendo del resultado de la multiplicación de los factores de crecimiento por temperatura y humedad relativa (FCTH).

4. Resultados

Se realizaron 2 experimentos de cultivo interior con ambiente controlado con una duración de 30 días cada uno (ver Fig. 4), donde se pusieron a prueba los valores de las variables ambientales que producen un mejor comportamiento en el desarrollo de una lechuga, encontrados en la documentación, como se muestra en la Tabla 1, facilitando el desarrollo del modelo digital que se presenta. Para a obtención de los valores de temperatura y humedad se utilizó el sensor JXBS-3001-TH—RS conectado a una placa de desarrollo ESP8266, los datos son subidos a un servidor en la nube para su resguardo y visualización. Se tomaron los valores de temperatura y humedad que produjeron un mayor peso fresco en la lechuga.

Como se muestra en las Figuras 5 y 6, se desarrolló un modelo digital de tres lechugas aeropónicas el cual tiene una interfaz gráfica amigable con el usuario con el objetivo de que se genere un entendimiento de cómo afectan las variables ambientales en el desarrollo de una lechuga haciendo uso de los cuatro estados de salud programados. El modelo digital cuenta con tres secciones principales, la sección 1 se encuentran dos deslizadores para la selección de la temperatura y humedad relativa a la cual se realizará el experimento y un botón de reinicio por si se busca repetir la ejecución del modelo digital.

En la sección 2, se puede visualizar el crecimiento de la lechuga y el estado de salud. Finalmente, en la sección 3 se observa de manera numérica los valores de temperatura y humedad relativa ingresados por el usuario. Mientras se encuentre en ejecución la simulación del modelo digital, los deslizadores están habilitados para cambiar la temperatura y la humedad relativa ingresadas, mientras el botón de reinicio se encuentra deshabilitado. Como se muestra en la Fig. 9, cuando las lechugas llegan a su tamaño máximo, independientemente del estado de salud, los deslizadores se deshabilitan y el botón de reinicio se habilita.

5. Conclusiones y trabajo a futuro

Los modelos digitales son herramientas muy útiles en muchas áreas, al ser representaciones computarizadas de sistemas físicos complejos, ya sea de agricultura,

ganadería, eléctrica, etc. Permiten realizar simulaciones, análisis y predicciones de manera eficiente y precisa, lo cual puede conllevar a mejoras significativas de los procesos. De igual manera se concluye que es posible la realización de modelos digitales didácticos los cuales ayudaran a un mejor entendimiento de los temas, sin la necesidad de realizar experimentación física y sin salir del hogar.

Durante el desarrollo del proyecto, el equipo mostró algunos inconvenientes para la ejecución del modelo digital, por lo que se sugiere el uso de un equipo de cómputo con características avanzadas. Como trabajo futuro, se planea sustituir las ecuaciones de comportamiento por algoritmos de aprendizaje automático, alimentándolos con una cantidad mayor de datos obtenidos de pruebas experimentales. Esto se debe a que el presente trabajo se llevó a cabo como un primer paso hacia la creación de un gemelo digital.

Referencias

1. United Nations: World Population Prospects 2022. Department of Economic and Social Affairs Population Division. http://population.un.org/wpp/ (2022)
2. Abbass, K., Qasim, M.Z., Song, H., Murshed, M., Mahmood, H., Younis, I.: A Review of the Global Climate Change Impacts, Adaptation, and Sustainable Mitigation Measures. Environmental Science and Pollution Research, vol. 29, no. 28, pp. 42539–42559 (2022). DOI: 10.1007/s11356-022-19718-6.
3. Contreras, J.: Retos alimentarios 2030: Objetivos, recomendaciones… alternativas y realidades. Journal of Behavior and Feeding, vol. 1, no. 1, pp. 86–95 (2021). DOI: 10.32870/jbf.v1i1.18.
4. Fasciolo, B., Awouda, A., Bruno, G., Lombardi, F.: A Smart Aeroponic System for Sustainable Indoor Farming. In: Proceedings of the Colloque International pour la Réduction des Coûts de Production, vol. 116, pp. 636–641 (2023). DOI: 10.1016/j.procir.2023.02.107.
5. International Society of Precision Agriculture: Precision Agriculture Definition. www.ispag.org/about/definition (2024)
6. García-Segura, D.R., Valdez-Aguilar, L.A., Ramírez-Rodríguez, H., Zermeño-González, A., Cadena-Zapata, M.: Producción de mini tubérculos de papa en aeroponía en comparación con suelo y polvo de coco. Revista Terra Latinoamericana, vol. 39 (2021). DOI: 10.28940/terra.v39i0.902.
7. Hati, A.J., Singh, R.R.: Smart Indoor Farms: Leveraging Technological Advancements to Power a Sustainable Agricultural Revolution. AgriEngineering, vol. 3, no. 4, pp. 728–767 (2021). DOI: 10.3390/agriengineering3040047.
8. González, J.P., Sanchez-Londoño, D., Barbieri, G.: A Monitoring Digital Twin for Services of Controlled Environment Agriculture. IFAC-PapersOnLine, vol. 55, no. 19, pp. 85–90 (2022). DOI: 10.1016/j.ifacol.2022.09.188.
9. Ariesen-Verschuur, N., Verdouw, C., Tekinerdogan, B.: Digital Twins in Greenhouse Horticulture: A Review. Computers and Electronics in Agriculture, vol. 199, pp. 107183 (2022). DOI: 10.1016/j.compag.2022.107183.
10. Van-der-Aalst, W.M.P., Hinz, O., Weinhardt, C.: Resilient Digital Twins: Organizations Need to Prepare for the Unexpected. Business and Information Systems Engineering, vol. 63, no. 6, pp. 615–619 (2021). DOI: 10.1007/s12599-021-00721-z.
11. Anthony-Howard, D., Ma, Z., Mazanti-Aaslyng, J., Norregaard-Jorgensen, B.: Data Architecture for Digital Twin of Commercial Greenhouse Production. In: Research and Innovation in the Fields of International Conference on Computing and Communication Technologies, pp. 1–7 (2020). DOI: 10.1109/rivf48685.2020.9140726.

12. Mokhtar, A., El-Ssawy, W., He, H., Al-Anasari, N., Sammen, S.S., Gyasi-Agyei, Y., Abuarab, M.: Using Machine Learning Models to Predict Hydroponically Grown Lettuce Yield. Frontiers in Plant Science, vol. 13 (2022). DOI: 10.3389/fpls.2022.706042.
13. Reyes-Yanes, A., Abbasi, R., Martinez, P., Ahmad, R.: Digital Twinning of Hydroponic Grow Beds in Intelligent Aquaponic Systems. Sensors, vol. 22, no. 19, pp. 7393 (2022). DOI: 10.3390/s22197393.
14. Jeong, D., Jo, S., Lee, I., Shin, H., Kim, J.: Digital Twin Application: Making a Virtual Pig House Toward Digital Livestock Farming. IEEE Access, vol. 11, pp. 121592–121602 (2023). DOI: 10.1109/access.2023.3313618.

ISSN 1870-4069

Minería de opiniones en el comercio electrónico usando n-gramas y algoritmos de aprendizaje automático

Francisco Antonio Castillo Velásquez, Maricarmen Rico Galeana, Nancy Sánchez Aguilar, José Marcos Zea Pérez, María del Consuelo Patricia Torres Falcón

Universidad Politécnica de Querétaro, México

{francisco.castillo, maricarmen.rico, nancy.sanchez}@upq.edu.mx
{marcos.zea, consuelo.torres}@upq.mx

Resumen. Actualmente hay 320 millones de usuarios de comercio electrónico en América Latina y el Caribe y se espera alcanzar 400 millones en el 2028. Solo las ventas netas de e-Commerce de las 100 principales tiendas online mexicanas representaron alrededor de US$33,000 millones en 2023. Gran parte de estas transacciones involucran una interacción con el cliente, entre las cuales destaca la posibilidad de redactar comentarios u opiniones sobre el producto o servicio adquirido. Estos datos representan información valiosa para las empresas, ya que con ella pueden tomar decisiones de venta y estrategias con los clientes. Conocer qué y cuántas opiniones de un producto o servicio fueron negativas o positivas resulta prácticamente imposible si se desea hacerlo de forma manual. La minería de opiniones (o análisis de sentimientos) tiene como tarea automatizar la asignación de etiquetas sobre una gran cantidad de textos. El presente trabajo de investigación tuvo como objetivo aplicar la minería de opiniones a un corpus de opiniones de transacciones de e-Commerce usando la técnica de n-gramas y algoritmos de aprendizaje automático. La metodología propone la compilación del corpus, obtención de características basadas en n-gramas, aplicación de algoritmos de aprendizaje automático, generación del modelo computacional y obtención de cifras de clasificación. Los resultados fueron muy alentadores, alcanzando un 88% de clasificación correcta con algoritmos de probabilidad, superando inclusive a otras propuestas del estado del arte, lo que sugiere la factibilidad de aplicación del modelo, con la ventaja adicional de su simplicidad e independencia del lenguaje.

Palabras clave: N-gramas, aprendizaje-automático, comercio-electrónico, análisis-de-sentimientos, minería-de-opiniones.

Opinion Mining in E-Commerce using n-Grams and Machine-Learning Algorithms

Abstract. There are currently 320 million e-commerce users in Latin America and the Caribbean, and it is expected to reach 400 million in 2028. The net e-

Commerce sales of the top 100 Mexican online stores alone represented around US$33 billion in 2023. Great part of these transactions involves an interaction with the customer, among which the possibility of writing comments or opinions about the product or service purchased stands out. This data represents valuable information for companies, since with it they can make sales decisions and strategies with customers. Knowing which and how many opinions of a product or service were negative or positive is practically impossible if we want to do it manually. Opinion mining (or sentiment analysis) has the task of automating the assignment of labels to many texts. The objective of this research work was to apply opinion mining to a corpus of opinions of e-Commerce transactions using the n-gram technique and machine learning algorithms. The methodology proposes the compilation of the corpus, obtaining characteristics based on n-grams, application of machine learning algorithms, generation of the computational model and obtaining a classification ranking. Very promising results were obtained, reaching 88% correct classification with probability algorithms, even surpassing other state-of-the-art proposals, which suggests the feasibility of applying the model, with the additional advantage of its simplicity and language independence.

Keywords: N-grams, machine-learning, e-commerce, sentiment-analysis, opinion mining.

1. Introducción

La práctica de adquirir bienes y servicios a través de canales electrónicos se está generalizando y ganando cada vez más aceptación entre los consumidores. El comercio electrónico ha hecho posible que los consumidores compren sin salir de la comodidad de sus hogares y con disponibilidad de 24 horas. Es así como esta industria ha experimentado un crecimiento exponencial en el número de sitios web de comercio electrónico [1]. Según la Asociación Mexicana de Ventas Online, las ventas de comercio electrónico en México superaron los US$33,000 millones en 2023.

De esta forma, el e-Commerce mexicano se posiciona como el segundo mercado online más importante en América Latina, superado únicamente por Brasil. A pesar de estas cifras, la tasa de abandono de las compras online representa un 70%. Para evitar que un futuro cliente desista de la compra se recomienda optimizar el proceso de checkout, ofrecer la posibilidad de finalizar la compra como invitado (muchas personas desisten cuando tienen que crear una cuenta) y tener referencias a través de opiniones de antiguos compradores [2]. El análisis de sentimientos (ADS), también conocido como análisis o minería de opiniones, le proporciona al cliente información acerca del producto o servicio antes de comprarlo

Por lo general, esta información es textual y recae en categorías de comentarios positivos o negativos. Esta información también es importante para las empresas y los especialistas en marketing para así hacer adecuaciones o mejoras a lo que están ofreciendo, satisfaciendo con ello las necesidades del cliente e incrementando sus ventas. Las empresas están integrando cada vez más el ADS en sus aplicaciones y sistemas. Esto se aplica a la atención al cliente, el seguimiento de la reputación en línea, la toma de decisiones basada en datos y mucho más. Los modelos de ADS son una parte importante del Procesamiento del Lenguaje Natural (PLN) y se utilizan para determinar

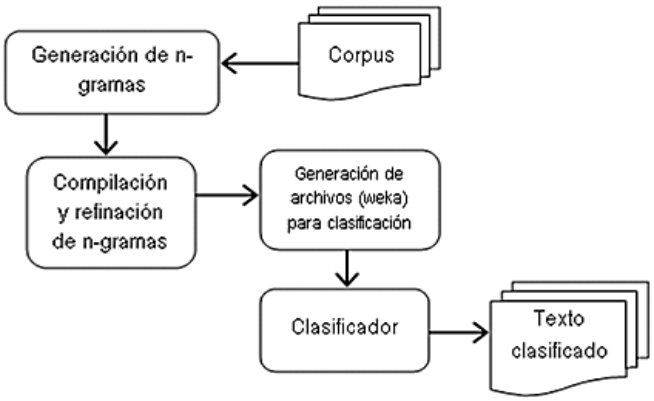


Fig. 1. Modelo de trabajo propuesto.

Te	baja	el	rizo	horrible	en	cuanto	lo	aplicas	,	por	el	precio	no	vale	la	pena	.
te	bajar	el	rizo	horrible	en	cuanto	lo	aplicar	,	por	el	precio	no	valer	el	pena	.
PP2CS00	*VMIP3S0*	*DA0MS0*	*NCMS000*	*AQ0CS00*	*SP*	*PR0MS00*	*PP3MSA0*	*VMIP2S0*	*Fc*	*SP*	*DA0MS0*	*NCMS000*	*RN*	*VMIP3S0*	*DA0FS0*	*NCFS000*	*Fp*

"Te baja el rizo horrible en cuanto lo aplicas, por el precio no vale la pena."

Fig. 2. Análisis PoS en FreeLing para la frase.

la actitud o emoción expresada en un texto, ya sea positiva, negativa o neutral. Algunos de los modelos trabajados en este campo son los basados en reglas, en bolsas de palabras, en palabras preentrenadas, aprendizaje profundo, transferencia de aprendizaje y de aprendizaje automático.

Es con este último modelo en el que está basado el presente trabajo de investigación, pero con el soporte de la técnica de n-gramas tradicionales, lo que conlleva la ventaja de ser un modelo más simple, eficiente e independiente del lenguaje.

2. Trabajo relacionado

En los últimos años se ha incrementado el interés en el estudio y aplicación del aprendizaje automático aplicado a la minería de opiniones. A continuación, se detallan algunos trabajos relevantes y actualizados sobre este campo. En [3] se propone un enfoque sintáctico a la minería de opiniones para el español, específicamente aplicando un parser de dependencias para obtener la estructura sintáctica, apoyado en diccionarios semánticos. Aunado a ello, estudiaron de forma especial las construcciones sintácticas de la negación, intensificación y cláusulas subordinadas adversativas.

La investigación de [4] propone un enfoque basado en aspectos y la combinación de dos modelos de aprendizaje profundo. Su trabajo fue orientado a opiniones en español del sector restaurantero, usando un corpus de aproximadamente 2000 opiniones. Un estudio más profundo sobre el análisis basado en aspectos se encuentra en [10]. Una revisión más detallada la llevó se llevó a cabo en [15], probando diversos clasificadores supervisados, entre ellos CNN multi-canal, LSTM (BERT) y RMDL (secuencial con cinco capas), alcanzado tasas de validación entre 78% y 80%.

El dataset utilizado fue de 34,660 comentarios en Amazon (en inglés) etiquetados como positivos o negativos. Un trabajo muy interesante lo encontramos en [13], donde hicieron uso de métodos de aprendizaje profundo (RNN, LSTM, BLSTM, CNN) y de

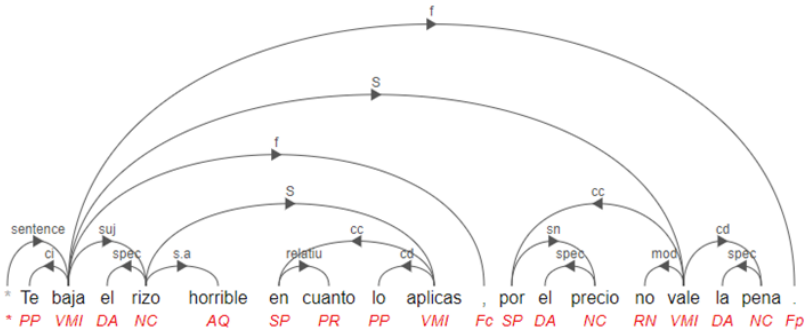


Fig. 3. Árbol de dependencias sintácticas para una oración del corpus.

aprendizaje automático (RF, MNB, SVM) aplicados a un dataset de 100 mil opiniones de comercio electrónico en tres lenguajes (inglés, turco y árabe) con tres categorías de sentimiento (positivo, negativo y neutral). Se usaron modelos pre-entrenados, entre ellos bert-base-multilingua-cased y xlm-roberta-base, obteniendo RNN el mejor desempeño para el inglés (89.5%), mientras que SVM para el árabe un 91.3%. Al tomar en cuenta los 3 lenguajes, SVM alcanzó los mayores valores de precisión. Por último, [9] representa un trabajo con una idea similar a la propuesta en esta investigación, con un enfoque sobre técnicas de bolsas y n-gramas de palabras como extractores de características y SVM, NB y KNN como clasificadores.

3. Metodología

En este apartado se explica el modelo propuesto de trabajo, iniciando con la compilación del corpus, el proceso de obtención de los n-gramas y finalizando con la tarea de clasificación. Los experimentos consistieron en la generación de n-gramas (mediante una herramienta libre de extracción); su compilación, que no es más que el almacenamiento dinámico de los n-gramas únicos (haciendo uso de estructuras matriciales y la implementación de un algoritmo para su manipulación); la refinación, que es la puesta a punto de los n-gramas únicos para que los caracteres no reconocidos por Weka sean detectados y cambiados; la generación de archivos de entrenamiento y clasificación; la aplicación de estadísticas de frecuencias y la aplicación de procesos de clasificación con los modelos de SVM (SMO), NB y J48 para distintas cantidades de n-gramas. Usamos un baseline de etiquetas de partes de oración (PoS). El modelo del trabajo se resume en la figura 1, donde el paso inicial (generación del corpus) se dividió en dos conjuntos: los comentarios originales y las dependencias generadas por el parser sintáctico de Freeling.

3.1. Compilación del corpus

La mayoría de los corpus disponibles en este campo de opiniones se encuentran en inglés, por lo que consideramos pertinente generar uno para el lenguaje español para futuros trabajos y para que esté disponible para la comunidad interesada en el tema. A través de herramientas de Web Scrapping aplicado a uno de los sitios más representativos de comercio electrónico (Amazon), se obtuvo un total de 10 mil

Tabla 1. Gramas de nivel 2,3,4 para las rutas de dependencias de la Fig. 2.

bigramas	trigramas	cuatrigramas
suj→spec	suj→S→cd	S→cc→sn→spec
suj→sa	suj→S→cc	suj→S→cc→relatiu
suj→S	S→cd→spec	
S→cd (2)	S→cc→sn	
S→cc		
S→mod		

Tabla 2. Gramas de nivel 2,3,4 para las partes de oración de la Fig. 2.

bigramas	trigramas	cuatrigramas
PP2CS00→VMIP3S0	PP2CS00→VMIP3S0→DA0MS0	PP2CS00→VMIP3S0→DA0MS0→NCMS000
VMIP3S0→DA0MS0	VMIP3S0→DA0MS0→NCMS000	VMIP3S0→DA0MS0→NCMS000→AQ0CS00
DA0MS0→NCMS000	DA0MS0→NCMS000→AQ0CS00	DA0MS0→NCMS000→AQ0CS00→SP
NCMS000→ AQ0CS00	NCMS000→AQ0CS00→SP	NCMS000→AQ0CS00→SP→PR0MS00
. . .	. . .	. . .

comentarios (5 mil positivos y 5 mil negativos) vertidos por clientes registrados. Algunos ejemplos de estos textos son los siguientes:

> *Había oferta y aproveché, el producto es el que siempre he utilizado y me es familiar y agradable el sabor*. (positivo).
>
> *Precio accesible, me llego en buenas condiciones y si lo volvería a comprar*. (positivo).
>
> *No llevaba ni un año y empezó a hacer un ruido extraño, ahora no sirve*. (negativo).
>
> *Te baja el rizo horrible en cuanto lo aplicas, por el precio no vale la pena*. (negativo).

Es importante mencionar que los comentarios no fueron editados, es decir, aquellos que presentan errores ortográficos u otro tipo de omisiones, se dejaron en su forma original.

3.2. Generación de n-gramas

Los n-gramas representan una técnica fundamental en el PLN. Se utilizan para analizar secuencias de palabras o caracteres en un texto al dividirlos en unidades contiguas de n elementos, que generalmente son palabras o caracteres. Estos elementos pueden ser considerados como "tokens" individuales y se utilizan para capturar información sobre la estructura y el contenido del texto. Tomando como ejemplo parte de la segunda opinión arriba mencionada ("*Precio accesible*,") podemos generar los bigramas de caracteres *Pr*, *re*, *ec*, *ci*, *io*, *o_*, *_a*, *ac*, *cc*, *ce*, *es*, *si*, *ib*, *bl*, *le* y *e*,.

De igual forma podemos generar los trigramas *Pre*, *rec*, *eci*, *cio*, *io_*, *o_a*, *_ac*, *cce*, *ces*, *esi*, *sib*, *ibl* y *ble*. Nuestro modelo hace un análisis estadístico de las apariciones de los gramas en cada una de las oraciones. Como parte innovadora, previamente se

Tabla 3. Baseline de la clasificación con gramas de PoS.

n-gramas	Clasificador	Tamaño del n-grama			
		2	**3**	**4**	**5**
500	NB	0.55	0.58	0.72	0.71
	SVM	0.70	0.72	**0.74**	0.71
	J48	0.70	0.71	0.68	0.68
1000	NB	0.69	0.71	0.70	0.69
	SVM	0.67	0.68	**0.71**	0.70
	J48	0.66	0.64	0.65	0.67
2000	NB	0.67	0.69	**0.70**	0.68
	SVM	0.64	0.66	0.65	0.68
	J48	0.61	0.63	0.64	0.63

Tabla 4. Resultados de la clasificación con gramas sintácticos.

n-gramas	Clasificador	Tamaño del n-grama			
		2	**3**	**4**	**5**
500	NBM	0.69	0.69	**0.88**	0.82
	SVM	0.72	0.72	0.75	0.74
	J48	0.71	0.71	0.71	0.70
1000	NBM	0.70	0.70	**0.84**	0.74
	SVM	0.70	0.70	0.73	0.69
	J48	0.63	0.63	0.65	0.64
2000	NBM	0.66	0.66	**0.79**	0.69
	SVM	0.64	0.64	0.68	0.68
	J48	0.60	0.60	0.61	0.60

obtendrá información sintáctica de las oraciones que representan los comentarios, esto para que los gramas incluyan información sobre la estructura de las oraciones.

La implementación de la generación de archivos Weka es un proceso semi-automático a través de un programa en Java, el cual está disponible para la comunidad académica e investigadora, junto con el corpus compilado y los archivos Weka generados.

3.3. Proceso de clasificación

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Proporciona herramientas para el pre-procesamiento, clasificación, regresión, clustering y visualización de datos. Es un software de código abierto basado en los términos de GNU-GPL. Cada registro de la sección de datos (@data) representa un comentario y cada valor numérico representa las ocurrencias de un grama en

particular en ese texto. El último valor, que en nuestro ejemplo no es numérico, representa la medida de la opinión (positiva o negativa).

El proceso de entrenamiento y clasificación también se llevó a cabo con el software Weka, que proporciona una diversidad de métodos. En particular, fueron usados los clasificadores NB, Optimización Mínima Secuencial (SMO - Support Vector Machines) y árboles de decisión (J48), ya que estos han mostrado buenos resultados en otros trabajos de investigación, como en [13].

En algunas primeras pruebas se llegó a utilizar un archivo de stopwords para el español, pero se notó que no aportó ninguna ventaja en los resultados, de hecho, fue todo lo contrario. Esto nos sugiere que todas las palabras aportan algo de valor en el proceso de clasificación.

3.4. Generación del baseline

El etiquetado de PoS es la tarea dentro del PLN que asigna una etiqueta de categoría gramatical a cada una de sus palabras. Ejemplos de estas categorías son los adjetivos, sustantivos y determinantes. A diferencia de los parsers sintácticos profundos (dependencias, constituyentes), la tarea de un parser sintáctico superficial, como lo es para las PosTags, requiere menos tiempo.

Este proceso se llevó a cabo con la librería de Freeling. Retomando la misma idea de lo mostrado en la tabla 1, podemos aplicarla para obtener gramas para categorías superficiales de PoS, con lo cual obtendríamos el resultado de la tabla 2. De esta manera se logró generar un baseline de trabajo donde notamos que el mayor porcentaje de clasificación correcta fue de 74%, para 500 cuatrigramas y el algoritmo SVM.

4. Resultados experimentales

Los experimentos fueron desarrollados sobre los datos de un corpus de minería de opiniones. Con el corpus de 1000 comentarios se hicieron diversas pruebas, siempre con un 60% de los datos para entrenamiento y el restante para clasificación. En la tabla 3 se muestran los resultados obtenidos. La tarea de clasificación consiste en seleccionar características para construir el modelo de espacio de vectores, algoritmos supervisados de entrenamiento y clasificación – decidir a qué clase pertenece el texto –en nuestro modelo de espacio de vectores.

En este trabajo presentamos resultados para tres clasificadores: SVM (SMO), NB y J48. En los resultados mostrados es interesante notar que con los clasificadores NB y SVM se obtiene una mayor exactitud para cuatrigramas, alcanzando un 88% con el primero de ellos; de hecho, los mejores resultados fueron para NB en prácticamente todas las pruebas. Es importante mencionar que se usó la medida TFD-ID.

Analizando más a profundidad la tabla 4 podemos observar que el clasificador con el porcentaje más alto en todas las pruebas de reseñas fue NaiveBayesMultinomial, logrando hasta un 88% con 500 gramas. Es una cifra similar a la mencionada en [9], pero las diferencias de los estudios es que ellos usaron gramas de palabras y el clasificador NaiveBayes y en nuestro trabajo se usaron gramas sintácticos de dependencias y el clasificador NaiveBayes Multinomial.

Comparado con el baseline, nuestra propuesta mejora en un 10% de clasificación correcta, aunque hay que destacar que SVM logra el porcentaje más alto en las pruebas del baseline. De igual forma, este trabajo corrobora lo mencionado en [11], en el sentido de que el clasificador Naïve Bayes funciona adecuadamente, incluso en situaciones de contexto pesado.

5. Conclusiones

Los resultados de los experimentos demuestran la factibilidad de usar modelos computacionales simples para la tarea de ADS, lo que redunda en un menor tiempo computacional de procesamiento. La contribución más importante es la simplicidad y la generalidad del modelo (puede ser aplicados a cualquier lenguaje).

En el comercio electrónico, la minería de opiniones es valiosa para comprender la satisfacción del cliente, identificar problemas con productos o servicios, y tomar decisiones informadas para mejorar la experiencia del cliente. Los ADS basados en n-gramas y algoritmos de aprendizaje automático pueden proporcionar información valiosa sobre tendencias y patrones en las opiniones de los clientes, lo que puede ayudar a las empresas a tomar medidas proactivas, como ajustar precios, mejorar la calidad del producto o el servicio, o desarrollar estrategias de marketing específicas.

Sin embargo, la idea aquí propuesta solo divide a los sentimientos en dos categorías (positivo y negativo), lo que quedaría en desventaja en áreas que necesitaran de más detalles, por lo que un trabajo futuro es trabajar con estas nuevas categorías de clasificación del texto (por ejemplo, “muy negativo”, “muy positivo”), ampliar el corpus de trabajo y analizar la generalización a otros dominios y lenguajes.

Referencias

1. Ayodeji, O.G., Kumar, V.: E-commerce Research Models: A Systematic Review and Identification of the Determinants to Success. International Journal of Business Information Systems (2020). DOI: 10.1504/ijbis.2020.10044532.
2. Statista: El comercio electrónico en México. Statista Research Department. http://es.statista.com/temas/6370/el-comercio-electronico-en-mexico/ (2024)
3. Singh, U., Saraswat, A., Azad, H.K., Abhishek, K., Shitharth, S.: Towards Improving e-Commerce Customer Review Analysis for Sentiment Detection. Scientific Reports, vol. 12, no. 1 (2022). DOI: 10.1038/s41598-022-26432-3.
4. Martínez-Seis, B.C., Pichardo-Lagunas, O., Miranda, S., Perez-Cazares, I.J., Rodriguez-Gonzalez, J.A.: Deep Learning Approach for Aspect-based Sentiment Analysis of Restaurants Reviews in Spanish. Computación y Sistemas, vol. 26, no. 2, pp. 899–908 (2022). DOI: 10.13053/cys-26-2-4258.
5. Aguilar, D., Sidorov, G., Batyrshin, I.: Caso de estudio de análisis de sentimientos en Twitter: Tratado de libre comercio de América del Norte. Research in Computing Science, vol. 147, no. 5, pp. 357–365 (2018). DOI: 10.13053/rcs-147-5-27.
6. Hernández, A., Ramírez, G., Villatoro, E.: Un método para el análisis de sentimientos bajo un enfoque supervisado usando recursos léxicos. Research in Computing Science, vol. 147, no. 6, pp. 221–233 (2018)

7. Henriquez, C., Guzmán, J., Salcedo, D.: Minería de opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. Procesamiento del Lenguaje Natural, vol. 56, pp. 25–32 (2016)
8. Moreno-Sandoval, L.G., Pomares-Quimbaya, A., Cruz-Gutiérrez, C.E., García-Pachón, J.F., Vanegas-Ramírez, D.F.: Comparación de métodos de análisis de sentimientos en comunidades de habla hispana. In: Encuentro Internacional de Educación en Ingeniería Asociación Colombiana de Facultades de Ingeniería, pp. 1–12 (2022). DOI: 10.26507/paper.2367.
9. Ali, M., Yasmine, F., Mushtaq, H., Sarwar, A., Idrees, A., Tabassum, S., Hayyat, B., Rehman, K.U.: Customer Opinion Mining by Comments Classification Using Machine Learning. International Journal of Advanced Computer Science and Applications, vol. 12, no. 5 (2021). DOI: 10.14569/ijacsa.2021.0120547.
10. Nazir, A., Rao, Y., Wu, L., Sun, L.: Issues and Challenges of Aspect-based Sentiment Analysis: A Comprehensive Survey. In: IEEE Transactions on Affective Computing, vol. 13, no. 2, pp. 845–863 (2022). DOI: 10.1109/taffc.2020.2970399.
11. López-Condori, J.J., Gonzales-Saji, F.O.: Análisis de sentimiento de comentarios en español en google play store usando BERT. Ingeniare, Revista Chilena de Ingeniería, vol. 29, no. 3, pp. 557–563 (2021). DOI: 10.4067/s0718-33052021000300557.
12. Sinnasamy, T., Sjaif, N.N.A.: A Survey on Sentiment Analysis Approaches in e-commerce. In: International Journal of Advanced Computer Science and Applications, vol. 12, no. 10, pp. 674–679 (2021). DOI: 10.14569/ijacsa.2021.0121074.
13. Savci, P., Das, B.: Prediction of the Customers' Interests Using Sentiment Analysis in e-commerce Data for Comparison of Arabic, English, and Turkish Languages. Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 3, pp. 227–237 (2023). DOI: 10.1016/j.jksuci.2023.02.017.
14. Agarwal, A., Biadsy, F., McKeown, K.R.: Contextual Phrase-level Polarity Analysis Using Lexical Affect Scoring and Syntactic n-grams. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics European Chapter of the Association for Computational Linguistics, pp. 24–32 (2009).
15. Singh, U., Saraswat, A., Azad, H.K., Abhishek, K., Shitharth, S.: Towards Improving e-commerce Customer Review Analysis for Sentiment Detection. Scientific Reports, vol. 12, no. 1 (2022). DOI: 10.1038/s41598-022-26432-3.
16. Vilares, D., Alonso, M.A., Gómez-Rodríguez, C.: A Syntactic Approach for Opinion Mining on Spanish Reviews. Natural Language Engineering, vol. 21, no. 1, pp. 139–163 (2013). DOI: 10.1017/s1351324913000181.
17. Yang, L., Li, Y., Wang, J., Sherratt, R.S.: Sentiment Analysis for e-commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. IEEE Access, vol. 8, pp. 23522–23530 (2020). DOI: 10.1109/access.2020.2969854.

ISSN 1870-4069

Clasificación de señales ECG mediante filtro UFIR y técnicas de aprendizaje automático

Victor Jiménez-Ramos, Roberto Baltazar-Castellanos,
César Hernández-Sanchez, Carlos Lastre-Domínguez

Tecnológico Nacional de México, IT Oaxaca,
Departamento de Ingeniería Electrónica,
México

victor.jimenez@itoaxaca.mx, {cesar.hernandez, carlos.lastre,
roberto.castellanos}@itoaxaca.edu.mx

Resumen. Las enfermedades cardiovasculares constituyen una de las principales causas de muerte a nivel mundial. Para diagnosticar patologías relacionadas con el corazón, el registro de electrocardiograma (ECG) es una herramienta diagnóstica fundamental. Este registro captura características morfológicas que pueden ser empleadas en sistemas automáticos de detección de patologías. Sin embargo, estas características pueden verse afectadas por ruido o artefactos. A lo largo de décadas, se han desarrollado técnicas para abordar este desafío, pero aún se requiere mejorar la precisión en la detección y clasificación automática de las señales ECG. En este contexto, el presente trabajo propone la clasificación de señales ECG que presentan arritmias, insuficiencia cardíaca congestiva y ritmo normal. Se destaca el análisis ANOVA y Kruskal-Wallis de las características espectrales, como la entropía espectral del filtro UFIR. Los hallazgos revelan una mejora significativa en la clasificación de los modelos de aprendizaje mediante el análisis de las curvas ROC.

Palabras clave: UFIR, inteligencia artificial, aprendizaje automático, curva ROC.

Classification of ECG Signals by UFIR Filter and Machine Learning Techniques

Abstract. Cardiovascular diseases are among the leading causes of death globally. Electrocardiogram (ECG) recording is a fundamental diagnostic tool to diagnose heart-related pathologies. This recording captures morphological characteristics that can be used in automatic pathology detection systems. However, these features can be affected by noise or artifacts. Over the years, techniques have been developed to address this challenge, but there is still a need to improve the accuracy of automatic detection and classification of ECG signals. In this context, this study proposes classifying ECG signals that present arrhythmias, congestive heart failure, and normal rhythm. It highlights the ANOVA and Kruskal-Wallis analysis of spectral features, such as the spectral

entropy of the UFIR filter. The findings of this study reveal a significant improvement in the classification of learning models by analyzing ROC curves.

Keywords: UFIR, artificial intelligence, machine learning, ROC curve.

1. Introducción

En la actualidad, las enfermedades cardiovasculares representan una de las principales causas de mortalidad a nivel mundial. Para abordar este desafío, la comunidad médica ha estado buscando constantemente estrategias tecnológicas que permitan la detección temprana de enfermedades cardíacas, salvando así vidas humanas. Una de las herramientas no invasivas más utilizadas con este propósito es el análisis del electrocardiograma (ECG), cuyos patrones son esenciales para predecir enfermedades cardíacas [1, 2]. El ECG registra una serie de características morfológicas, como las ondas P, el complejo QRS, la onda T y la onda U. Aunque esta última puede no ser siempre visible para el profesional médico, las tres primeras son suficientes para detectar patologías específicas, como arritmias o incluso infartos de miocardio [3]. Sin embargo, el ruido asociado a otros sistemas del cuerpo humano, como el respiratorio o el muscular, así como los artefactos generados por enfermedades como el Parkinson o movimientos involuntarios, pueden dificultar la identificación precisa de estos patrones [4].

En las últimas décadas, se han llevado a cabo numerosos estudios para desarrollar técnicas de filtrado y clasificación automática de las señales ECG. Entre ellas se encuentran los filtros convencionales Chebyshev y Butterworth [5]. Sin embargo, estas técnicas pueden presentar limitaciones, como un desfase significativo a medida que aumenta el orden del filtro. Otra técnica conocida es el filtro de Kalman, que se ha aplicado a señales ECG, aunque para un rendimiento óptimo del filtro se requiere un modelo del sistema conocido. Algunos trabajos han presentado un banco de filtros basados en Kalman para abordar este problema, pero deben considerar aspectos frecuenciales de las ondas P, complejo QRS y T para un funcionamiento efectivo del filtro [6].

Otras alternativas incluyen el uso de la transformada de Fourier para el análisis en el dominio de la frecuencia, aunque está limitada por problemas de resolución tiempo-frecuencia [7]. Una solución a este problema es la transformada wavelet, que ha sido ampliamente utilizada como filtro y para la extracción de características en señales ECG [8], aunque su selección adecuada puede requerir un proceso iterativo al seleccionar la función wavelet madre adecuada y tiempo computacional adicional.

También se han empleado técnicas de aprendizaje profundo (*deep learning*) para clasificar arritmias [9-14]. Incluso se han explorado técnicas de inteligencia artificial con estructuras más complejas, como las redes neuronales LSTM (long short-term memory) y las GANs (generative adversarial networks), para la clasificación, reconstrucción y detección automática de patrones cardíacos en señales ECG [15-21]. Sin embargo, estas técnicas pueden enfrentar limitaciones debido al alto consumo de recursos computacionales y la alta complejidad de comprensión de los modelos.

Recientemente, ha surgido una técnica prometedora conocida como Filtro UFIR, que se adapta a diferentes dinámicas y morfologías de señales ECG en entornos

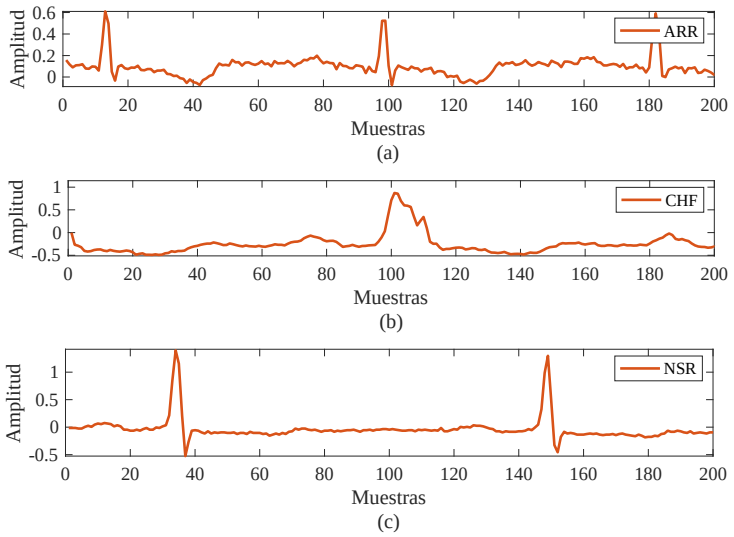


Fig. 1. Patologías de señales ECG: a) arritmia (ARR), insuficiencia cardíaca congestiva (CHF) y los ritmos sinusal normal (NSR).

estacionarios y no estacionarios, y además es fácil de implementar [22]. Este filtro produce estados asociados con un modelo polinomial en términos de una serie de Taylor, lo que permite estimaciones de derivadas de una señal, en este caso, registros ECG.

Esta característica podría ser aprovechada para extraer patrones que aumenten la capacidad discriminativa de un modelo de aprendizaje automático basado en *machine learning*. Este estudio propone un análisis de la clasificación de diversas patologías cardiovasculares, incluyendo el ritmo normal sinusal (NSR), la insuficiencia cardíaca congestiva (CHF) y las arritmias (ARR) (ver Figura 1). El objetivo principal es aprovechar las características extraídas de los estados del electrocardiograma (ECG) mediante el filtro UFIR para mejorar la clasificación de los modelos de aprendizaje automático.

El trabajo se organiza en las siguientes secciones: en la Sección I se detalla el modelo matemático del filtro UFIR; en la Sección II se aborda el proceso de extracción y selección de características; la Sección III describe el análisis del desempeño de los modelos utilizando la curva ROC; y finalmente, se exponen las conclusiones correspondientes.

2. Filtro UFIR

2.1. Modelo de espacio de estados de la señal de ECG

En este modelo, la señal ECG se representa dentro de un intervalo de tiempo $[m, n]$ de longitud N, donde $m = n - N + 1$. La representación de la señal se realiza

Algoritmo 1: Algoritmo del filtro iterativo UFIR

Datos: Y, N, A, C, W
Result:

1: Inicio
2: for $k = N - 1, N$ … do
3: $m = k - N + 1, s = k - N + K;$
4: $G_s = (W_{m,s}^T W_{m,s})(Y_{m,s});$
5: $\hat{x}_s = G_s(W_{m,s}^T)(Y_{m,s});$
6: for $i = s + 1: k$ do
7: $\tilde{x}_i^- = A_i \tilde{x}_{i-1};$
8: $G_i = [C_i^T C_i + (A_i G_{i-1} A_i^T)^{-1}]^{-1};$
9: $K_i = G_i C_i^T$
10: $\tilde{x}_i = \tilde{x}_i^- + K_i(y_i - H_i \tilde{x}_i^-)$
11: end for
12: $\hat{x}_n = \tilde{x}_n$
13: $\hat{x}_{n-q} = A^{-q} \hat{x}_n$
14: end for
15: Resultado: $\hat{x}_k$

utilizando un grado polinomial determinado en el espacio de estados, lo que permite una descripción precisa de la señal de ECG dentro del marco temporal definido. Es importante destacar que la señal de ECG se considera invariable en el tiempo y determinista. Se parte del supuesto de que la medición de la señal de ECG está afectada por un ruido con media cero, cuya desviación estándar es desconocida y que sigue una distribución gaussiana, aunque no necesariamente. Bajo estas condiciones, la representación de una señal de ECG se expresa de la siguiente manera:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{\mathbf{k+1}}, \tag{1}$$

$$y_k = \mathbf{C}\mathbf{x}_{\boldsymbol{k+1}} + v_n, \tag{2}$$

donde x_k es el vector de proceso de la señal de ECG, y_k es la observación de medición de la señal de ECG, v_k es el ruido de medición medio cero con distribución desconocida, $\mathbf{C}$ es la matriz de observación definida como $\mathbf{C} = [10 \ldots 0]$ y la matriz definida $\mathbf{A}$ es la matriz del sistema representada de la siguiente manera:

$$\mathbf{A} = \begin{bmatrix} 1 & \tau & \frac{(\tau)^2}{2} & \cdots & \frac{(\tau)^{K-1}}{(K-1)!} \\ 0 & 1 & \tau & \cdots & \frac{(\tau)^{K-2}}{(K-2)!} \\ 0 & 0 & 1 & \cdots & \frac{(\tau)^{K-3}}{(K-3)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}. \tag{3}$$

La matriz $\mathbf{A}$ es una representación matricial de la expación de series de potencia de Taylor o Maclaurin, donde K representa el número de estados [22]. Conocidas las

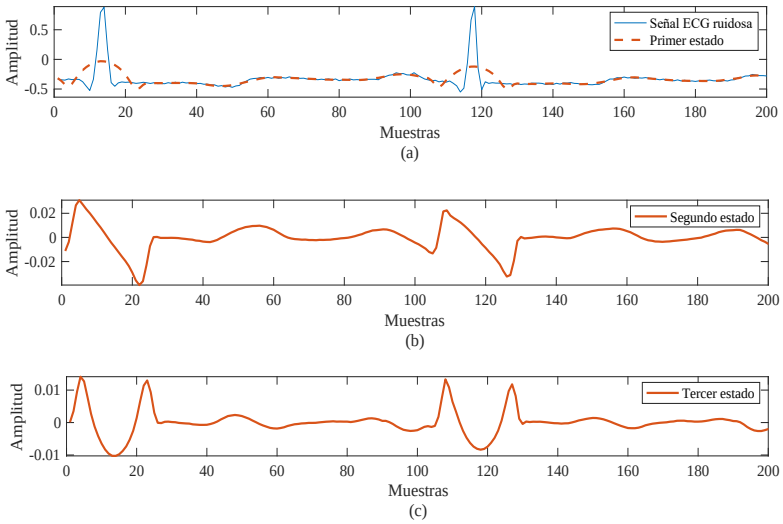


Fig. 2. Estimaciones del filtro UFIR. a) Estimación de la señal ECG, b) Primera derivada de la señal ECG, c) Segunda derivada de la señal ECG.

anteriores variables, sobre un horizonte $[m,n]$ de una serie de puntos del ECG, el fitro UFIR se puede representar como:

$$\mathrm{x}_k = (W^T{}_{m,n} W_{m,n})^{-1} Y_{m,n}\,, \tag{4}$$

donde $\boldsymbol{Y_{m,n}}$ es el vector de medición extendido de la señal ECG y la matrix $\boldsymbol{W_{mn}}$ se conoce como la matriz aumentada, ambas matrices se pueden representar de la siguiente forma, respectivamente:

$$\boldsymbol{Y_{m,n}} = [y_m^T\ y_{m+1}^T \dots y_n^T]^T, \tag{5}$$

$$\boldsymbol{W_{m,n}} = \begin{bmatrix} C(A^{n-m})^{-1} \\ \vdots \\ CA^{-1} \\ C \end{bmatrix}. \tag{6}$$

Es importante resaltar que la matrix $\boldsymbol{W_{mn}}$ contiene los coeficientes del filtro los cuales no son dependientes de la señal de entrada.

2.2. Algoritmo iterativo UFIR

A continuación, presentamos el algoritmo del filtro UFIR similar al filtro de Kalman, el cual se puede trabajar de manera iterativa, [22] (ver algoritmo 1).

$\boldsymbol{Y}$es la señal ECG con ruido (medición de la señal ECG), donde la variable N es el horizonte o ventana de puntos representada y q es una variable de paso que se puede determinar por las siguientes ecuaciones:

Tabla 1. Comparación del error cuadrático promedio y su desviación estándar de los filtros estudiados. UFIR-1: Filtro UFIR con q-lag 1, UFIR-2: Filtro UFIR con q-lag 2, db6: Filtro wavelet dautchevets, lowpass: filtro pasa bajas, medfil: filtro mediano.

Filtro	Promedio	Desviación estándar
UFIR-1	**13.5497**	**1.8783**
UFIR-2	**12.2957**	**1.7115**
db6	31.0054	3.1018
lowpass	16.063	2.0693
medfil	18.276	2.2115

$$q = \frac{N_{opt} - 1}{2}, \tag{7}$$

$$q = \frac{N - 1}{2} - \sqrt{\frac{N^2 - 1}{12}}. \tag{8}$$

Para el análisis del error cuadrático medio, se define el término UFIR-1 como el filtro UFIR con la variable de paso q determinada por la ecuación 7. Asimismo, el término UFIR-2 indica que se ha considerado la variable de paso q determinada por la ecuación 8. A continuación, se implementa el filtro UFIR-1. En este estudio, hemos determinado q-lag 1 y q-lag 2 para asociar las ecuaciones 7 y 8, respectivamente. En este caso, cada parámetro está definido como:

$$A = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

$$\text{C=[1 0 0]}, \boldsymbol{W} = \begin{bmatrix} CA^{-2} \\ CA^{-1} \\ C \end{bmatrix}, N = 21, \tag{10}$$

donde la variable Y es la señal ECG con ruido. Dado los parámetros, obtenemos las estimaciones presentadas en la figura 2.

Los estados representados en la figura 2 son el producto del filtro UFIR, inicialmente descrito por las ecuaciones 1 y 2, en función de las variables mencionadas. Cada estado está asociado a la estimación y derivadas de la señal ECG.

El primer estado corresponde a la estimación de la señal ECG, es decir, la señal ECG suavizada.

El segundo estado representa la estimación de la primera derivada de la señal ECG estimada.

Finalmente, el tercer estado indica la estimación de la segunda derivada de la señal ECG. [23, 24].

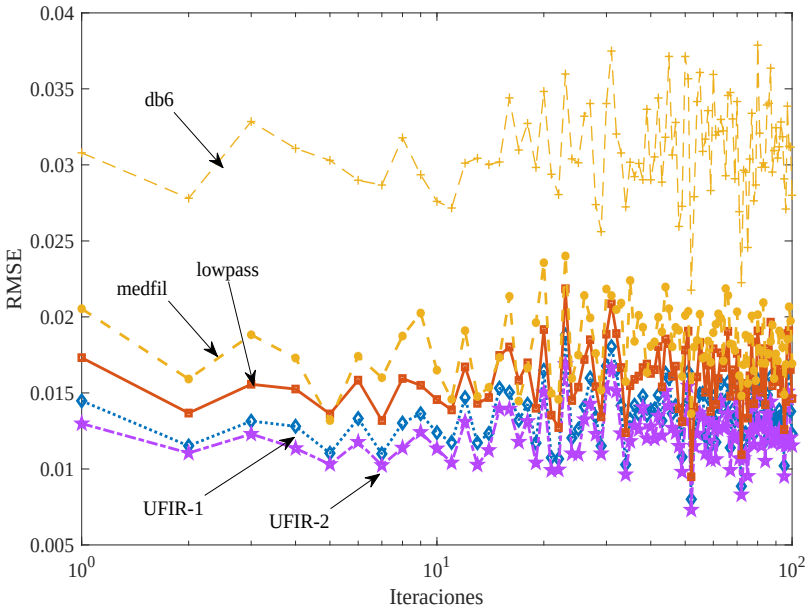


Fig. 3. Desempeño del error cuadrático medio (RMSE) de los filtros estudiados. UFIR-1: Filtro UFIR con q-lag 1, UFIR-2: Filtro UFIR con q-lag 2, db6: Filtro wavelet Daubechies con 6 coeficientes, lowpass: filtro pasa bajas, medfil: filtro mediana.

2.3. Análisis del error cuadrático medio (RMSE)

La estimación de la señal de ECG por los filtros estudiados se compara en términos de RMSE. El error cuadrático medio se determina mediante la ecuación 6:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{L}}\sum_{\mathrm{i}=0}^{\mathrm{L}}(\tilde{\mathrm{x}}_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}})^2}, \tag{11}$$

donde

$\tilde{x}_i$ es la muestra de señal de ECG estimada por los filtros.

y_i es la muestra de señal de referencia de ECG.

L es el tamaño de las muestras.

Se llevó a cabo un experimento utilizando una señal ECG sintética con 100 iteraciones y un nivel de ruido aleatorio de -6dB. En la Figura 3 se muestra el desempeño del error cuadrático medio (RMSE) de los diferentes filtros analizados. Como se puede observar en la Figura 3, se evidencia la variabilidad de cada filtro en función del número de iteraciones. Los filtros basados en UFIR muestran una menor variabilidad frente al ruido aleatorio en comparación con otros tipos de filtros.

Por lo tanto, se selecciona el filtro UFIR para el proceso de extracción y selección de características. En la tabla 1 podemos observar que el filtro UFIR produce menos

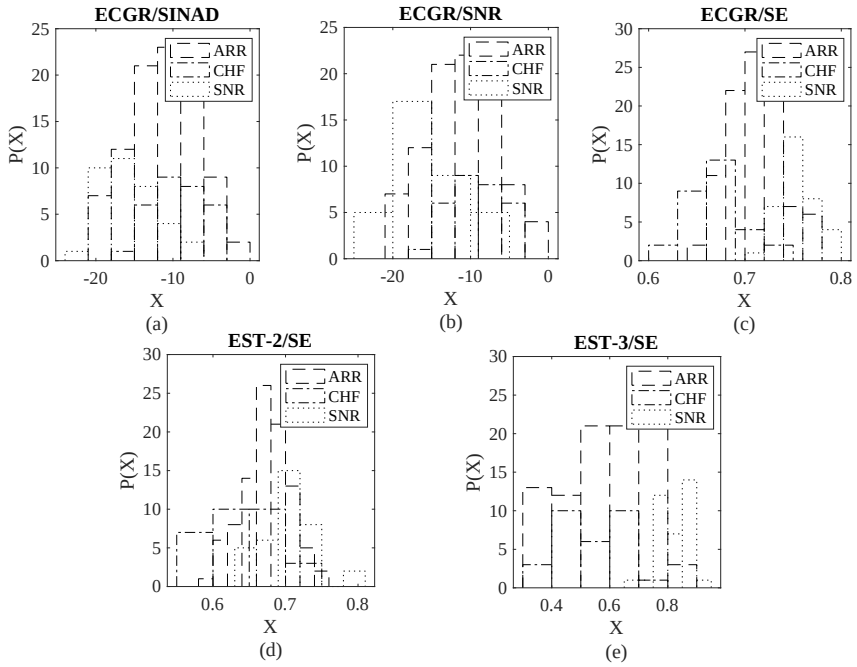


Fig. 4. Histogramas de características seleccionadas.

variabilidad en el sentido del error cuadrático medio comparado con filtros basados en transformada wavelet, filtro pasa bajas, filtro mediana.

3. Extracción y selección de características

Luego de completar el proceso de estimación utilizando el filtro UFIR, se llevaron a cabo extracciones de características tanto estadísticas, armónicas e impulsivas en el dominio temporal de las señales mencionadas, así como características espectrales en el dominio de la frecuencia. Estas características fueron luego sometidas a evaluación mediante análisis de varianza (ANOVA), calculando el valor F (Ver ecuación 10):

$$F = \frac{MSB}{MSW}, \tag{12}$$

dónde:

MSB es el cuadrado medio entre grupos.

MSW es el cuadrado medio dentro del grupo.

También se realizó un análisis de Kruskal-Wallis por el medio del test estadístico (H or χ^2, chi-cuadrado) representado por la ecuación 11:

Tabla 2. Ranking de características por análisis ANOVA y Kruskal-Wallis. EST-3/ SE: entropia espectral del estado 3, ECGR/SE: entropia espectral de la señal ruidosa, EST-2/ SE: entropia espectral del segundo estado, ECGR/SNR: relación señal a ruido de la señal ECG ruidosa, ECGR/SINAD: relación de distorción señal a ruido de la señal ECG.

Características	ANOVA	Kruskal-Wallis
EST-3/ SE	**81.2397**	**79.9787**
ECGR/SE	53.8797	42.0385
EST-2/ SE	25.0059	12.3776
ECGR/SNR	24.1638	19.5574
ECGR/SINAD	24.0130	14.6998

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{SS_i}{n_i} - 3(N+1), \tag{13}$$

dónde:

k es el número de grupos.

N es el número total de observaciones.

n_i es el número de observaciones en el i -ésimo grupo.

SS_i es la suma de los cuadrados de los rangos dentro del i -ésimo grupo.

Como se detalla en la tabla 1, con el fin de detectar posibles diferencias significativas entre las clases basadas en estas características. Como resultado de este procedimiento, se identificaron las cinco características más relevantes, algunas de las cuales se derivan de la señal de electrocardiograma ruidosa (ECGR), mientras que otras provienen de los estados 2 y 3 del filtro UFIR.

Como pueden ver en la figura 3, se representa la distribución P(X) de las características analizadas. Entre las características más importantes, sobresalen la SINAD (*signal to noise and distorsion ratio*), relación señal a ruido (SNR) y entropía espectral (SE). Entre todas ellas resalta la entropía espectral del estado 3 resultante del algoritmo UFIR. La base de datos de la señales ECG están descritas en [25, 26], del cual en este estudio se analizaron un total de 162 señales, 96 señales con ARR, 30 señales con CHD y 36 señales con SNR.

4. Análisis de curva ROC

Con base en las características seleccionadas, se llevó a cabo el proceso de entrenamiento y prueba utilizando una división hold-out 80/20 y validación cruzada 10 con diversos modelos de aprendizaje. Como se observa en las Figuras 4 y 5, los clasificadores que obtuvieron los mejores resultados durante el proceso de prueba están fundamentados en un ensamble de árboles de decisión, discriminante lineal y máquinas de soporte vectorial con kernel cuadrático y cúbico. Al evaluar cada modelo mediante la curva ROC, se encontró que el área bajo la curva (AUC) mostró resultados cercanos a 0.9 para las clases de señales ECG, especialmente en los modelos basados en máquinas de soporte vectorial y discriminante lineal.

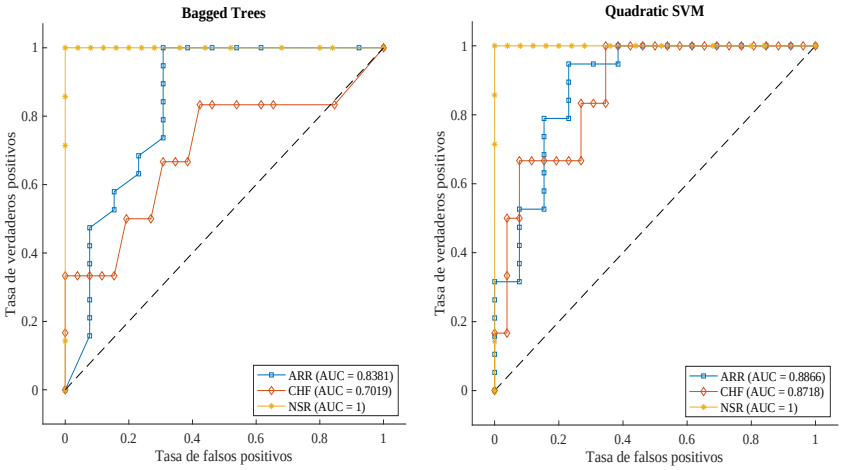


Fig. 5. Curvas ROC:. a) Bagged Trees: Ensanble de árboles, b) Quadratic SVM: Máquinas de soporte vectorial con kernel cuadrático.

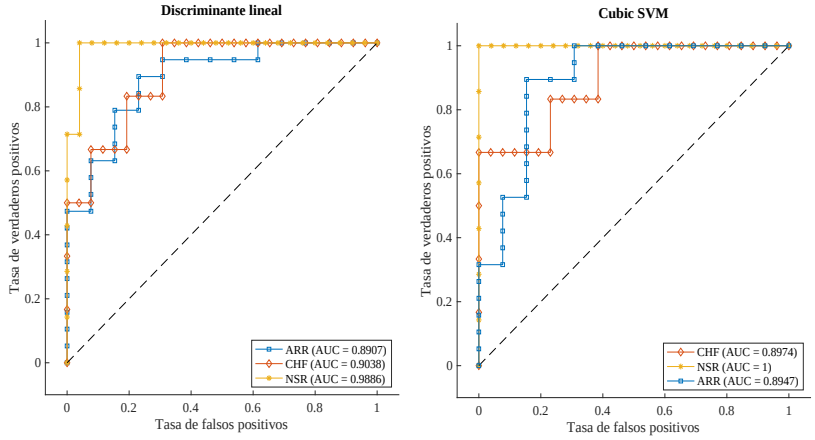


Fig. 6. Curvas ROC: a) Discriminante lineal, b) Cubic SVM: Máquinas de soporte vectorial con kernel cúbico.

5. Conclusiones

El filtro UFIR ha demostrado ser una opción estable frente al ruido aleatorio, evaluado en términos del error cuadrático medio (RMSE), en comparación con otros filtros como los basados en wavelets, los pasa bajos y los de mediana.

Esta técnica UFIR emerge como una herramienta prometedora que, combinada con estrategias de aprendizaje automático, puede producir resultados significativos. El análisis de las curvas ROC revela el desempeño prometedor de los algoritmos de

aprendizaje utilizados, destacando especialmente los modelos basados en máquinas de soporte vectorial y discriminante lineal.

Es importante destacar que las derivadas de la señal ECG ofrecen características que tienen el potencial de mejorar el diagnóstico de patologías en estas señales. Como perspectiva futura, se contempla la ampliación del conjunto de datos disponible y la exploración de estados superiores a los estudiados en el filtro UFIR.

Referencias

1. Goldberger, A.L., Goldberger, Z.D., Shvilkin, A.: Goldberger's Clinical Electrocardiography. Perfusion (2018)
2. Armstrong, M.L.: Los electrocardiogramas: Método sistemático para su lectura. El Ateneo, (1974)
3. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C., Stanley, H.E.: Physiobank, Physiotoolkit, and Physionet: Components of a New Research Resource for Complex Physiologic Signals. Circulation, vol. 101, no. 23 (2000). DOI: 10.1161/01.cir.101.23.e215.
4. Akbilgic, O., Kamaleswaran, R., Mohammed, A., Ross, G.W., Masaki, K., Petrovitch, H., Tanner, C.M., Davis, R.L., Goldman, S.M.: Electrocardiographic Changes Predate Parkinson's Disease Onset. Scientific Reports, vol. 10, no. 1 (2020). DOI: 10.1038/s41598-020-68241-6.
5. Basu, S., Mamud, S.: Comparative Study on the Effect of Order and Cut off Frequency of Butterworth Low Pass Filter for Removal of Noise in ECG Sgnal. In: IEEE 1st International Conference for Convergence in Engineering, pp. 156–160 (2020). DOI: 10.1109/icce50343. 2020.9290646.
6. Hesar, H.D., Mohebbi, M.: An Adaptive Kalman Filter Bank for ECG Denoising. In: IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 1, pp. 13–21 (2021). DOI: 10.1109/jbhi.2020.2982935.
7. Tripathy, R.K., Dash, D.K., Ghosh, S.K., Pachori, R.B.: Detection of Different Stages of Anxiety from Single-channel Wearable ECG Sensor Signal Using Fourier–bessel Domain Adaptive Wavelet Transform. IEEE Sensors Letters, vol. 7, no. 5, pp. 1–4 (2023). DOI: 10.1109/lsens.2023.3274668.
8. Amri, M.F., Rizqyawan, M.I., Turnip, A.: ECG Signal Processing Using Offline-wavelet Transform Method Based on ECG-IoT Device. In: 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp. 1–6 (2016). DOI: 10.1109/icitacee.2016.7892404.
9. Hou, Y., Liu, R., Shu, M., Xie, X., and Chen, C.: Deep Neural Network Denoising Model Based on Sparse Representation Algorithm for ECG Signal. In: IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–11 (2023). DOI: 10.1109/tim.2023. 3251408.
10. Hou, Y., Liu, R., Shu, M., Xie, X., Chen, C.: Deep Neural Network Denoising Model Based on Sparse Representation Algorithm for ECG Signal. IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–11 (2023). DOI: 10.1109/tim.2023. 3251408.
11. Islam, M.S., Islam, M.N., Hashim, N., Rashid, M., Bari, B.S., Farid, F.A.: New Hybrid Deep Learning Approach Using BiGRU-BiLSTM and Multilayered Dilated CNN to Detect Arrhythmia. IEEE Access, vol. 10, pp. 58081–58096 (2022). DOI: 10.1109/access.2022. 3178710.
12. Hou, Y., Liu, R., Shu, M., Xie, X., Chen, C.: Deep Neural Network Denoising Model Based on Sparse Representation Algorithm for ECG signal. In: IEEE Transactions on

Instrumentation and Measurement, vol. 72, pp. 1–11 (2023). DOI: 10.1109/tim.2023.3251408.
13. Xiao, Q., Lee, K., Mokhtar, S.A., Ismail, I., Pauzi, A.L.b.M., Zhang, Q., Lim, P.Y.: Deep Learning-based ECG Arrhythmia Classification: A Systematic Review. Applied Sciences, vol. 13, no. 8, pp. 4964 (2023). DOI: 10.3390/app13084964.
14. Kiranyaz, S., Devecioglu, O.C., Ince, T., Malik, J., Chowdhury, M., Hamid, T., Mazhar, R., Khandakar, A., Tahir, A., Rahman, T., Gabbouj, M.: Blind ECG Restoration by Operational Cycle-GANs. In: IEEE Transactions on Biomedical Engineering, vol. 69, no. 12, pp. 3572–3581 (2022). DOI: 10.1109/tbme.2022.3172125.
15. Shaker, A.M., Tantawi, M., Shedeed, H.A., Tolba, M.F.: Generalization of Convolutional Neural Networks for ECG Classification Using Generative Adversarial Networks. IEEE Access, vol. 8, pp. 35592–35605 (2020). DOI: 10.1109/access.2020.2974712.
16. Nankani, D., Baruah, R.D.: Investigating Deep Convolution Conditional Gans for Electrocardiogram Generation. In: International Joint Conference on Neural Networks, pp. 1–8 (2020). DOI: 10.1109/ijcnn48605.2020.9207613.
17. Nankani, D., Baruah, R.D.: Investigating Deep Convolution Conditional GANs for Electrocardiogram Generation. In: International Joint Conference on Neural Networks, pp. 1–8 (2020). DOI: 10.1109/ijcnn48605.2020.9207613.
18. Berger, L., Haberbusch, M., Moscato, F.: Generative Adversarial Networks in Electrocardiogram Synthesis: Recent Developments and Challenges. Artificial Intelligence in Medicine, vol. 143, pp. 102632 (2023). DOI: 10.1016/j.artmed.2023.102632.
19. Jyotishi, D., Dandapat, S.: An LSTM-based Model for Person Identification Using ECG signal. IEEE Sensors Letters, vol. 4, no. 8, pp. 1–4 (2020). DOI: 10.1109/lsens.2020.3012653.
20. Yamamoto, K., Hiromatsu, R., Ohtsuki, T.: ECG Signal Reconstruction Via Doppler Sensor by Hybrid Deep Learning Model with CNN and LSTM. IEEE Access, vol. 8, pp. 130551–130560 (2020). DOI: 10.1109/access.2020.3009266.
21. Shmaliy, Y.S., Zhao, S.: Optimal and Robust State Estimation: Finite Impulse Response and Kalman Approaches. John Wiley and Sons (2022). DOI:10.1002/9781119863106.
22. Shmaliy, Y.: Unbiased FIR Filtering of Discrete-time Polynomial State-space Models. In: IEEE Transactions on Signal Processing, vol. 57, no. 4, pp. 1241–1249 (2009). DOI: 10.1109/tsp.2008.2010640.
23. Shmaliy, Y.S., Zhao, S., Ahn, C.K.: Unbiased Finite Impulse Response Filtering: An Iterative Alternative to Kalman Filtering Ignoring Noise and Initial Conditions. IEEE Control Systems Magazine, vol. 37, no. 5, pp. 70–89 (2017). DOI: 10.1109/MCS.2017.2718830.
24. Moody, G., Mark, R.: The Impact of the MIT-BIH Arrhythmia Database. IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 3, pp. 45–50 (2001). DOI: 10.1109/51.932724.
25. Clifford, G., Liu, C., Moody, B., Lehman, L., Silva, I., Li, Q., Johnson, A., Mark, R.: AF Classification from a Short Single Lead ECG Recording: The Physionet Computing in Cardiology Challenge 2017. Computing in Cardiology, vol. 44, pp. 1 (2017). DOI: 10.22489/cinc.2017.065-469.

ISSN 1870-4069

Generación de sustitutos alimentarios mediante inteligencia artificial: Un enfoque combinado de modelado supervisado y algoritmos genéticos

Daniel Hernández-Mota, Cesar Lozano-Díaz, Raquel Zúñiga-Rojas

Instituto Tecnológico y de Estudios Superiores de Occidente,
México

{daniel.hernandezm, cesarlozano, rzuniga}@iteso.mx

Resumen. Esta investigación explora el uso de técnicas de IA, específicamente Aprendizaje Automático (ML) y Algoritmos Genéticos, para impulsar la innovación práctica en la ingeniería de alimentos. El enfoque innovador integra diversas fuentes de datos sobre características relevantes de los alimentos, como moléculas y perfiles de sabor con sus respectivos grupos funcionales de sabor, así como valores nutricionales. Se desarrolló un modelo de clasificación binaria de Bosque Aleatorio que compara productos alimenticios por pares, aprendiendo a identificar similitudes con un rendimiento prometedor (AUC PR de 0.898). Este modelo se integró en un algoritmo genético iterativo que propuso listas de ingredientes optimizadas para replicar productos objetivo como queso, leche y mantequilla. Los candidatos generados lograron puntajes de similitud entre 0.5 y 0.7, indicando una probabilidad del 80% de conservar propiedades nutricionales y sensoriales comparables a los productos originales, pero con una composición de ingredientes completamente diferente. Esta metodología demuestra el potencial de la IA para innovar en el diseño y personalización de productos alimenticios, contribuyendo a la diversificación, sostenibilidad y accesibilidad en la industria alimentaria.

Palabras clave: Inteligencia artificial, aprendizaje automático, algoritmos genéticos, ingeniería de alimentos.

Generation of Food Substitutes Using Artificial Intelligence a Combined Approach of Supervised Modeling and Genetic Algorithms

Abstract. This research explores the use of Artificial Intelligence (IA) techniques, specifically Machine Learning (ML) and Genetic Algorithms, to drive practical innovation in food engineering. The innovative approach integrates diverse data sources on relevant food characteristics, such as flavor molecules and profiles with their respective functional groups, as well as nutritional values. A Random Forest binary classification model was developed that compares food products in pairs, learning to identify similarities with promising performance (Area Under the Curve of the Precision-Recall relation (AUC PR) of 0.898). This model was integrated into an iterative genetic algorithm that proposed optimized lists of ingredients to replicate target products

such as cheese, milk, and butter. The generated candidates achieved similarity scores between 0.5 and 0.7, indicating an 80% probability of preserving nutritional and sensory properties comparable to the original products, but with a completely different ingredient composition. This methodology demonstrates the potential of AI to innovate in food product design and personalization, contributing to diversification, sustainability, and accessibility in the industry.

Keywords: Artificial intelligence, machine learning, genetic algorithms, food engineering.

1. Introducción

La IA se está integrando cada vez más en diversas esferas de la vida cotidiana, impactando significativamente en múltiples sectores gracias a su capacidad para procesar y aprender de grandes volúmenes de datos. En particular, el campo de la ingeniería de alimentos ha comenzado a explorar las capacidades del ML para innovar en el diseño y la optimización de productos alimenticios. Tradicionalmente, este diseño se ha basado en la experiencia de los expertos; sin embargo, la incorporación de técnicas avanzadas de IA promete superar las limitaciones de los enfoques convencionales mediante el uso de algoritmos que descubren patrones complejos y ofrecen soluciones menos sesgadas y más eficientes [1-3].

El uso de ML en la ingeniería de alimentos no es completamente nuevo. Estudios previos han aplicado estos métodos para el desarrollo de productos como alimentos estructurados [4], o incluso productos individuales como la mayonesa [5], y galletas [6], demostrando de esta manera la versatilidad y eficacia de la IA para mejorar las propiedades nutricionales y sensoriales de los alimentos, así como para reducir el desperdicio alimentario.

Adicionalmente, los algoritmos genéticos han ganado popularidad dentro del ámbito de la creatividad computacional, usándose para generar innovaciones como nuevas recetas de comida y sopas [7, 8]. Estos estudios destacan la capacidad de los modelos de ML para ajustarse y predecir características deseables en alimentos basados en datos tanto objetivos como subjetivos proporcionados durante el entrenamiento [9,10]. En el aprendizaje automático, se distinguen dos enfoques principales: supervisado y no-supervisado. Los modelos supervisados, que son los más utilizados, buscan predecir un valor deseado ajustando un modelo sobre un conjunto de datos con variables de respuesta conocidas [9].

Estos modelos son a menudo descritos como "cajas negras" debido a su capacidad para adaptarse a diversos contextos y tipos de datos, desde tabulares hasta sensoriales, dependiendo de la cantidad de datos y la experiencia del usuario con los modelos [10]. Entre los algoritmos más comunes se encuentran la regresión lineal, la regresión logística, y técnicas más complejas como las redes neuronales y algoritmos de ensamble basados en árboles, como bosques aleatorios y potenciación del gradiente. En contra parte, los algoritmos no-supervisados se pueden emplear para realizar agrupaciones de los datos o reducciones en la dimensión de estos.

Estos algoritmos no-supervisados se pueden emplear en conjunto con los supervisados para mejorar el poder predictivo de los algoritmos al aumentar la calidad de información que contienen las variables. Este proyecto investiga si es posible

mejorar el diseño y la personalización de productos alimenticios utilizando un enfoque combinado de aprendizaje supervisado y algoritmos genéticos para simular un conjunto de recetas con distintos ingredientes y de esta manera llegar a generar un producto similar, tanto nutricional como de sabor, pero completamente diferente en composición. Hipotetizamos que la implementación de un sistema de IA que integra técnicas avanzadas de aprendizaje supervisado con algoritmos genéticos puede ofrecer propuestas innovadoras que no solo compitan, sino que potencialmente sustituyan a productos alimenticios convencionales, mejorando así la accesibilidad y calidad nutricional de los alimentos. Este enfoque pretende no solo satisfacer las expectativas nutricionales y sensoriales de los consumidores, sino también contribuir significativamente a la reducción del desperdicio de alimentos, alineándose con los objetivos de sostenibilidad global.

2. Metodología

Las principales herramientas y software utilizados en este estudio incluyen:

- **Base de Datos de FlavorDB y USDA Branded Foods, Edamam API:** Utilizados para obtener datos de ingredientes e información nutricional [11-13].
- **Python3:** Lenguaje de programación principal [14].
- **Librerias pandas (2.0.3), numpy (1.24.4), nltk (3.8.1), matplotlib (1.3.1), seaborn (0.12.2), Scikit-Learn (3.7.3), Shap (0.43.0):** Librerías desarrolladas en python, utilizadas para realizar el análisis de datos, la manipulación de la información numérica, así como la limpieza de la información y la visualización de los resultados [15-19]. También utilizada para implementar modelos de aprendizaje automático, tanto supervisados (Bosque Aleatorio) como no supervisados (análisis de componentes principales) y finalmente la explicabilidad del modelo [20,21].
- **Algoritmos genéticos:** Se desarrolló un algoritmo genético para optimizar combinaciones de ingredientes en la generación de recetas.

2.1. Procedimiento

Adquisición de datos. Se recopilaron datos de FlavorDB, de la base de datos USDA Branded Foods y de la API Edamam. FlavorDB proporcionó información respecto a nombres de moléculas de sabor, perfiles de sabor detallados y grupos funcionales de las moléculas asociadas con cada ingrediente alimenticio. La USDA proporcionó información respecto a los ingredientes de cada producto y también las categorías a las que pertenecen dichos productos. Y la API de Edamam proporcionó información nutricional, estandarizada a medidas de 100g para cada entidad.

Selección y Limpieza de Datos. Se excluyeron manualmente las entidades de FlavorDB consideradas demasiado generales (por ejemplo, “jugo de frutas”, “otros quesos”, etc.) o que fueran específica a productos ya generados (por ejemplo, “tacos”, “hotcakes”, etc.) para centrarse solamente en los ingredientes fundamentales, obteniendo 677 distintas entidades de los 936 originales. Respecto a la USDA, la limpieza de texto implicó transformar todas las descripciones de los ingredientes,

utilizando técnicas convencionales de procesamiento de lenguaje natural: cambiar el texto a minúsculas, eliminar caracteres especiales y aplicar técnicas de derivación (stemming) a cada una de las palabras, esto mismo se realizó para cada ingrediente de FlavorDB.

Integración de Datos. Los datos limpios de la USDA se cruzaron con los de FlavorDB para obtener una selección tanto de registros de la USDA que contuvieran información que estuviera de igual manera en FlavorDB como para ayudar a generar una lista simplificada de ingredientes.

Ingeniería de Características. De manera individual, se sometieron tanto los perfiles de sabor, grupos funcionales y datos de moléculas a un Análisis de Componentes Principales (PCA) para obtener una reducción de dimensionalidad, creando un conjunto condensado de características para cada entidad. El perfil de sabor se redujo de 592 elementos a 50 componentes, los grupos funcionales de 84 elementos se redujeron a 20 componentes, y las distintas moléculas se redujeron de 1702 a 100 componentes. Las propiedades nutricionales se mantuvieron como los 34 elementos que originalmente fueron proporcionados por la API de Edamam. Esto generó que cada entidad (e_i) tuviera en total 204 descriptores, como se aprecia en (1):

$$e_i = [c_1,\ c_2,\ c_3,\ c_4, \dots\ ,\ c_{204}], \tag{1}$$

donde $c_{[1-100]} \in$ Moléculas de sabor, $c_{[101-150]} \in$ Perfil de sabor, $c_{[151-170]} \in$ Grupos funcionales de sabor, $c_{[171-204]} \in$ Propiedades nutricionales.

Representación de producto. Un producto (p_i) se define de manera simplificada como la lista de ingredientes, es decir un vector de entidades, esto se denota en (2):

$$pi = [e_1,\ e_2,\ e_3,\ e_4, \dots,\ e_n]. \tag{2}$$

Entonces también se puede interpretar a un producto como una matriz donde cada elemento fila representa la información de las características de cada entidad, en otras palabras, un producto consiste en elementos que cuentan con características multidimensionales, es decir que cada característica tiene su propia dimensionalidad interna, esto se aborda de manera similar a [22]. La representación se encuentra en (3):

$$p_i = [e_1,\ e_2, \dots,\ e_n] = [[c_1,\ c_2, \dots,\ c_{204}]_1,\ [c_1,\ c_2, \dots,\ c_{204}]_2, \dots, [c_1,\ c_2, \dots,\ c_{204}]_n]. \tag{3}$$

Por simplicidad se redefine un producto obteniendo el valor promedio de cada característica (o columna), esto se encuentra en (4):

$$p'_i \coloneqq [c'_1, c'_2, c'_3, c'_4, \dots, c'_{204}] \text{ donde } c'_k = \frac{1}{n}\sum_{j=1}^{n} c_{kj}. \tag{4}$$

Variable de respuesta. Se desarrolló un conjunto de etiquetas (y_i) con granularidad producto. Para esto se consideró de manera manual un refinamiento de las categorías proporcionadas de la USDA. Estas etiquetas en total fueron las siguientes: bread, butter, cheese, egg, fruit, honey, meat, milk, oil, seafood, vegetable, y yogurt. Para ilustrar el refinamiento, la etiqueta y_{milk} estaba compuesta por las siguientes categorías de la USDA: “Milk”, “Plant Based Milk” y “Milk/Milk Substitutes”. Sin embargo, una vez teniendo estas etiquetas, en lugar de abordar el problema como un problema

supervisado de multiclasificación, se decidió transformar la notación a un problema de clasificación binaria.

Al hacer esta transformación, ya no se requieren las etiquetas (y_i) propuestas, sino que se realiza el análisis utilizando pares de datos ($p'_i,\ p'_j$) con insumo del modelo y la similitud por pares S_{ij} para la variable de respuesta, donde se define que ambos dos elementos pertenecen a la misma etiqueta, entonces son similares ($S_{ij} = 1$). Y si pertenecen de manera individual a distintas etiquetas entonces no son similares ($S_{ij} = 0$) [23]. Esto se aprecia en (5 y 6):

$$\forall p'_i \in y_a \wedge \forall p'_j \in y_b \text{ donde } a = b, \text{ entonces } S_{ij} = 1, \tag{5}$$

$$\forall p'_i \in y_a \wedge \forall p'_j \in y_b \text{ donde } a \neq b, \text{entonces } S_{ij} = 0. \tag{6}$$

2.2. Análisis de datos

Entrenamiento del Modelo. Se entrenó un modelo de Bosque Aleatorio [24], de la librería de Scikit Learn manteniendo los parámetros iniciales, utilizando un sistema de clasificación binaria basado en la similitud de productos alimenticios dentro de categorías específicas. El conjunto de datos de 5,851 productos se expandió a más de 34 millones al realizar las comparaciones para determinar su similitud. emparejando cada producto con otro. De este conjunto, debido a capacidades de memoria y procesamiento, se obtuvo una muestra aleatoria del 10% de los registros reduciendo el volumen de información a solo 3.4 millones de registros.

Validación del Modelo. El conjunto de datos para entrenar y validar el modelo se dividió en dos subconjuntos: el conjunto de entrenamiento (CE) (considerando solamente el 70% de los productos únicos disponibles) y el conjunto de prueba (CP) (contemplando el 30% de los productos únicos disponibles sobrantes), asegurando que no hubiera superposición de productos entre los conjuntos para prevenir sesgo en la evaluación del modelo. En otras palabras, no existía ningún producto en el entrenamiento que estuviera en el CP y viceversa.

Una vez entrenado el modelo, se procedió a evaluar su desempeño con estos productos nunca vistos en el entrenamiento. Además de esto, se pudo verificar el desempeño del modelo con otro subconjunto de datos, utilizando algunos de los valores sobrantes no utilizados para el proceso es decir del conjunto original de los 34 millones, se obtuvieron aquellos donde no había productos que hayan estado en el CE para mejorar la validación del modelo.

Implementación del Algoritmo Genético: Una vez validado el modelo y cuantificado su desempeño, se utilizó como función de optimización para un algoritmo genético destinado a generar productos de alimentos [25], buscando la similitud de un producto específico a la vez. El algoritmo iteró a través de varias propuestas de listas de ingredientes para proponer una combinación óptima basada en las características, tales como el contenido nutricional y el sabor. Primeramente, para cada etiqueta, se define un producto a desarrollar a través de la lista de sus entidades correspondientes.

Después se inicializa una población contemplando distintas listas de ingredientes (llamados también candidatos) generadas de manera aleatoria y se hace la evaluación

Tabla 1. Resultados del modelo en distintos conjuntos de evaluación: CE, CP y CR. Los resultados muestran que el desempeño del modelo en diversas métricas es notablemente alto. En particular, un valor superior a 0.8 en el AUC PR es un indicador de que el modelo está identificando las clases de manera efectiva y precisa.

Métrica	CE	CP	CR
Cantidad de registros	1674846	307924	308354
AUC ROC	0.999828	0.975239	0.975707
AUC PR	0.998695	0.898611	0.898407
PP	0.998694	0.895873	0.895532

del modelo para seleccionar los mejores candidatos los cuales pasarían sus genes a la siguiente iteración. Esta selección aleatoria puede estar restringida tal que no se utilicen ciertas entidades no deseadas.

Una vez se tiene un conjunto de candidatos selectos, se aplican técnicas de reproducción por pares de manera aleatoria, donde se busca realizar distintas combinaciones de las listas de ingredientes: concatenación simple de la lista de ingredientes, mantener la primera mitad y concatenar la segunda mitad, mantener la segunda mitad y concatenar la primera mitad, etc. Luego se genera un proceso de mutaciones donde de manera aleatoria se agregan o quitan ingredientes.

Posteriormente se vuelve a realizar una evaluación con el modelo comparando los productos con el modelo. Este proceso se sigue ejecutando a través de varias generaciones hasta llegar a algún criterio de detención. En este caso, la cantidad específica de 100 de generaciones fue utilizado como criterio de detención debido a restricciones de tiempo. Finalmente, se guarda la información de los candidatos que tengan el valor más cercano a 1 (el cual es un valor que nos indica una alta similitud de productos), y estos serán contemplados para el desarrollo del producto deseado.

Este enfoque metódico aprovecha tanto el poder predictivo del aprendizaje automático como el potencial creativo de los algoritmos genéticos para proponer productos alimenticios innovadores que cuenten no solo con propiedades nutricionales similares, sino que también con un perfil de sabor similar, cambiando por completo los ingredientes base.

3. Resultados

3.1. Modelo

Para evaluar el desempeño del modelo, se utilizaron dos conjuntos de datos distintos. El primero es el CP, que incluye el 30% de los productos disponibles. El segundo conjunto, denominado conjunto remanente (CR), consiste en una muestra del 90% restante de los datos originales (34 millones de registros) que no se consideraron para el entrenamiento principal ni para la evaluación. Para CR también se contempló que no hubiera superposición con el CE, resultando en un tamaño similar al del CP.

Las métricas seleccionadas para medir el rendimiento del modelo no requirieron la definición de un umbral específico, o punto de corte. Se utilizó el área bajo la curva

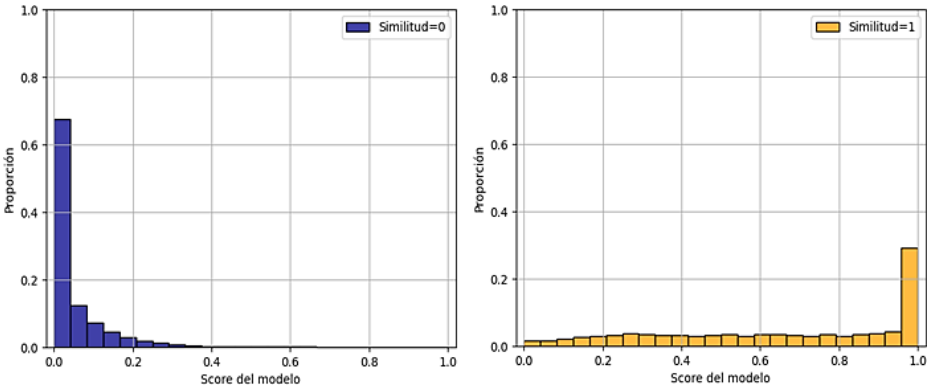


Fig. 1. Histograma que muestra la distribución proporcional del score del modelo para cada etiqueta en el conjunto de prueba; en este caso, la suma de todas las barras en cada gráfico debe resultar en 1. En el lado izquierdo, se presenta la distribución para la clase donde la similitud entre productos es 0 ($S_{ij}=0$). Aquí se observa una concentración predominante de scores bajos, prácticamente el 70% de valores de no-similitud se encuentran en el intervalo donde el score es lo más bajo; lo que indica que el modelo identifica de manera efectiva los casos en que dos productos no son similares. En el lado derecho, se muestra la distribución para los casos donde la similitud entre productos es 1 ($S_{ij}=1$). Esta gráfica refleja una concentración en el extremo de valores altos, el 30% de valores de similitud se encuentran en el intervalo donde el score del modelo es el más alto; sin embargo, también se aprecia que una menor proporción de casos se distribuye en scores más bajos de manera uniforme. En otras palabras, el 70% de casos de similitud están distribuidos similarmente a lo largo de distintos scores del modelo. Esto indica que existen varios casos en las que al modelo le resulta más difícil identificar similitudes entre productos, asignándoles scores relativamente bajos, lo que conduce a un aumento en la incidencia de falsos negativos.

ROC (AUC ROC), el AUC PR y la Precisión Promedio (PP) como indicadores de desempeño.

Los resultados obtenidos demuestran un rendimiento prometedor del modelo en la identificación de similitudes entre productos, y estos se detallan en la Tabla 1.

La distribución del score del modelo muestra una marcada localización en los extremos, dependiendo del valor de la variable de respuesta. Cuando el valor de esta variable es 0 ($S_{ij} = 0$), lo cual indica que dos productos no son similares, se observa que el score del modelo se concentra en rangos bajos.

Por el contrario, un valor de 1 en la variable de respuesta ($S_{ij}=1$) señala que dos productos son similares, y en estos casos, el score del modelo tiende a ser alto, evidenciando una clara separación en las distribuciones. Esta tendencia se ilustra claramente en las figuras 1 y 2 donde, se deduce que la variable de respuesta está notablemente desbalanceada.

La proporción de valores positivos ($S_{ij}=1$) respecto al total es solo del 11.5%, lo que es relevante al evaluar los resultados del modelo, ya que este desequilibrio puede afectar la precisión del modelo. Para determinar una métrica de precisión por intervalo, se analiza la proporción de similitud calculando el promedio de la variable de respuesta en cada intervalo de score, como se muestra en la figura 3.

Cada intervalo presenta un coeficiente de similitud distinto, indicando que en los scores bajos la proporción de similitud es baja. Por ejemplo, en el conjunto de intervalos

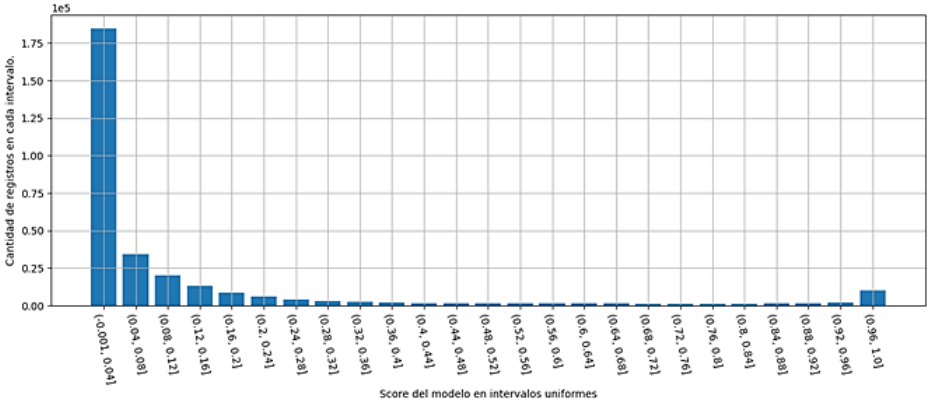


Fig. 2. Histograma que muestra la distribución del score del modelo para todos los registros, utilizando intervalos uniformes. Se observa que la mayoría de los registros se localizan en el lado izquierdo del histograma, mientras que solo unos pocos alcanzan los valores más altos en el lado derecho. A partir de estos datos, se puede calcular la proporción de valores que caen en cada intervalo para evaluar el desempeño del modelo según el rango de score (véase figura 3).

que van desde [0, 0.24] se puede apreciar que la proporción de similitud es menor al 20%, indicando que a scores bajos dos productos no serán similares. Sin embargo, esta proporción aumenta conforme el score se incrementa. Esto sugiere que, para detectar similitudes con alta probabilidad, los scores más altos son mejores indicativos, ya que, en estos intervalos, la proporción de valores similares es significativamente más alta. Por ejemplo, en el intervalo de (0.96 a 1], aproximadamente el 99% de los valores se consideran similares. Por ende, si la comparación de dos productos genera un score del modelo de 0.97, tenemos una alta certeza que van a ser productos similares.

3.2. Algoritmo renético

El algoritmo genético se empleó para generar nuevos candidatos de productos utilizando el resultado del modelo de clasificación binaria como elemento fundamental para optimizar la similitud. Se implementaron tres escenarios específicos, y en cada uno, el algoritmo iteró a lo largo de 100 generaciones, obteniendo así un candidato final que tuviera el score del modelo más alto.

Creación de producto similar al queso. En este escenario, el objetivo era generar un producto que representase el producto generado por la lista en (7):

$$p_i = [\text{Cheese}]. \tag{7}$$

Para evitar la convergencia del algoritmo hacia productos conocidos y prevenir el estancamiento, se excluyeron las siguientes entidades del proceso de generación:

```
['Blue Cheese', 'Camembert Cheese', 'Cheddar Cheese', 'Cheese',
'Comte Cheese', 'Cottage Cheese', 'Cream Cheese', 'Emmental
Cheese', 'Feta Cheese', 'Goat Cheese', 'Gruyere Cheese',
'Limburger Cheese', 'Mozzarella Cheese', 'Munster Cheese',
'Parmesan Cheese', 'Provolone Cheese', 'Ricotta Cheese', 'Romano
```

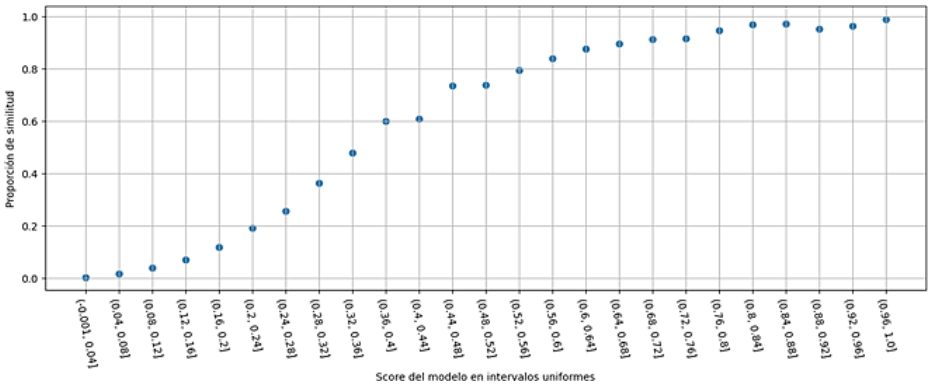


Fig. 3. Proporción de similitud (valor promedio de la variable de respuesta) en cada intervalo de score utilizando el conjunto de prueba. Se observa que a medida que el score aumenta, también lo hace el coeficiente de similitud. Por ejemplo, en los intervalos con scores del modelo que van de 0.72 a 1, la proporción de similitud se aproxima al 90%. Esto indica que, dentro de este rango, es mucho más probable encontrar una alta similitud entre dos productos; es decir, si la comparación de dos productos resulta en un score dentro de estos valores, es altamente probable que sean similares, y esta probabilidad aumenta a medida que el score es más alto. Este patrón se puede utilizar como un mecanismo de calibración para determinar la probabilidad de que un candidato sea similar dado un valor de score específico.

`Cheese', 'Roquefort Cheese', 'Sheep Cheese', 'Swiss Cheese', 'Tilsit Cheese'].`

Tras 100 generaciones, el algoritmo generó un producto como el candidato más cercano:

`['Guinea hen', 'Pacific rockfish', 'Mutton', 'Cream', 'Roe', 'Swordfish', 'Cardamom', 'Milk Powder', 'Sage', 'Sapodilla', 'Margarine like spread', 'Coriander', 'Northern bluefin tuna', 'Rum', 'Raisin', 'Smoked Fish', 'Buttermilk', 'Cognac Brandy', 'Tomato', 'Tamarind', 'Lamb', 'Bonito'].`

Comparando ambos productos con el modelo, el score obtenido fue de 0.58, lo que, según los resultados mostrados en la Figura 3, equivale a una similitud ligeramente superior al 80%. Posteriormente, se llevó a cabo un análisis de las variables más influyentes utilizando SHAP (SHapley Additive exPlanations), para cuantificar la contribución de cada variable en función de la categoría de datos a la que pertenecían. En este análisis, se destacó que los factores nutricionales y las moléculas de sabor fueron los elementos que más influyeron en las decisiones del modelo, como se muestra en la Figura 4.

Creación de producto similar a la leche. En este caso específico, el objetivo era generar un producto representado por la lista `['Milk']`, véase (8):

$$p_i = [\texttt{'Milk'}]. \tag{8}$$

De manera similar al producto pasado, se restringió al algoritmo el acceso a las siguientes entidades: `['Buttermilk', 'Condensed Milk', 'Evaporated`

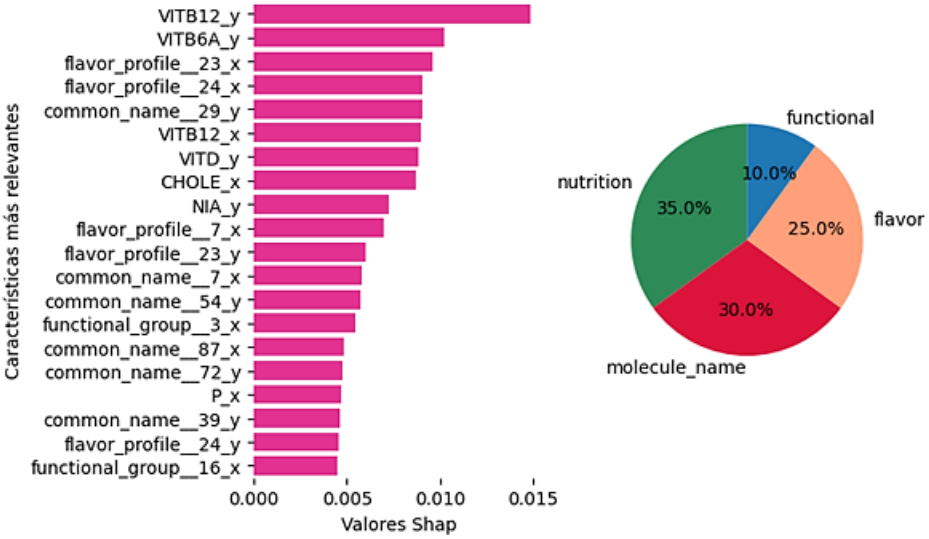


Fig. 4: Las 20 variables más importantes a través de SHAP y porcentaje de representación del grupo al cual pertenecen de los resultados obtenidos del algoritmo genético al realizar una propuesta de queso. En general el sabor se compone de las tres categorías que excluyen nutrición, por lo que en este caso el sabor se tomó como 65% de indicador, mientras que la nutrición solo un 35%. Pero de ese 65%, lo más relevante era la presencia de las moléculas de sabor.

`Milk', 'Goat Milk', 'Milk', 'Milk Fat', 'Milk Human', 'Milk Powder', 'Milkfish', 'Milkshake', 'Sheep Milk', 'Skimmed Milk', 'Soy Milk'].`

Tras 100 generaciones, el algoritmo produjo el siguiente producto: `['Salt', 'Rose hip', 'Cocoa']`. Cuantificando la similitud, se alcanzó un score de 0.63, lo que indica una similitud ligeramente superior al 80%. Es probable que el ingrediente "Cocoa" aparezca recurrentemente en el entrenamiento debido a que probablemente hay mucha leche donde un ingrediente adicional es cocoa, lo que sugiere una alta frecuencia de ocurrencia, para siguientes iteraciones quizá se limite este valor adicional y detectar qué tanto cambian los resultados. En este caso, las moléculas de sabor fueron el factor más influyente en la decisión del modelo, seguido de cerca por la nutrición y el perfil de sabor, como se puede ver en la figura 5.

4. Discusión

Los resultados obtenidos en este estudio indican que es viable generar sustitutos de productos alimenticios mediante el uso de algoritmos genéticos guiados por un modelo de clasificación supervisada. En los resultados de los sustitutos de productos, los scores obtenidos oscilan entre 0.52 y 0.63 en tan solo 100 generaciones.

Esto sugiere que el modelo puede tanto identificar como recomendar ingredientes que potencialmente replican las características de alimentos específicos. Esto responde afirmativamente a nuestra pregunta de investigación sobre si la IA puede mejorar el diseño y personalización de productos alimenticios, al menos en un nivel preliminar.

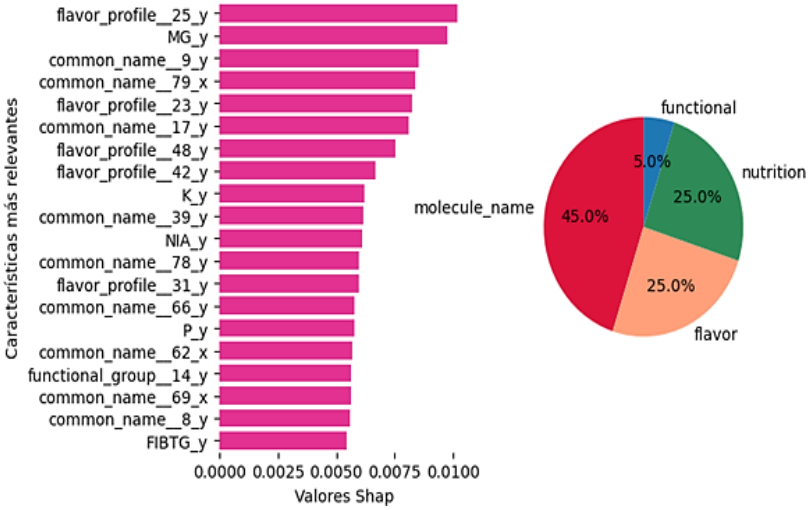


Fig. 5. Las 20 variables más importantes a través de SHAP y porcentaje de representación del grupo al cual pertenecen de los resultados obtenidos del algoritmo genético al realizar una propuesta de leche.

Los hallazgos tienen amplias implicaciones para la industria alimentaria, particularmente en la innovación de productos.

Al poder crear variantes de alimentos existentes que conservan cualidades nutricionales y sensoriales deseables, se podrían diversificar las opciones disponibles para los consumidores y responder mejor a necesidades dietéticas específicas.

Además, esta tecnología podría contribuir a la sostenibilidad, permitiendo el desarrollo de productos que maximicen el uso de recursos disponibles o subutilizados. A pesar de los resultados prometedores, el estudio presenta varias limitaciones.

Primero, la necesidad de utilizar más generaciones para obtener resultados más precisos señala una limitación en la capacidad actual del algoritmo para converger rápidamente hacia la solución óptima.

Además, la falta de validación práctica de los productos generados con expertos en alimentación y nutrición es un aspecto crítico que podría afectar la aplicabilidad real de los sustitutos desarrollados.

Otro aspecto limitante es que el modelo actual no especifica las proporciones de los ingredientes ni sugiere métodos de preparación, lo cual es esencial para la realización práctica de cualquier receta. Para futuras investigaciones, sería beneficioso extender el modelo para incluir recomendaciones sobre las proporciones de ingredientes y métodos de preparación.

Esto haría que los resultados fueran más aplicables en contextos prácticos. Además, incrementar el número de generaciones en los algoritmos genéticos podría mejorar la precisión y relevancia de los productos generados. Sería también esencial implementar estudios que involucren a expertos culinarios y nutricionistas para validar la viabilidad y aceptación de los productos diseñados. Finalmente, explorar la integración de consideraciones sobre aditivos y otros componentes alimentarios en el modelo ampliaría su utilidad.

5. Conclusión

Este estudio ha desarrollado un modelo de clasificación binaria eficaz que determina la similitud entre dos productos alimenticios, exhibiendo métricas de rendimiento relativamente altas.

A través del uso de este modelo, se impulsó un algoritmo genético que identifica iterativamente candidatos potenciales, representados por listas de entidades, seleccionando las mejores propuestas basadas en su similitud con el producto deseado. Las propuestas generadas demostraron tener scores que indican una probabilidad de aproximadamente el 80% de similitud con el producto objetivo. Además, se ha logrado una interpretación detallada de las características más influyentes en cada resultado, proporcionando una cuantificación porcentual de los factores que más contribuyen a la toma de decisiones.

Los hallazgos de esta investigación son significativos para el campo de la ingeniería de alimentos y la tecnología de alimentos, ofreciendo un nuevo enfoque para la creación de productos alimenticios. Este enfoque no solo permite la innovación en términos de desarrollo de productos que puedan satisfacer necesidades específicas de los consumidores, sino que también contribuye a la optimización de recursos y a la reducción del desperdicio alimentario.

Aunque este es solo el comienzo, la aplicación de inteligencia artificial en este contexto abre múltiples oportunidades para mejorar la calidad, accesibilidad y personalización de los alimentos. La investigación sugiere un camino prometedor hacia la integración de técnicas más sofisticadas, como la inclusión de aditivos que mejoren características como la consistencia, el sabor y la estabilidad de los productos alimenticios. Así, este estudio no solo enriquece la comprensión académica y aplicada de la ingeniería de alimentos, sino que también establece una base sólida para futuras investigaciones y desarrollos en el sector.

Agradecimientos. Los autores expresan su más profundo agradecimiento al Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) y al Fondo de Apoyo a la Investigación (FAI) por hacer posible la realización de este proyecto, a través de su impulso a la investigación y apoyo económico. Asimismo, extienden su gratitud al Sistema de Investigación, Ciencia y Tecnología (SICyT) y al Consejo Estatal de Ciencia y Tecnología de Jalisco (COECyTJAL) por el apoyo brindado por medio de la Convocatoria “Grupo de Trabajo Quebec-Jalisco 2023-2024”, el cual fue fundamental para el desarrollo del modelo de inteligencia artificial propuesto. Gracias al respaldo de estas instituciones, se logró avanzar en la exploración de nuevas técnicas aplicadas a la ingeniería de alimentos, sentando las bases para futuras innovaciones en este campo de gran relevancia para la sociedad.

References

1. Arenas, A., Macias, B., Gómez, A. Miramontes, A., Michel, L., Trapero, R., Vela, A., Ramírez, P., Pérez, I., Barrera, M., Ramírez, H., Valdés, J.: Diseño y desarrollo de alimentos con inteligencia artificial. Instituto Tecnológico y de Estudios Superiores de Occidente (2023)

2. Negro, A: Graph-powered machine learning. Manning Publications Co. (2021)
3. Nozaki, N., Konno, E., Sato, M., Sakairi, M., Shibuya, T., Kanazawa, Y., Georgescu, S.: Application of artificial intelligence technology in product design. Fujitsu Scientific and Technical Journal, vol. 53, no. 4, pp. 43–51 (2017)
4. Meeuse, F.M.: Process synthesis for structured food products. Computer Aided Chemical Engineering, vol. 20, pp. 937–942 (2007). DOI: 10.1016/s1570-7946(07)80009-5.
5. Dubbelboer, A., Janssen, J., Krijgsman, A., Zondervan, E., Meuldijk, J.: Integrated product and process design for the optimization of mayonnaise creaminess. Computer Aided Chemical Engineering, vol. 37, pp. 1133–1138 (2015). DOI: 10.1016/b978-0-444-63577-8.50034-6.
6. Zhang, X., Zhou, T., Zhang, L., Fung, K.Y., Ng, K.M.: Food product design: A hybrid machine learning and mechanistic modeling approach. Industrial and Engineering Chemistry Research, vol. 58, no. 36, pp. 16743–16752 (2019). DOI: 10.1021/acs.iecr.9b02462.
7. Varshney, L.R., Wang, J., Varshney, K.R.: Associative algorithms for computational creativity. The Journal of Creative Behavior, vol. 50, no. 3, pp. 211–223 (2015). DOI: 10.1002/jocb.121.
8. Morris, R.G., Burton, S.H., Bodily, P.M., Ventura, D.: Soup over bean of pure joy: Culinary ruminations of an artificial chef. In: International Conference on Computational Creativity, pp. 119–125 (2012)
9. Müller, A.C., Guido, S.: Introduction to machine learning with Python: A guide for data scientists. O'Reilly Media (2016)
10. Shwartz-Ziv, R., Armon, A.: Tabular data: Deep learning is not all you need. Information Fusion, vol. 81, pp. 84–90 (2022). DOI: 10.1016/j.inffus.2021.11.011.
11. Garg, N., Sethupathy, A., Tuwani, R., Dokania, S., Iyer, A., Gupta, A., Agrawal, S., Singh, N., Shukla, S., Kathuria, K., Badhwar, R., Kanji, R., Jain, A., Kaur, A., Nagpal, R., Bagler, G.: Flavordb: A database of flavor molecules. Nucleic Acids Research, vol. 46, no. D1, pp. D1210–D1216 (2017). DOI: 10.1093/nar/gkx957.
12. U.S. Department of Agriculture, Agricultural Research Service: Beltsville Human Nutrition Research Center. FoodData Central (2024)
13. EDAMAM: Food database API (2024)
14. Van-Rossum, G., Drake, F.L.: Python 3 reference manual (2009)
15. The Pandas Development Team: pandas-dev/pandas: Pandas (2020)
16. Harris, C.R., Millman, K.J., van-der-Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van-Kerkwijk, M.H., Brett, M., Haldane, A., del-Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., et al.: Array programming with NumPy. Nature, vol. 585, no. 7825, pp. 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
17. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: Analyzing text with the natural language toolkit. Reilly Media, Inc. (2009)
18. Hunter, J.D.: Matplotlib: A 2D graphics environment. Computing in Science and Engineering, vol. 9, no. 3, pp. 90–95 (2007). DOI: 10.1109/mcse.2007.55.
19. Waskom, M.L.: Seaborn: Statistical data visualization. Journal of Open Source Software, vol. 6, no. 60, pp. 3021 (2021). DOI: 10.21105/joss.03021.
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research, vol. 12, pp. 2825–2830 (2011)
21. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 4768–4777 (2017)

22. Olszewski, D.: Fraud detection using self-organizing map visualizing the user profiles. Knowledge-Based Systems, vol. 70, pp. 324–334 (2014). DOI: 10.1016/j.knosys.2014.07.008.
23. Hsu, Y.C., Lv, Z., Schlosser, J., Odom, P., Kira, Z.: Multi-class classification without multi-class labels. In: Proceedings of the International Conference on Learning Representations, pp. 1–16 (2019)
24. Breiman, L.: Random forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001). DOI: 10.1023/a:1010933404324.
25. Amorim, A., Wanderley-Góes, L.F., Ribeiro-da-Silva, A., França, C.: Creative flavor pairing: Using rdc metric to generate and assess ingredients combination. In: Proceedings of the International Conference on Innovative Computing and Cloud Computing, pp. 33–40 (2017)

ISSN 1870-4069

Detección automática de palabras altisonantes en tweets utilizando redes neuronales

Ricardo Ismael Armas-Araujo[1], Yulia Ledeneva[2]

[1] Universidad Autónoma del Estado de México, México

[2] Instituto Literario, Unidad Académica Profesional Tianguistenco, México

armas5540@gmail.com, yledeneva@yahoo.com

Resumen. En la actualidad, las redes sociales y muchos medios de comunicación cuentan con problemas graves en la moderación de contenidos. La práctica más común en las redes sociales es una comunicación altisonante entre los miembros de las comunidades. Un buen manejo de y monitoreo de los comentarios en las redes sociales requiere de una herramienta actual y útil que nos permita identificar algún conjunto de las palabras más usadas en el lenguaje, para este trabajo será el lenguaje castellano y sus amplias palabras de esta índole. Para tener una gran herramienta de control de palabras altisonantes se explora como una opción viable el uso de las redes neuronales, ya que estas son capaces de aprender de un gran conjunto de datos y tener predicciones cada vez más precisas conforme avanza su entrenamiento. Esto implica mayor nivel de conocimiento de la misma red, que nos ayudara a encontrar este tipo de palabras en distintas frases. Por lo anterior mencionado, se propone en el presente trabajo una serie de pasos para la creación de una red neuronal y su entrenamiento, así como explorar las palabras utilizadas para la red neuronal.

Palabras clave: Redes neuronales, procesamiento de lenguaje natural, redes sociales, palabras altisonantes, análisis de texto.

Automatic Detection of Offensive Words in Tweets Using Neural Networks

Abstract. Currently, social media and many media outlets face serious issues in content moderation. The most common practice on social media is a profane communication among community members. Effective management and monitoring of comments on social media require a current and useful tool that allows us to identify a set of the most used words in the language; for this work, it will be the Spanish language and its extensive words of this nature. To have a great tool for controlling profane words, the use of neural networks is explored as a viable option, as they are capable of learning from a large dataset and making increasingly precise predictions as their training progresses. This implies a higher level of knowledge of the network itself, which will help us find this type of words in different phrases. Therefore, the present work proposes a series of steps

for the creation and training of a neural network, as well as exploring the words used for the neural network.

Keywords: Neural networks, natural language processing, social networks, profane words, text analysis.

1. Introducción

Hoy en día el área de Procesamiento de Lenguaje Natural (PNL) es de gran importancia ya que gracias a esta podemos interactuar entre las computadoras y el lenguaje humano de una manera más amigable. Actualmente, en las empresas se utilizan algunas de las técnicas de PLN en diferentes ámbitos que nos ofrecen una amplia forma de interacción humano-computadora a través del lenguaje, proporcionando así las bases para una mejor comprensión de este contexto. Uno de los puntos clave dentro del PLN son los modelos, como el modelo de representación de palabras, ya que estos nos ayudan a entender el significado de las palabras en su contexto.

Por lo tanto, es indispensable contar con un modelo de representación o detección de palabras, ya que gracias a este se pueden identificar palabras clave en textos largos. De igual manera se utiliza la tokenización para la separación de las oraciones. La aplicación directa de esta investigación pretende mejorar la moderación de las palabras altisonantes en contenido de redes sociales. Cuando se desarrolla un sistema robusto para el reconocimiento de este tipo de palabras se contribuye a la creación de entornos en línea más seguros y civilizados. Los comentarios dentro de la sociedad en muchas ocasiones son mal usados en diferentes contextos. En este artículo se pretende identificar de palabras altisonantes de un corpus de tweets, esto se realizará por el mal uso de estas palabras en la vida cotidiana de las personas.

Sin embargo, nosotros podemos detectarlas antes de que se hagan tweets o publicaciones de odio hacia cualquier persona, evitando de esta manera una discusión por estos puntos. Además, en [1] se menciona que en México el albur se utiliza en todas partes, ya que muchas palabras tienen doble sentido y la combinación de verbos sustantivos, como coincide [9] en la riqueza de este lenguaje. La organización de este trabajo se estructura en secciones que abarcan desde la primer sección de este trabajo, que es la introducción donde se explicó el objetivo de esta investigación; la segunda sección de trabajos relacionados es donde se mencionan trabajos similares que abordan esta misma problemática; en la tercer sección de artefactos propuestos se menciona como se abordó el problema y que metodologías se utilizaron para ello; la cuarta sección menciona los resultados de la investigación y conclusiones generales.

2. Trabajos relacionados

El PLN, como mencionan algunos autores en [10], involucra la detección del análisis semántico en textos, es decir, la interpretación del significado de estos textos.

Esta detección también se enfoca en la interacción entre el lenguaje humano y la computadora, ya que es la estrategia más básica para la detección de palabras en frases, como refiere [3]. La tokenización es un método que facilita la separación de las palabras

Fig. 1. Metodología de una red neuronal.

dentro de los textos [9]. Esta técnica, al hacer esta separación, permite la identificación efectiva y clara de las palabras en las oraciones, según lo mencionado en [5]. Necesitamos de técnicas como la tokenización porque no podemos manejar manualmente todo el contenido dentro de las redes sociales [6].

Teniendo en cuenta este notorio problema, debemos destacar que la tarea puede complicarse por los diferentes significados que algunas palabras clave pueden tener cuando se utilizan en distintos contextos [7], ya que, como se menciona en [1], en el contexto mexicano el albur siempre está presente con un doble sentido, añadiendo más complejidad a la detección de estos mensajes o palabras concretas. Existen muchos modelos para la identificación de palabras en conjuntos de datos.

En [4] se menciona que actualmente algunos autores han utilizado modelos de regresión logística para la detección de estas palabras, aunque con las nuevas tecnologías podemos optar por el uso de redes neuronales convolucionales (CNN). Por otro lado, como se menciona en [8], en la última década se han realizado campañas de evaluación del procesamiento de lenguaje natural y sus enfoques, donde destacan la importancia de la detección de mensajes de odio o altisonantes, ya que estos pueden ayudar a detectar problemas antes de que ocurran.

En el trabajo [2], para el conjunto de datos sobre misoginia que utilizaron, las redes neuronales convolucionales mostraron los mejores resultados en la detección de misoginia, superando a otros modelos de procesamiento de lenguaje en velocidad y eficacia en la detección del lenguaje ofensivo.

3. Metodología propuesta

Se proponen distintos modelos para la detección de las palabras clave en los enunciados que recibimos día con día en nuestras redes sociales. Sin embargo, a menudo no podemos centralizar de manera efectiva cómo nuestros mensajes pueden llegar a ser ofensivos. Por esta razón, se propone el siguiente diagrama de metodología para el desarrollo de una red neuronal convolucional (CNN) que constará de 4 capas.

La primera capa tendrá 1000 neuronas, la segunda no utiliza neuronas, ya que es una operación de reducción de dimensionalidad que calcula el promedio de los valores de características para cada ventana de la secuencia, la tercera contará con 24 neuronas y utilizará una función de activación ReLU, y la última capa estará compuesta por una única neurona, sumando un total de 1025 neuronas. El diseño a continuación (ver Figura 1) nos permitirá identificar eficazmente estas palabras clave.

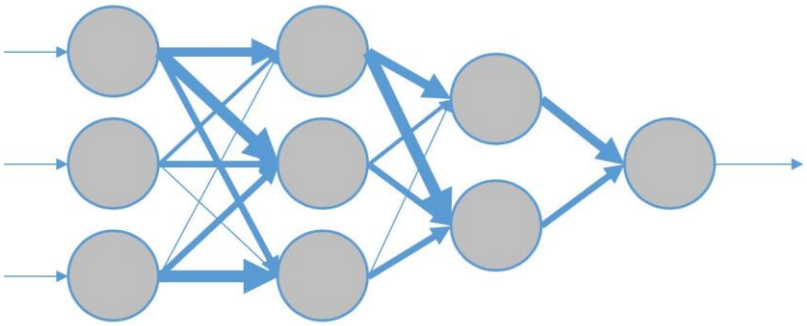


Fig. 2. Modelo de red neuronal.

3.1. Conjunto de datos

En nuestro conjunto de datos, disponemos de una lista de palabras altisonantes que ya han sido extraídas de textos y clasificadas como términos ofensivos o expresiones ofensivas. En este caso, cargaremos solo los términos ofensivos para analizar si la red neuronal puede identificar dichos términos en las frases que le proporcionemos. Además, crearemos algunos datos de ejemplo que contienen frases con términos ofensivos y otras sin ellos.

Clasificaremos las frases como 0 para no ofensivas y 1 para ofensivas. Esto nos permitirá tener un mejor rendimiento al momento de entrenar nuestra red neuronal. Para el análisis, debemos tener en cuenta cómo está estructurado el conjunto de palabras ofensivas en nuestro archivo, que se representa de la siguiente manera en un formato donde los datos se pueden extraer de mejor manera (ver Tabla 1).

3.2. Preprocesamiento de datos

Una vez cargados los datos, procederemos a su manipulación y limpieza. En primer lugar, realizaremos una tokenización. Configuraremos un tokenizador para convertir nuestro texto en secuencias de enteros, donde cada entero representa una palabra en un diccionario que definiremos con un máximo de 1000 palabras.

Finalmente, utilizaremos secuencias que se normalizarán a una longitud fija de 10, rellenando con ceros al final si son más cortas. La tokenización es una de las partes cruciales del procesamiento de datos [10] y para la red neuronal propuesta. Por lo tanto, un buen preprocesamiento de los datos nos permitirá tener una mayor precisión al momento de crear nuestro modelo de red neuronal.

3.3. Construcción del modelo de red neuronal

Una red neuronal tiene la característica de simular o imitar el proceso de aprendizaje del ser humano [11]. Este proceso se centra en la construcción de un modelo con neuronas unidas por una serie de caminos. Se seleccionó este modelo debido a su notable eficacia mencionada en trabajos relacionados.

Tabla 1. Formato de términos ofensivos.

Columna 1	Columna 2	Columna 3
T1	TERMINO_OFENSIVO	tonto
T2	TERMINO_OFENSIVO	hipócrita

Para la metodología propuesta, utilizaremos 4 capas en la red neuronal, como se muestra a continuación: Para la construcción de nuestro modelo, primeramente, debemos utilizar una capa de embedding, que se define para lograr un mejor entrenamiento de las redes neuronales [12]. En términos más explícitos, es una capa que convierte los índices de palabras en vectores densos de tamaño fijo.

De igual manera, utilizamos una capa de pooling para reducir la dimensionalidad promedio de las características. Al final, utilizamos capas densas (dense) para la clasificación y la salida binaria, donde 0 indica no ofensivo y 1 indica ofensivo. Para esta red neuronal, utilizaremos un total de 4 capas, como se mencionó anteriormente (ver Figura 2).

Como se observa en la imagen, las 4 capas nos permitirán manipular de mejor manera las palabras que se deben detectar. En la primera capa, se ingresan los textos tokenizados para que la capa de embedding convierta las palabras en vectores. Posteriormente, la segunda capa realizará el pooling para reducir la dimensionalidad de los vectores de características. La tercera capa transformará los vectores mediante una función de activación ReLU. Finalmente, la última capa determinará la clasificación como ofensiva o no ofensiva.

3.4. Entrenamiento del modelo

En este apartado, nuestro modelo comienza a utilizar todo lo realizado anteriormente para un buen entrenamiento. Un buen entrenamiento debe contar con una cantidad suficiente de datos que nos permita analizar varios puntos de vista. Existen distintos caminos para lograr el objetivo, que en este caso es la identificación de las palabras altisonantes.

En nuestro caso, pasamos los datos al modelo e indicamos cuántas iteraciones debe realizar. El objetivo es que en cada iteración la precisión en la identificación de las palabras mejore progresivamente. Además, definimos una función dentro de un condicional (if) con el conjunto de palabras altisonantes. Esta función se encargará de identificar si alguna palabra altisonante de la lista se encuentra dentro del mensaje enviado por el usuario.

3.5. Predicción del modelo

La última parte es la predicción, donde verificamos si se encuentra un término ofensivo. Mandamos un mensaje que teclea el usuario y este se compara con la lista de palabras altisonantes. Esto se logra convirtiendo el texto obtenido del usuario a una secuencia y luego a un vector, de la misma forma que los datos de entrenamiento. Finalmente, se realiza la predicción. Utilizaremos 10 épocas para la detección de las palabras altisonantes, como se mencionó anteriormente.

Tabla 2. Datos resultantes ofensivos obtenidos con la red neuronal.

Época	Pérdida	Exactitud
1	0.6935	0.5200
2	0.6923	0.6200
3	0.6912	0.6600
4	0.6903	0.6800
5	0.6893	0.7000
6	0.6883	0.7000
7	0.6872	0.7200
8	0.6960	0.7600
9	0.6847	0.8200
10	0.6833	0.8400

Esto se hace con la finalidad de que el modelo de red neuronal tenga más datos para reducir la pérdida y mejorar su exactitud al momento de predecir una oración con palabras altisonantes.

4. Experimentación y resultados

Una vez realizados todos los puntos de la metodología, se efectúan iteraciones en la red neuronal e identificamos problemas que surgen al aplicarla, especialmente cuando una palabra ingresada tiene algún carácter especial o alguna característica que modifica la palabra. Esto puede influir en la identificación de un término ofensivo, ya que las palabras usadas en el modelo no incluyen caracteres especiales. Otro problema es la clasificación de cada palabra; dado que el corpus para nuestro estudio también incluye expresiones ofensivas, es crucial dividir estas adecuadamente para evitar clasificaciones erróneas y, por ende, identificaciones incorrectas por parte del modelo.

Como se observa en la tabla anterior (Tabla 2), podemos denotar como en cada iteración la exactitud (accuracy) va en aumentos lo que significa que se está reconocimiento de mejor manera las palabras en una posible frase, en este caso en particular podemos observar cómo se manda una frase con algunos términos ofensivos y como se realiza el hallazgo de ciertos términos.

En este otro caso (Tabla 3), podemos observar cómo se le llega a mandar una frase que no resulta tener alguna de estos términos ofensivos, lo que se observa por las métricas de perdida y precisión del modelo, así como el resultado final que dentro de la ejecución del programa se manda un mensaje donde se menciona que no se llegan a determinar palabras obscenas dentro de nuestro mensaje.

Del conjunto de palabras altisonantes se determina cuáles son las principales palabras que se llegan a repetir o utilizar más en nuestro conjunto de datos: La gráfica anterior (Figura 3) muestra que la mayor palabra que se encontró es mierda con 1480 repeticiones, puto en segundo lugar con 804, puta con 706 repeticiones, pringada con 440 repeticiones, gorda con 336 repeticiones, coño con 331 repeticiones. Una vez vista

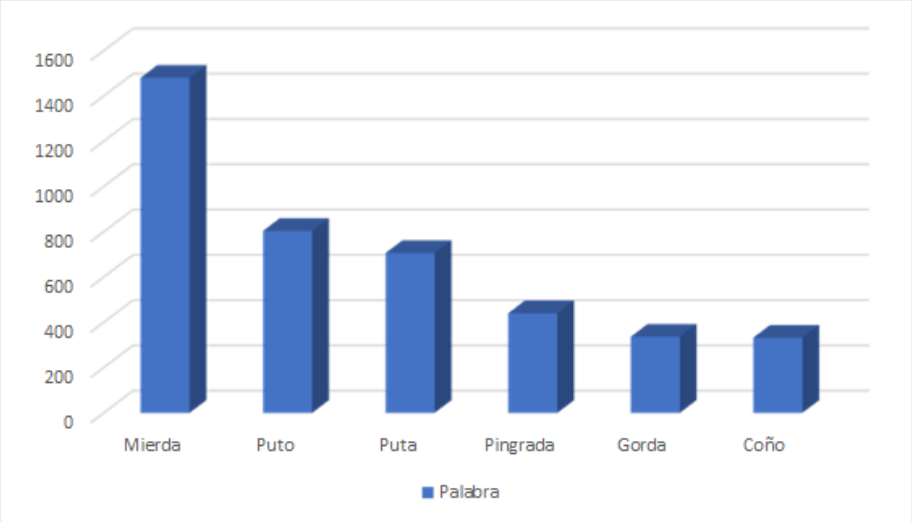


Fig. 3. Palabras más usadas en el conjunto de datos.

la forma en que nos arrojó las palabras podemos determinar que el uso de una red neuronal nos ayudó a realizar la detección de manera precisa y con un tiempo no tan excesivo; contamos con que la red neuronal tarde alrededor de 5 minutos en el entrenamiento de las épocas y notamos que los resultados fueron favorables, la red detecta de buena manera las palabras en el corpus, el único inconveniente presente es que si nosotros utilizamos palabras que se encuentran fuera del corpus o con modificaciones de doble sentido como puede ser mi3rda, utilizando un número en lugar de una letra, el modelo no lo llega a detectar; para mejores resultados podemos expandir el corpus y complementar el idioma castellano para la mejora de detección.

4.1. Parámetros de la red neuronal

Para la red neuronal propuesta se comenzó con un proceso de tokenización, que utiliza Keras como tokenizador, este a su vez fue configura con un máximo de 1000 palabras basadas en la frecuencia de aparición, lo que indica que solo las mil palabras más frecuentes se usaran para el entrenamiento de nuestra red, y cualquier otra palabra se marcara como "<OOV>" que representa las palabras fuera del vocabulario. Para las secuencias se hace un truncado a rellenado para asegurar que tengan longitud uniforme de 10 palabras, esto es necesario para que el modelo procese los datos de entrada de una manera más consistente, ya que las entradas de las redes neuronales deben ser del mismo tamaño.

Para la arquitectura de la red neuronal, se comienza con una capa Embedding que transforma los índices en las palabras de vectores densos de 16 dimensiones; la primera capa nos ayuda a que el modelo pueda aprender representaciones diversas y útiles basadas en su contexto; por otro lado, la segunda capa es una `GlobalAveragePooling1D´ que nos ayuda a reducir la dimensionalidad promedio de las incrustaciones dentro de la secuencia, para simplificar nuestra red. De igual manera,

Tabla 3. Datos resultantes no ofensivos obtenidos con la red neuronal.

Época	Pérdida	Exactitud
1	0.6919	0.5400
2	0.6899	0.6400
3	0.6881	0.8200
4	0.6863	0.8600
5	0.6842	0.9600
6	0.6819	0.9800
7	0.6798	0.9600
8	0.6772	0.9400
9	0.6747	0.9800
10	0.6719	0.9800

se introducen otras dos capas densas conectadas entre sí, la primera de 24 nodos con una función ReLU, que nos ayuda a introducir no linealidades en el modelo, aprendiendo así patrones más complejos, y la segunda capa que es densa con un solo nodo y una función de activación sigmoidea, comúnmente utilizada en problemas de clasificación binaria como en nuestro caso si un texto es ofensivo o no.

Para la compilación del modelo, este compila una función de perdida 'binary_crossentropy', la cual es eficiente para comparar las salidas de la función sigmoidea con las etiquetas binarias en los datos del entrenamiento; también se hace uso de un optimizador Adam, indispensable por su eficacia y ajuste automático del ritmo en que se da el aprendizaje, por último, la exactitud se usa para monitorear el rendimiento del modelo en el entrenamiento.

Finalmente, nuestro modelo se entrena por medio de 10 épocas, haciendo que las representaciones aprendidas por las iteraciones minimicen la perdida y de una mejora en la precisión de la detección de las palabras altisonantes según su contenido ofensivo. El enfoque que se presenta combina técnicas de procesamiento de lenguaje natural con aprendizaje profundo, ofreciendo un modelo que sea capaz de identificar y detectar palabras ofensivas basadas en características lingüísticas especiales.

4.2. Descripción de corpus

OffendES_spans es un corpus en español creado a partir del corpus OffendES, con la identificación automática de términos ofensivos utilizando el lexicón SHARE [13]. El corpus consta de 47.128 comentarios anotados con términos y expresiones ofensivos. Los comentarios fueron anotados de manera manual utilizando un esquema de anotación detallado. Los comentarios fueron recopilados de diferentes redes sociales: Twitter, Instagram y YouTube. Los comentarios publicados fueron ofensivos o hirieron los sentimientos de otros usuarios según su género, raza, religión, ideología u otras características personales.

Tabla 4. Presencia de términos ofensivos de léxicos en los comentarios recuperados.

Red Social	Término ofensivo	Término no ofensivo	Total
YouTube	19,449	184,414	203,863
Instagram	3,142	58,209	61,351
Twitter	1,197	18,728	19,925
Total	23,788	259,865	283,622

Primero, se recolectaron un total de 283.622 comentarios (ver la Tabla 4). Luego, los comentarios se filtraron según dos principales limitaciones: la presencia de lenguaje potencialmente ofensivo y diversidad léxica. Para evitar la creación de un corpus con pocos o ningún comentario ofensivo, se etiquetaron todos los comentarios con banderas que determinaban si el comentario contenía alguna de las palabras encontradas en cinco léxicos controlados diferentes [14]. Se seleccionaron todos los comentarios con lenguaje potencialmente ofensivo (23.788 comentarios).

5. Conclusiones y trabajo futuro

Como llegamos a ver los entrenamientos de redes neuronales no son una tarea fácil, ya que implica un buen entrenamiento, creación y carga de la misma red neuronal, de igual manera debemos considerar nuestros datos, ya que por medio de estos nuestra red tendrá un cierto porcentaje de error o asertividad.

Como todo lo nuevo si no llegamos a profundizar en el tema tendremos problemas al momento de realizar un programa de esta índole, una vez conociendo el entorno y los métodos básicos utilizados como la tokenización, podemos avanzar en los modelos de redes sin mayor complejidad, lo que de igual manera tendrá un impacto en los resultados.

La propuesta de red neuronal explorada nos ayuda a comprender la identificación de las palabras altisonantes castellanas en una frase enviada por el usuario, algunos aspectos de las palabras pueden hacer que el modelo las identifique o no, sin embargo, con un conjunto de palabras con acentos o algún otro carácter especial contenida dentro de esta no tendrá mayor problema de identificación.

Cada una de las fases exploradas corresponde a puntos cruciales en el diseño de una red neuronal, la fase donde procesamos los datos es una de las más importante de todas las vistas, ya que en ella se puede visualizar la subida de los datos para nuestra red neuronal y si se llegan a presentar fallos puede comprometer nuestro modelo y sus parámetros.

Como trabajo a futuro podemos destacar lo anteriormente mencionado un conjunto de datos con distintos tipos de caracteres especiales puede ser un reto para identificar más palabras, ya que muchas veces la gente cambia incluso una letra por un número para que las palabras sean difíciles de identificar para los algoritmos como el presentado en este trabajo. Si contemplamos otros tipos de modelos y datos especiales para la red neuronal propuesta podremos mejorar algoritmos como estos para que identifiquen palabras como las ya mencionadas.

Referencias

1. Guzmán, E., Beltrán, B., Tovar, M.: Clasificación de frases obscenas o vulgares dentro de tweets. Research in Computing Science, vol. 85, pp. 65–74 (2014)
2. De la Peña, G.: Análisis y detección de odio en mensajes de Tweeter. Universitat Politècnica de València, pp. 11–14 (2019)
3. De la Peña, G.: Deep Analyzer at SemEval-2019 Task 6: A deep learning-based ensemble method for identifying offensive tweets. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 582–586 (2019). DOI: 10.18653/v1/S19-2104.
4. Shushkevich, E.: Automatic misogyny detection in social media: A survey. Computación y sistemas, vol. 23, no. 4, pp. 1159–1164 (2021). DOI: 10.13053/cys-23-4-3299.
5. Clarke, I., Grieve, D.J.: Dimensions of abusive language on twitter. In: Proceedings of the First Workshop on Abusive Language Online, pp. 1–10 (2017). DOI: 10.18653/v1/w17-3001.
6. Ahluwalia, R. Shcherbinina, E., Callow, E., Nascimento, A.C., De-Cock, M: Detecting misogynous Tweets. University of Washington, pp. 1–7 (2018)
7. Canós, J.S.: Misogyny identification through SVM at IberEval 2018. Universidad Politécnica de Valencia, pp. 229–233 (2018)
8. Frenda, S., Ghanem, B., Montes-y-Gómez, M.: Exploration of Misogyny in Spanish and English tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval´18), colocated with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN´18), vol. 2150, pp. 260–267 (2018)
9. Ramos, O.: Análisis sobre el idioma español en México, con base en la frecuencia de palabras azules rojas y obscenas y vulgares en Twitter. Universidad de Puebla, pp. 1–8 (2016)
10. Pérez, A.: Detección del discurso de odio de twitter. Universidad Politécnica de Valencia, pp. 27–33 (2020)
11. Castaneda-Sanchez, W.A., Polo-Escobar, B.R., Vega-Huincho, F.: Artificial neural networks: A measurement of forecast learnings as potential demand. Universidad Ciencia y Tecnología, vol. 27, no. 118, pp. 51–60 (2023). DOI: 10.47460/uct.v27i118.686.
12. López, D.: Aprendizaje profundo para la extracción de aspectos en opiniones textuales. Revista Cubana de Ciencias Informáticas, vol. 13, no. 2, pp. 105–145 (2019)
13. Plaza-del-Arco, F.M., Montejo-Ráez, A., Ureña-López, L.A., Martín-Valdivia, M.T.: OffendES: A new corpus in spanish for offensive language research. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pp. 1096–1108 (2021)
14. Plaza-del-Arco, F.M., Molina-González, M.D., Ureña-López, A., Martin, M.: Sinai at semeval-2020 task 12: Offensive language identification exploring transfer learning models. In: Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 1622–1627 (2020). DOI: 10.18653/v1/2020.semeval-1.211.

ISSN 1870-4069

Aplicación de regresión polinomial como método de predicción de datos en señales obtenidas a partir de movimientos de cabeza

Luis Alberto Hernández Montiel, Edmundo Bonilla Huerta, Edwyn Martínez Carrillo, Roberto Morales Caporal

Tecnológico Nacional de México,
Campus Apizaco, Tlaxcala,
México

{D23370018, edmundo.bh, M22371203,
roberto.mc}@apizaco.tecnm.mx

Resumen. En este artículo, se presenta un modelo basado en regresión polinomial como método de predicción de datos, para homogeneizar las señales obtenidas de movimientos de cabeza que realiza una persona al ser estimulada para generar una emoción. Primero, se estimula a la persona proyectándole vídeos para generar la emoción de disgusto. Después, se rastrean los macro y micro movimientos que realizó el participante durante la estimulación, mediante la transformación de las muestras del dominio espacial a un dominio frecuencial utilizando análisis de vídeo y momentos de Hu. A las señales obtenidas de este proceso, se les aplica regresión polinomial como técnica de predicción de datos para homogeneizar las señales a un tamaño estándar. Los resultados que genera este algoritmo muestran una predicción más ajustada al patrón que sigue la señal generada por los movimientos corporales de una persona.

Palabras clave: Movimientos de cabeza, regresión polinomial, análisis de vídeo, momentos Hu.

Application of Polynomial Regression as a Data Prediction Method in Signals Obtained from Head Movements

Abstract. In this article, a model based on polynomial regression as a data prediction method to homogenize the signals obtained from head movements made by a person when stimulated to generate an emotion is presented. First, the person is stimulated by projecting videos to generate the emotion of disgust. Then, the macro and micro movements performed by the participant during the stimulation are tracked by transforming the samples from the spatial domain to the frequency domain using video analysis and Hu moments. To the signals obtained from this process, polynomial regression is applied as a data prediction technique to homogenize the signals to a standard size. The results generated by this algorithm show a more accurate prediction of the pattern that follows the signal generated by the body movements of a person.

Keywords: Head movements, polynomial regression, video analysis, Hu moments.

1. Introducción

El reconocimiento de emociones es un campo en la computación que ha crecido en los últimos años. Diversas investigaciones se han enfocado en obtener una mejor comprensión de las diferentes emociones que expresa una persona durante el día. Una de las principales áreas donde se muestran diferentes expresiones es el rostro, ya que gracias a las gesticulaciones que la persona hace, se puede obtener una gran variedad de datos de diferentes emociones. Pero, las distintas partes del cuerpo pueden dar información sobre estos cambios de estados emocionales, tales como el torso, las manos, la piel (cambio de temperatura) e incluso los pies. En este trabajo, se propone el estudio de los movimientos de la cabeza de una persona como método de reconocimiento de emociones. Primero se estimula una emoción en un participante, utilizando vídeos seleccionados específicamente para la emoción de asco. Después se filman las reacciones (movimientos) de la cabeza del participante, convirtiéndolas en una señal que es reestructurada con regresión polinomial para encontrar la emoción que está presentando la persona.

2. Estado del arte

La necesidad La necesidad de reconocer emociones de forma automática da lugar a proponer nuevos estudios utilizando estas las reacciones del cuerpo como datos para analizar una emoción y crear nuevas terapias o dispositivos para una mejor calidad de vida [1]. Pero hacer un estudio de las emociones que expresa una persona no es una tarea fácil. En muchas ocasiones, una persona reacciona con una expresión muy pronunciada y otras no realizan ninguna, además de que todas las personas se expresan de forma diferente en una misma situación. Esto implica la implementación de sistemas que utilicen más de un dispositivo para reconocer dicha emoción.

Con este enfoque, se ha propuesto la utilización de sistemas multimodales [2,3] capaces de analizar diferentes partes del cuerpo al mismo tiempo. Otra técnica propuesta es el estudio del parpadeo. En los trabajos de Maffei [4] y Demiral [5], se analiza la asociación que existe entre el parpadeo y el estado afectivo de la persona, para encontrar una emoción específica. Otros trabajos proponen la utilización de rasgos gestuales no verbales [6] para detectar emociones, entre ellos los movimientos oculares [7], movimientos de pies [8] y movimientos corporales [9].

A pesar de las diferentes técnicas propuestas, aún no se llega a una solución concreta; por lo tanto, siguen apareciendo nuevas propuestas que ayudan a reconocer emociones. En este trabajo, se propone el análisis de los movimientos de la cabeza de una persona para crear señales que puedan ayudar a reconocer una emoción. Primero, se estimula al participante utilizando vídeos específicos para la emoción de disgusto, después se rastrean esos movimientos para trastornarlos en una señal. El siguiente paso es utilizar la regresión polinomial como método de predicción de datos para transformar la señal y así dejarla lista para su análisis. El método se describe a continuación.

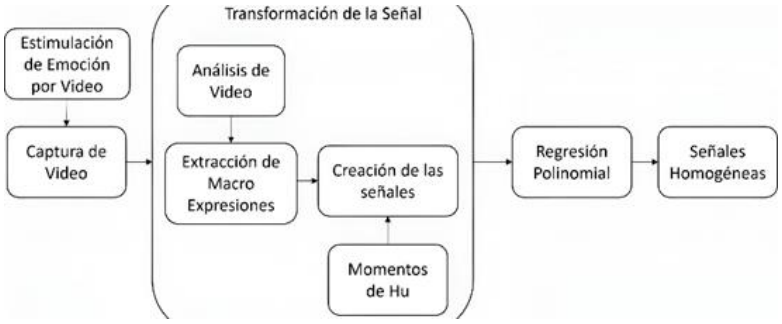


Fig. 1. Diagrama general del método propuesto.

3. Materiales y métodos

Encontrar patrones en un vídeo es una tarea compleja, ya que muchos de los vídeos tienen poca iluminación, ruidos ambientales o desfases de tiempo, además de que requiere un costo computacional elevado. Este problema genera la necesidad de crear algoritmos capaces de trabajar bajo estas condiciones y extraer características relevantes de una secuencia de vídeo. Para dar solución a este problema, se propone la trasformación del vídeo de su forma espacial a una forma de frecuencia para formar una señal que capture las macro y micro expresiones generadas por un participante. Para ello, se propone un algoritmo combinando técnicas de visión artificial, momentos de Hu y regresión polinomial para registrar y mejorar las señales generadas por los diferentes movimientos de cabeza que realiza una persona que ha sido grabada. La figura 1 muestra el proceso general del algoritmo propuesto.

3.1. Creación de la señal

Para crear una señal que capture los movimientos de cabeza que realizó el participante, se proponen 3 fases. Primero, se estimula a la persona proyectándole vídeos para generar una emoción específica. En la fase dos se hace la transformación del vídeo de un espacio visual a uno de frecuencias. En la fase tres se homogeneizan las señales aplicando regresión polinomial como técnica de predicción de datos. Con estas tres fases se pretende hacer un preprocesado para clasificar las señales. Cada una de las fases se describe a continuación.

3.1.1. Estimulación de emociones y captura de video

En esta fase, se estimula al participante proyectándole vídeos para generarle la emoción de disgusto. La toma de muestras para este estudio se realizó en un área controlada en un espacio de 2x4 m, evitando corrientes de aire, ruidos ambientales y se iluminó con luz natural. La recolección de muestra se realizó entre las 9:00 y 11:00 a. m. en un lapso de dos semanas a un grupo de 24 estudiantes conformado por 13 hombres y 11 mujeres del Instituto Tecnológico de Apizaco, con un rango de edad de 18 a 27 años.

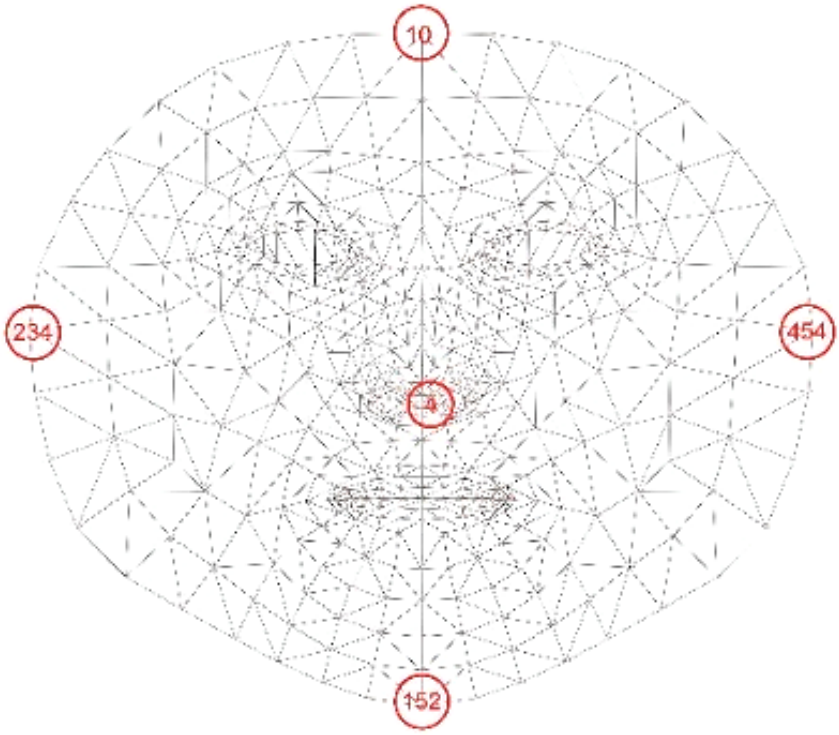


Fig. 2. Malla generada por face_mesh de mediapipe.

Para la estimulación de la emoción, se proyectó un vídeo de los 70 que contiene la base de datos de LATEMO-E [10]. Este video tiene el porcentaje más alto de vistas para la emoción de "Disgusto", el vídeo es "Planet Terror", con 79%. El vídeoclip tiene una duración promedio entre 1 y 2 minutos. Los criterios de selección de este vídeo fueron los siguientes: (1) la película fue sugerida con más frecuencias, (2) la coherencia del argumento de la escena con el tema que se está abordando. El material multimedia fue presentado en una laptop Lenovo IdeaPad Slim 3 con una pantalla de 15.6" y altavoces integrados. La cámara utilizada para grabar fue la cámara web integrada a la laptop con una resolución de 1080p enfocada a la parte superior del cuerpo, colocada a 1 metro de distancia del participante.

3.1.2. Captura del movimiento

Para trasformar el vídeo a una señal, se hace lo siguiente. Primero se implementa la librería OpenCV para filmar los movimientos que hace el participante al ser estimulado. Esta librería se utiliza en aplicaciones de visión por computadora en tiempo real [11]. Con la implementación de esta librería capturamos los movimientos de cabeza del participante. Se utilizan solo los movimientos de esta parte del cuerpo porque es una de las partes que tienen más reacciones durante la estimulación.

Fig. 3. Puntos seleccionados con mediapipe para rastrear el movimiento.

El siguiente paso es analizar el vídeo para rastrear los movimientos de la cabeza, para ello utilizamos la librería Mediapipe [12]. Esta librería es un framework basado en aprendizaje automático para procesar datos de vídeo, dibujando una serie de puntos conectados entre sí para crear distintas soluciones como detectores de pose, manos, caras, u objetos, además de que tiene una función específica para cada parte del cuerpo lo que permite trabajar con ellas por separado. Para rastrear el movimiento de cabeza, se utiliza la función mediapipe fase_mesh [13]. Esta librería crea una malla de puntos que cubren la cabeza (figura 2).

Para detectar el movimiento de la cabeza, se localizan cinco puntos de referencia de toda la cabeza, los cuales son la frente, la barba, los laterales cercanos a las orejas y la nariz. Con el uso de estos puntos, se logra rastrear los movimientos de la cabeza del participante. Al tener todos los puntos localizados dentro del vídeo, el siguiente paso es convertir el movimiento capturado en forma de frecuencia y crear una señal a partir de esos movimientos.

3.1.3. Transformación de la señal

Después de rastrear los movimientos (plano visual del vídeo), el siguiente paso es transformarlos en una señal (plano de frecuencias). Esto se realiza de la siguiente forma. Para transformar los movimientos de cabeza encontrados, lo primero que se hace es encontrar las coordenadas de cada uno de los puntos seleccionados y etiquetarlas para ver los movimientos que hace cada uno de esos puntos de interés. Después de que se obtiene el ancho y alto del fotograma del vídeo, el siguiente paso es utilizar los puntos de referencia (landmarkers) que fueron encontrados en la zona de interés de la persona. Estos landmarkers se obtienen al aplicar la siguiente ecuación:

$$\begin{aligned}(x) &= LMs[i].x * w, \\ (y) &= LMs[i].y * h,\end{aligned} \tag{1}$$

donde "x", "y" son las coordenadas de interés, son las landmarkers generadas por la función de mediapipe, [i] son los puntos con los que se rastreará el movimiento. "h" es el alto y "w" es el ancho del fotograma.

Con esta ecuación es posible obtener distintos pares de coordenadas en los ejes (x, y). El siguiente paso es aplicar momentos invariantes. Los momentos invariantes

Fig. 4. Señal obtenida por el movimiento de cabeza del participante.

propuestos por Hu [14] extraen características para reconocer objetos que no se encuentran en la misma posición. En nuestro caso, utilizamos el momento de Hu para calcular el punto exacto donde el landmark genera un movimiento de la siguiente forma [14]:

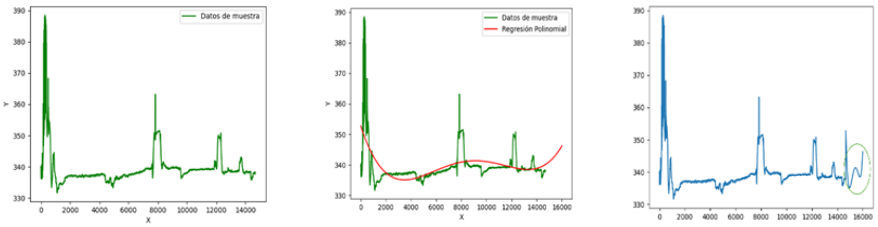
$$m_{pq}=\sum_{x}\sum_{y}(x)^{p}(y)^{q}f(x,y), \tag{2}$$

donde m_{pq} es el momento resultado dentro Dell fotograma, "p" y "q" son el orden del momento, "x", "y" son las coordenadas correspondientes al píxel dentro del fotograma. Al aplicar este método, se ha trasformado el vídeo de su plano visual a un plano de frecuencias, creando diferentes señales para cada participante. El siguiente paso es homogeneizar esas señales, lo cual se explica a continuación.

3.1.4. Homogeneización de la señal

Al aplicar los pasos anteriores, se genera una señal que ha capturado los macro y micro movimientos de los participantes durante la estimulación de una emoción y las ha transformado en una señal con diferentes tamaños de longitud, unas más largas que otras. Esto sucede porque el algoritmo solo rastrea los movimientos que hace el participante al ser estimulado sin importar el tamaño del vídeoclip, además de que un participante puede tener una reacción en un punto del vídeoclip diferente al otro participante y, aunque el algoritmo registra las dos reacciones, no crea la misma señal para los dos. Esto genera un problema ya que, al no tener señales con tamaños iguales, se puede crear un sobre entrenamiento en un algoritmo de clasificación.

Para solucionar este problema, se propone utilizar un método de regresión polinomial que sirve para homogeneizar las señales. La regresión polinomial es un modelo de análisis de regresión en el que la relación entre la variable independiente X y la variable dependiente Y se modela con un polinomio de n-ésimo grado en X. Para



(a) (b) (c)

Fig. 5. Resultado al aplicar regresión polinomial.

obtener la regresión polinomial se debe calcular el polinomio con la siguiente ecuación [15]:

$$y=\beta 0 \ + \ \beta 1X \ + \ \beta 2X^2 \ + \ \beta 3X^3 \ + \ \cdots \ + \ \beta pX^p \ + \ \varepsilon, \tag{3}$$

donde ε representa el error en la estimación o la diferencia entre el valor estimado y el valor observado. El polinomio resultante de la ecuación 3, puede ser abordado mediante una regresión lineal múltiple aplicando las ecuaciones 4 y 5:

$$X1 = X, X2 = X2, \ldots., Xp = Xp, \tag{4}$$

$$Y = \beta 0 + \beta 1X^1 + \beta 2X^2 + \beta 3X^3 + \beta pX^p + \varepsilon. \tag{5}$$

Para la obtención de la regresión lineal múltiple, es posible plantear el modelo a partir de una matriz para *n* muestras de datos, tal como se presenta en la ecuación 6:

$$\begin{pmatrix} y1 \\ y2 \\ \vdots \\ yn \end{pmatrix} = \begin{pmatrix} 1 & x11 & \cdots & x1p \\ 1 & x21 & \cdots & x2p \\ 1 & \vdots & \cdots & \vdots \\ 1 & xn1 & \cdots & xnp \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_p \end{pmatrix}. \tag{6}$$

De este modo, mediante la notación de matrices es posible expresar la ecuación 6, tal como se muestra en la ecuación 7:

$$Y \ = \ X\beta \ + \ \varepsilon. \tag{7}$$

Dado que el objetivo de la ecuación matricial 7, es encontrar el vector de coeficientes β, es posible mediante propiedades matriciales (matriz traspuesta y matriz inversa) expresar la ecuación 8:

$$\beta \ = \ (XTX) - 1XTY. \tag{8}$$

Así, el vector resultante β contiene los diferentes coeficientes ($\beta 0$, $\beta 1$, $\beta 2$, …, βp) del polinomio presentado en la ecuación 3. Al aplicar esta técnica, se puede asegurar que todas las señales tendrán el mismo tamaño y no hay grandes variaciones entre ellas, preparándolas para su clasificación.

4. Experimentos y resultados

En esta sección se muestran los resultados más relevantes de cada una de las fases del proceso. Los experimentos fueron realizados en el lenguaje de programación Python 3.11.5, utilizando las librerías mediapipe, opencv y scikit-learn. El algoritmo fue desarrollado en una laptop Dell inspiron 3000 con un procesador Ryzen 7 y 24 GB de RAM. Los mejores resultados son descritos a continuación. Se obtuvo un video por cada participante, resultando 24 videos diferentes. Para transformar el vídeo de su plano visual a una señal, se hace un análisis del vídeo, lo cual nos permite obtener los diferentes movimientos que hace el participante durante la estimulación.

Este análisis se realiza utilizando los puntos que generan las funciones de la librería MediaPipe y los resultados de este paso se muestran en la figura 3. En la figura 3 se observan los 3 puntos seleccionados para la cabeza, los cuales se encuentran en la barba, frente, laterales (cercanos a los oídos) y finalmente la nariz. Con estos 3 puntos es posible abarcar toda la zona de interés, en este caso la cabeza. Al tener los puntos específicos seleccionados, el siguiente paso es rastrear los movimientos que hace el participante durante la proyección del vídeoclip.

Para solucionarlo, se buscan las coordenadas de cada punto (landmarkers) dibujados en la figura 5 utilizando la ecuación 1. Al obtener esas coordenadas, el siguiente paso es aplicar momentos invariantes de Hu (ecuación 2), los cuales convierten el comportamiento de las coordenadas en una frecuencia dentro de un hiperplano. Esto indica que cada vez que el participante hace un movimiento (macro o micro), el momento de Hu lo transforma en una perturbación dentro de la señal que se está capturando.

A mayor movimiento, mayor es la perturbación dentro de la señal; por el contrario, pequeñas o nulas perturbaciones se capturan cuando la persona está sin movimiento. La figura 4 muestra la señal generada por la parte del cuerpo seleccionada. La figura 4 muestra la señal obtenida al aplicar los momentos de Hu a las coordenadas de los landmarkers marcados en la figura 3. La imagen muestra que un mayor movimiento se trasforma en una perturbación más grande que el resto de la señal. Esto permite crear señales con picos más grandes debido a las diferentes perturbaciones que los momentos de Hu han identificado.

Al analizar los vídeos de cada uno de los participantes, se puede notar que las 24 señales creadas no son iguales, tienen diferentes tamaños, lo cual puede generar un error al momento de entrenar un algoritmo de aprendizaje. Para solucionar este problema, aplicamos el método de regresión polinomial (ecuación 3), el cual predice datos que nos ayudan a complementar las señales obtenidas por los momentos de Hu y que todas las señales tengan un tamaño estándar. En este experimento, se utilizó un polinomio de grado 4, ya que fue el que mejor ajustó los nuevos datos a los datos originales. La figura 5 muestra el resultado al aplicar la regresión polinomial de grado 4.

4.1. Movimientos de cabeza

La figura 5 muestra el resultado de aplicar regresión polinomial de grado 4 a una señal. La imagen está dividida de la siguiente forma: a la izquierda (a) muestra la señal de entrada que fue obtenida por los momentos de Hu. En el centro (b), se muestra la curva obtenida al aplicar la regresión polinomial de grado 4. A la derecha (c), se muestra

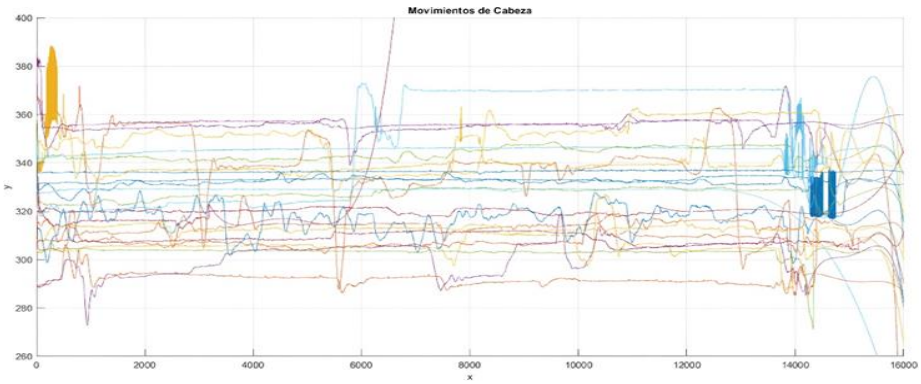


Fig. 6. Resultado al aplicar regresión polinomial.

la señal resultante de este proceso. Los nuevos datos están encerrados en el círculo verde. Como se puede ver en la figura 5, la regresión polinomial crea una curva siguiendo la forma de la señal, tratando de ajustar los picos para generar datos que tengan características similares.

Los datos resultantes crean una nueva señal (figura 8-c círculo verde), la cual es anexada a la señal original y de esta forma se amplifica dicha señal. Al aplicar este método, se ha logrado homogenizar las señales a un mismo tamaño, lo cual se puede ver en la figura 6. La figura 6 muestra el resultado al aplicar la regresión polinomial a todas las señales. Como se puede ver, todas las señales se han homogenizado a un mismo tamaño, dejando las señales listas para su análisis.

5. Conclusiones y trabajos futuros

Analizar los macro y micro movimientos de alguna parte del cuerpo que realiza una persona representa un problema difícil, ya que las personas pueden expresarse de diferente manera en las mismas situaciones. De esta forma, crear una señal a partir de ese movimiento puede ser complicado. Esto se debe a que el participante puede tener una reacción muy visible o no tener ninguna.

Para abordar este problema, se propone trasformar las micro y macro expresiones que realiza esa persona, centrándose en el movimiento de su cabeza cuando se le induce a experimentar una emoción específica. El algoritmo transforma el espacio visual del vídeo en una frecuencia en la que los movimientos se perciben como patrones y se utilizan para discretizar una señal.

Después, se aplica un método de predicción de datos para homogeneizar las señales a un mismo tamaño y así evitar un sobreajuste al momento de clasificar los datos. Los resultados indican que el algoritmo mapea los macro y micro movimientos que realiza la persona al ser estimulada; después, crea nuevos datos a parir del análisis de la señal y los agrega a la misma para tener que todas las señales tengan un mismo tamaño y así estar listas para su análisis.

5.1. Trabajos futuros

En trabajos futuros se pretende analizar el movimiento de más partes del cuerpo, además de estimular a los participantes con vídeos de diferentes emociones. Se recolectarán más muestras y se implementarán algoritmos de aprendizaje máquina para clasificar esas señales para encontrar diferentes emociones.

Referencias

1. Soriano-Méndez, J.J., Riaño-Gómez, D.A., Salazar-Morales, O.: Ratón USB para personas tetrapléjicas controlado con el movimiento de la cabeza. Ingeniería, vol. 19, no. 2 (2014). DOI: 10.14483/udistrital.jour.reving.2014.2.a02.
2. Gunes, H., Piccardi, M.: Bi-modal emotion recognition from expressive face and body gestures. Journal of Network and Computer Applications, vol. 30, no. 4, pp. 1334–1345 (2007). DOI: 10.1016/j.jnca.2006.09.007.
3. Zatarain-Cabada, R., Barrón-Estrada, M.L., Cárdenas-López, H.M.: Reconocimiento multimodal de emociones orientadas al aprendizaje. Research in Computing Science, vol. 148, no. 7, pp.153–165 (2019)
4. Maffei, A., Angrilli, A.: Spontaneous blink rate as an index of attention and emotion during film clips viewing. Physiology and Behavior, vol. 204, pp. 256–263 (2019). DOI: 10.1016/j.physbeh.2019.02.037.
5. Demiral, Ş.B., Kure-Liu, C., Benveniste, H., Tomasi, D., Volkow, N.D.: Activation of brain arousal networks coincident with eye blinks during resting state. Cerebral Cortex, vol. 33, no. 11, pp. 6792–6802 (2023). DOI: 10.1093/cercor/bhad001.
6. Glowinski, D., Dael, N., Camurri, A., Volpe, G., Mortillaro, M., Scherer, K.: Toward a minimal representation of affective gestures. IEEE Transactions on Affective Computing, vol. 2, no. 2, pp. 106–118 (2011). DOI: 10.1109/t-affc.2011.7.
7. Mahanama, B., Jayawardana, Y., Rengarajan, S., Jayawardena, G., Chukoskie, L., Snider, J., Jayarathna, S.: Eye movement and pupil measures: A review. Frontiers in Computer Science, vol. 3 (2022). DOI: 10.3389/fcomp.2021.733531.
8. Hoffmann, E.R.: A comparison of hand and foot movement times. Ergonomics, vol. 34, no. 4, pp. 397–406 (1991). DOI: 10.1080/00140139108967324.
9. Senecal, S., Cuel, L., Aristidou, A., Magnenat-Thalmann, N.: Continuous body emotion recognition system during theater performances. Computer Animation and Virtual Worlds, vol. 27, no. 3-4, pp. 311–320 (2016). DOI: 10.1002/cav.1714.
10. Michelini, Y., Acuña, I., Guzmán, J.I., Godoy, J.C.: LATEMO-E: A film database to elicit discrete emotions and evaluate emotional dimensions in Latin-Americans. Temas em Psicologia, vol. 27, no. 2, pp. 473–490 (2019). DOI: 10.9788/tp2019.2-13.
11. Bradski, G.: The OpenCV library. Dr. Dobb's Journal, vol. 25, no. 11, pp. 120–125, (2000).
12. Google AI for Developers: Guía de soluciones de MediaPipe (2024)
13. Maffei, A., Angrilli, A.: Spontaneous blink rate as an index of attention and emotion during film clips viewing. Physiology & Behavior, vol. 204, pp. 256–263 (2019). DOI: 10.1016/j.physbeh.2019.02.037.
14. Sharma, S., Choudhary, S., Kumar-Sharma, V., Goyal, A., Malik-Balihar, M.: Image watermarking in frequency domain using hu's invariant moments and firefly algorithm. In: International Journal of Image, Graphics and Signal Processing, vol. 14, no. 2, pp. 1–15 (2022). DOI: 10.5815/ijigsp.2022.02.01.
15. Chanchí-Golondrino, G.E., Campo-Muñoz, W.Y., Sierra-Martinez, L.M.: Aplicación de la regresión polinomial para la caracterización de la curva del COVID-19, mediante técnicas de machine learning. Investigación e Innovación en Ingenierías, vol. 8, no. 2, pp. 87–105 (2020)

ISSN 1870-4069

Algoritmo genético aplicado al alineamiento múltiple de secuencias genéticas

Luz Andrea Garcia Sena, David Israel Perez Valerio,
Adriana Berenice Maldonado Garcia, Ernesto Ríos-Willars

Universidad Autónoma de Coahuila,
Facultad de Sistemas,
Coahuila

{andrea_sena, davidvalerio, berenice.maldonado,
riose}@uadec.edu.mx

Resumen. El Problema del Alineamiento Múltiple de Secuencias es uno de los procesos fundamentales para la bioinformática. Este artículo aborda un análisis detallado sobre el funcionamiento de tres distintas versiones de un algoritmo aplicadas al Alineamiento múltiple de secuencias genéticas, dichas versiones tienen el objetivo de mejorar los alineamientos a partir de los procesos de mutación y reproducir de individuos con secuencias genéticas de la mundial base de datos NCBI. Inicialmente se dispone de una primera versión del algoritmo utilizada como base o punto de partida para la creación de sus dos variantes posteriores, cuyo propósito de estas dos versiones es optimizar la calidad de los resultados del algoritmo inicial estableciendo las modificaciones necesarias para mejorar la calidad de los individuos iniciales y la forma en la que se reproducen para crear nuevas generaciones de individuos más perfeccionadas y mejor alineadas. En este artículo se da a conocer la estructura y el funcionamiento de estas tres versiones con el propósito de realizar un análisis para obtener los resultados más destacables de acuerdo con las pruebas realizadas en cada versión.

Palabras clave: Alineamiento múltiple de secuencias, MSA, bioinformática, algoritmo genético.

A Genetic Algorithm for the Multiple Sequence Alignment Problem

Abstract. The Multiple Sequence Alignment Problem is one of the fundamental processes for bioinformatics. This article deals with a detailed analysis of the operation of three different versions of an algorithm applied to the Multiple Alignment of Genetic Sequences, these versions have the objective of improving the alignments from the mutation and reproduction processes of individuals with genetic sequences from the global NCBI database. Initially, a first version of the algorithm is available as a basis or starting point for the creation of its two subsequent variants, whose purpose of these two versions is to optimize the quality of the results of the initial algorithm by establishing the necessary modifications to improve the quality of the initial individuals and the way in which they reproduce to create new generations of individuals that are more

perfected and better aligned. This article presents the structure and operation of these three versions in order to carry out an analysis to obtain the most outstanding results according to the tests carried out in each version.

Keywords: Multiple sequence alignment, MSA, bioinformatics, genetic algorithm**.**

1. Introducción

A partir de finales de los años 80 en adelante, el término "bioinformática" ha sido mayormente empleado para describir métodos computacionales destinados al análisis comparativo de datos genómicos. No obstante, su definición original abarcaba un ámbito más amplio, siendo concebida como el estudio de los procesos informáticos dentro de sistemas bióticos [1]. En los organismos, el ADN despliega su función como el portador del material genético, transmitiendo así la información hereditaria de una generación a otra. Todos los seres vivos divergen a lo largo del tiempo a partir de un ancestro común, evolucionando mediante cambios en su ADN [2]. Por tanto, la capacidad de secuenciar el ADN de un organismo se define como un requisito esencial en la investigación biológica.

Diversas investigaciones han centrado su objetivo en desarrollar herramientas para llevar a cabo la secuenciación del ADN. Anteriormente, este proceso se limitaba a secuenciar unas pocas decenas o cientos de nucleótidos de forma simultánea. Sin embargo, en la actualidad, la secuenciación del ADN se realiza mediante máquinas de alto rendimiento que pueden secuenciar miles de millones de bases en un solo día. En el ámbito de la biología computacional, el alineamiento de secuencias de ADN, ARN o proteínas emerge como una tarea esencial y recurrente. Este proceso implica la comparación de un conjunto de secuencias para identificar las regiones donde coinciden y donde difieren entre sí.

El alineamiento múltiple de secuencias reviste una importancia fundamental en la bioinformática y la biología computacional, ya que proporciona perspectivas sobre la estructura y función de las biomoléculas [3]. El algoritmo genético representa un método de búsqueda heurística adaptativa, fundamentado en los principios de la genética de poblaciones. Su introducción se atribuye a John Holland en los inicios de la década de 1970. Este algoritmo se caracteriza por ser un enfoque de búsqueda *bioinspirado* a partir de la mecánica de la selección natural y los procesos genéticos [4]. Los algoritmos evolutivos se emplean para abordar problemas que carecen de una solución eficiente y bien definida.

Este enfoque se utiliza en la resolución de problemas de optimización y en modelado y simulación, donde se recurre a la aleatoriedad. Los Algoritmos Genéticos (GA) representan una solución para la población de candidatos (conocidos como individuos, organismos o genotipos) en problemas de optimización, avanzando hacia opciones más efectivas. Cada candidato presenta un conjunto de características (los genes o fenotipo) que pueden evolucionar y cambiar; la evolución comienza con una población de individuos aleatorios y se desarrolla de manera iterativa, considerando la población como base para cada reproducción.

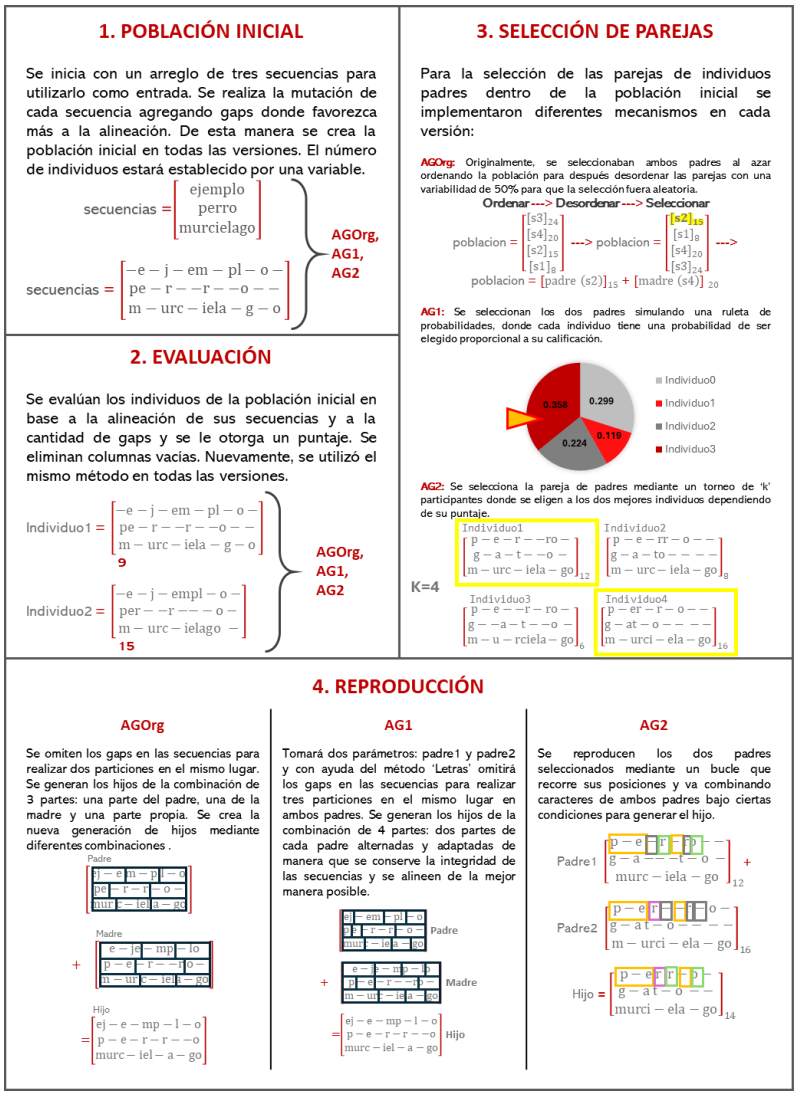


Fig. 1. Infografía comparativa sobre el funcionamiento de los mecanismos principales en las tres versiones.

En cada generación, se evalúa la aptitud (fitness) de todos los individuos, siendo esta aptitud usualmente el valor de la característica objetivo que se está optimizando. Cuando los individuos son lo suficientemente aptos, se seleccionan probabilísticamente de la población existente y sus genes se modifican para crear una nueva generación de individuos (recombinados y potencialmente mutados al azar). El AG continúa este proceso hasta que se ha alcanzado un número máximo de generaciones o de satisfacción [5]. Se destaca que lenguajes de alto nivel como Python o Perl suelen ser más eficientes en el procesamiento de listas o cadenas en comparación con C/C++/Java [4].

No obstante, el éxito práctico de estos algoritmos aún se encuentra en fase de investigación. La principal razón radica en la aleatoriedad inherente a estos algoritmos en el proceso de toma de decisiones.

Esta aleatoriedad introduce una complejidad adicional tanto en la ejecución como en la comprensión del proceso durante la aplicación de los algoritmos [6].

2. Planteamiento del problema

Cuando hay dos secuencias que pueden acomodar un número limitado de inserciones de huecos, el número de alineaciones aumenta a medida que aumenta la longitud de las secuencias. El número de alineaciones potenciales para un par de secuencias m y n, con 'k' inserciones (m,n) [7], se puede calcular usando la ecuación (1):

$$f = \sum_{k=0}^{\min(m,n)} \frac{(m+n-k)!}{k!\,(m-k)!\,(n-k)!} \,. \tag{1}$$

Cuando se intenta resolver el problema de alineamiento múltiple de secuencias, utilizando un enfoque de fuerza bruta, el problema se vuelve NP-completo. Por el contrario, la programación dinámica tiene una complejidad de O(LN), donde L es la longitud de la secuencia y N es el número de secuencias [8]. Los investigadores tienen como objetivo mejorar la eficiencia de este problema a través de enfoques heurísticos y metaheurísticos e implementaciones paralelas para reducir los costos computacionales [9].

Se requiere el desarrollo de un algoritmo de alineamiento múltiple de secuencias genéticas con el propósito de procesar dicho alineamiento, utilizando los métodos más idóneos, secuencias de caracteres, principalmente provenientes de paquetes de software diseñados para el alineamiento de secuencias de ADN y proteínas en formato FASTA. Este algoritmo debe ser capaz de operar eficientemente con cadenas de caracteres de cualquier longitud.

3. Materiales y métodos

Para el desarrollo del algoritmo original (AGOrg) realizado en el lenguaje de programación Java, así como la segunda versión (AG1), en cambio, para la segunda variante optimizada (AG2), se hizo una migración al lenguaje de programación Python, considerando que esto podría contribuir a su optimización.

Para todos los procesamientos se comparan las métricas de *tiempo* y *fitness* alcanzado por el algoritmo. Los tres algoritmos genéticos se publican en la plataforma GitHub para futuras referencias en el link github.com/luzAndreaGs/Algoritmosde AlineamientoMultipledeSecuencias. Los experimentos necesarios se llevaron a cabo utilizando un equipo portátil con un procesador Ryzen 5, una tarjeta gráfica AMD Radeon, un disco de estado sólido de 256GB y 12GB de memoria RAM. Para la realización de estos mismos experimentos, se emplearon nueve archivos de secuencias FASTA de secuencias genéticas de ADN como entrada, obtenidas de la base de datos NCBI [10]:

Tabla 1. Promedios de los resultados de fitness y tiempo para cada algoritmo en cada grupo de secuencias.

Grupo	Fitness			Tiempo		
	AGOrg	**AG1**	**AG2**	**AGOrg**	**AG1**	**AG2**
C1	8480.4	8608.8	8609.8	14	14	11.9
C2	1402	1404.6	1407	4.1857	3.8158	2.8608
C3	50268.43	50203.4	50304.13	224.08	149.89	83.02684

- Bacterial Escherichia coli strain con 2297 bases de longitud.
- Bacterial Enterococcus faecalis strain con 2805 bases de longitud.
- Bacterial Porphyromonas gingivalis strain con 1196 bases de longitud.
- Bacterial Bifidobacterium longum con 590 bases de longitud.
- Bacterial Helicobacter pylori isolate con 315 bases de longitud.
- Bacterial Staphylococcus aureus con 305 bases de longitud.
- Gen Humano MAPK3 con 9116 bases de longitud.
- Gen Humano MAPK13 con 14009 bases de longitud.
- Gen Humano PRKACA con 26075 bases de longitud.

Las secuencias se organizaron en tres grupos para realizar las pruebas del funcionamiento de los algoritmos: el primer grupo está compuesto por las secuencias bacteriales Escherichia coli strain, Enterococcus faecalis strain y Porphyromonas gingivalis strain; el segundo grupo se compone de las secuencias bacteriales Bifidobacterium longum, Helicobacter pylori isolate y Staphylococcus aureus; finalmente el tercer grupo de secuencias se compone de genes humanos MAPK3, MAPK13 y PRKACA.

4. Algoritmos propuestos

Se describen los algoritmos propuestos con los que se realizó una comparativa de desempeño para el alineamiento de las secuencias descritas. **Versión inicial del algoritmo (AGOrg):** El algoritmo se compone de cuatro clases: “Individuo”, “Generación”, “Archivos” y “Algoritmo”. Aquí la explicación del trabajo que tienen cada una de las clases y sus respectivos métodos:

Individuo. Hace el trabajo completo para manipular y mutar un arreglo de secuencias de caracteres (que convierte en una matriz) para alinearlas de manera que tengan la misma longitud y estén ordenadas en columnas.

- *Calificar().* Asigna puntuaciones para las secuencias basándose en su alineación de las letras y la presencia de gaps.
- *Mutar().* Realiza la mutación de las secuencias: primero, busca si hay gaps en cada secuencia y determina qué gaps eliminar y cuáles agregar.
- *ModificarGaps().* Realiza dichas modificaciones.
- *Alinear().* Alinea las secuencias de manera que todas tengan la misma longitud, esto lo hace agregando gaps al final de cada secuencia de ser necesario.

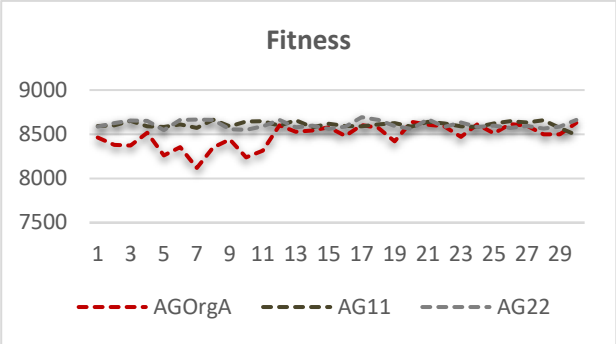


Fig. 2. Gráfica que representa y compara el puntaje del fitness de cada corrida en cada una de las versiones del primer grupo de secuencias.

– *Eliminar().* Después de haber mutado el arreglo de secuencias elimina las columnas que solo contienen gaps.

Generación. Se encarga de manejar las generaciones de individuos, donde cada uno de ellos representa una posible solución al problema de alineamiento de secuencias. La clase consta de una variable (tipo int) que actualmente está inicializada con el número 4, representa el número de individuos por generación y se puede modificar por cualquier número múltiplo de 4 debido a que cada pareja tiene 4 hijos.

– *Ordenar().* Acomoda la población de individuos de acuerdo con su puntuación (de mayor a menor) utilizando el algoritmo básico de burbuja.
– *DesordenarParejas().* Desacomoda aleatoriamente las parejas de individuos para crear una variabilidad del 0.50 al momento de seleccionar la pareja de padres.
– *Letras().* Cuenta únicamente las letras (ignorando los gaps) de cada una de las secuencias, itera sobre cada una de ellas para posteriormente realizar las particiones en la misma letra de cada secuencia en ambos padres.
– *Hijo().* Crea un nuevo individuo (hijo) a partir de dos individuos padres con dos particiones en cada uno (se dividen en 3 secciones) en la misma posición con ayuda del método Letras(). El hijo hereda y combina dos secciones de las secuencias de los padres (una de cada uno) y de acuerdo con ellas adapta una tercera parte que es propia de el mismo hijo.
– *Reproducir().* Ordena la población de individuos, luego desordena las parejas y finalmente reproduce una nueva generación de individuos a partir de los padres de la generación actual.

Archivos. Esta clase se encarga de leer archivos de texto. Utiliza un objeto de Scanner para leer el archivo línea por línea, omite la primera línea, luego lee el resto del archivo y concatena cada línea en la cadena txt. Si ocurre una excepción *FileNotFoundException*, el método imprime un mensaje de error.

Algoritmo. Es la clase que contiene el método principal (main) para ejecutar el algoritmo de ordenamiento de secuencias. Comienza instanciando un objeto de la clase *Archivos* o llamando a un arreglo de Strings llamado secuencias que contiene las secuencias de caracteres que se utilizaran como entrada del algoritmo; en caso de instanciar archivos, se declaran variables con las rutas de los archivos que se van a utilizar en el arreglo de secuencias. Instancia la clase *Generación* para crear un objeto de esta clase pasando el arreglo como argumento del constructor, esto inicializa la primera generación de individuos con las secuencias dadas.

Llama al método *Ordenar* con el objeto de *Generación* creado para ordenar la población inicial de individuos. La muestra en pantalla previamente ordenada y separando por un espacio los caracteres de las secuencias de cada individuo. Después hay un bucle que se ejecuta 10 veces. En cada iteración del bucle se lleva a cabo la reproducción (método *Reproducir*) del objeto de *Generación*, seguido del método para ordenar la nueva población.

Finalmente se imprime la población actual de individuos en cada bucle de reproducción con el respectivo puntaje de cada uno. De esta manera trabajan en conjunto las cuatro clases componentes del algoritmo original. Dentro de las clases, se detectó que el algoritmo únicamente hace su trabajo completo para un arreglo de tres secuencias, al querer utilizar más de tres secuencias no realiza la reproducción debido a un error en el método implementado, esto nos llevó a la conclusión de que sería un buen punto de mejora. Esta versión inicial fue hecha por una alumna de noveno grado (actualmente) de la Facultad de Sistemas de la Universidad Autónoma de Coahuila quien autorizó compartir el código alumnos de un grado inferior y utilizarlo para realizar las mejoras posteriores.

4.1 Segunda versión del algoritmo (Ag1)

En esta versión de la optimización del algoritmo, se trabajaron principalmente los siguientes puntos:

1. Correcto funcionamiento de la reproducción considerando cualquier cantidad de secuencias en los individuos.
2. Mejorar la calidad en la selección de los individuos padres.
3. Realizar más particiones en los individuos padres.
4. Implementar un nuevo método para asegurar que los individuos hijos mantienen íntegras sus secuencias.

Para trabajar la mayoría de estos objetivos, se modificó de manera considerable la clase “Generación” debido a que en ella se encuentran los métodos para la reproducción. A continuación, se explican los métodos que sufrieron cambios.

Constructor. El constructor declara una matriz *x* de caracteres con una longitud igual a la cantidad de elementos en el array de entrada (*entrada.length*), la matriz tendrá el mismo número de filas que la longitud del array de entrada. Itera sobre cada cadena en el array de entrada,cada cadena, se utiliza el método *toCharArray()* para convertirla en un array de caracteres. El array de caracteres resultante se asigna a la fila correspondiente en la matriz *x*. Se itera sobre cada índice *i* desde 0 hasta *num - 1*, donde *num* es el número de individuos en la población (anteriormente definido con el valor de

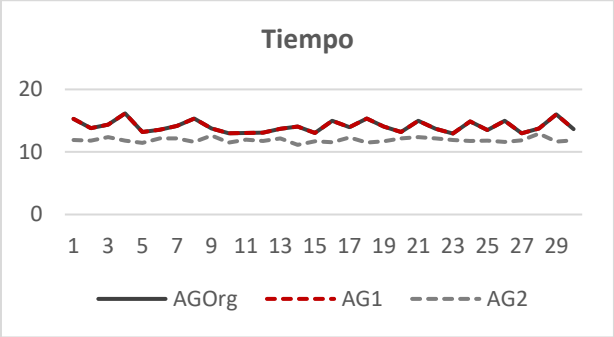


Fig. 3. Gráfica que representa y compara el tiempo de cada corrida en cada una de las versiones del primer grupo de secuencias.

4). Para cada índice *i*, se crea un nuevo objeto de la clase Individuo, pasando la matriz *x* como argumento al constructor de Individuo.

Esto crea un nuevo individuo en la población inicial utilizando las secuencias de caracteres de la matriz *x*. No hay una gran diferencia con la versión original, en resumen, el constructor ahora solo crea la matriz una sola vez y la utiliza repetidas ocasiones para crear los individuos de la población inicial a diferencia de la versión original, que crea la matriz repetidas veces a la par de los individuos dentro del bucle, lo que podría resultar un poco confuso al analizar el código.

SeleccionPorRuleta. Se calcula la sumatotal de las calificaciones de todos los individuos y en base a ella las probabilidades de cada individuo de ser seleccionado, esto se hace iterando sobre los individuos dividiendo su calificación entre la suma total. Se selecciona un numero aleatorio para la reproducción con ayuda de la función random.nextDouble() de la clase "Random" de java, que selecciona un valor aleatorio en el rango [0,1) es decir, que incluye el 0 pero excluye el 1. Se acumulan las probabilidades de selección de cada individuo en un valor acumulado. A medida que se recorre la lista de individuos, se compara el valor aleatorio generado con las probabilidades acumuladas.

Cuando el valor aleatorio es menor o igual que una de las probabilidades acumuladas, se selecciona el individuo correspondiente. Si ningún individuo cumple esta condición (por ejemplo, si el valor aleatorio es mayor que todas las probabilidades acumuladas), se selecciona un individuo aleatorio de la población.

Estas modificaciones hacen más certera la probabilidad de selección de un individuo sea proporcional a su calificación, lo que significa que los individuos con mejores calificaciones tendrán una mayor probabilidad de ser seleccionados lo que dará como resultado individuos hijos de mejor calidad.

Hijo. Toma dos variables como parámetros: *padre1* y *padre2*. Se inicializan las variables *Letras()* para contar el número de letras en cada parte, *n* para no perder el índice en el arreglo de las secuencias del hijo y *secHijo* para almacenar las partes de la secuencia del hijo (su tamaño depende de la cantidad de secuencias que hay en los individuos). Se declara un arreglo de tres particiones para definir los límites de las cuatro partes en las que estarán divididas las secuencias, todas tendrán los límites en la

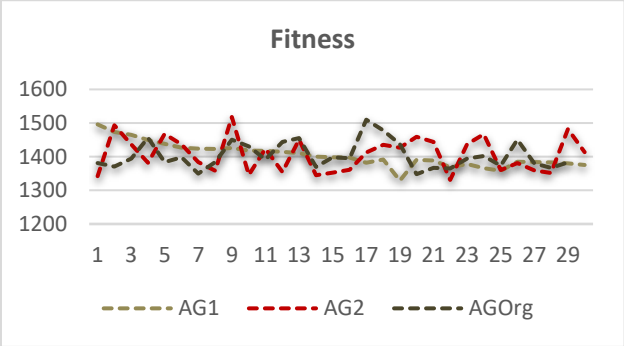


Fig. 4. Gráfica que representa y compara el puntaje del fitness de cada corrida en cada una de las versiones del segundo grupo de secuencias.

misma letra o el mismo lugar gracias al método *Letras()* como en la versión original del algoritmo.

Se inicia un bucle que itera sobre cada secuencia de los individuos padres considerando cualquier cantidad de secuencias y ya no únicamente tres para posteriormente recorrer las cuatro partes de cada secuencia y combinarlas de manera que el hijo hereda la primera y tercera parte del hijo del *padre1* y la segunda y cuarta parte del *padre2* de acuerdo con los límites establecidos. Se guarda la secuencia del hijo en el arreglo *secHijo*. Se convierte el arreglo *secHijo* en un arreglo de caracteres y. Se crea y devuelve un nuevo objeto Individuo con las secuencias del hijo. Este método divide las secuencias de los individuos padres en cuatro partes y las combina para formar la secuencia del hijo. Recorre las partes de todas las secuencias de los padres de manera simultánea

Reproducir. Se crea un nuevo arreglo de individuos llamado hijos con la misma longitud que la población actual (*num*). Se inicia un bucle for que se ejecutará *num* veces, donde i representa el índice de cada hijo que se va a crear. Dentro del bucle, se seleccionan aleatoriamente dos individuos padres llamando al método s*eleccionPorRuleta()*. Elige un individuo de la población actual donde la probabilidad de selección está proporcionalmente relacionada con la calificación de cada individuo.

Es por ello que ya no hace falta ordenar la población para seleccionar la pareja de padres. Se llama al método *Hijo()* para crear un nuevo individuo hijo a partir de los dos individuos padres seleccionados. El individuo hijo resultante se agrega al arreglo de hijos en la posición i. Una vez que se han creado todos los hijos, el arreglo de la población actual (población) se actualiza con el arreglo de hijos, lo que reemplaza la población anterior con la nueva generación de individuos.

verificarSecuencia. Se crea una variable temporal *sb* para modificar las secuencias de los individuos sin necesidad de crear otro objeto que construye las secuencias de los hijos sin los gaps. Itera sobre cada secuencia en el arreglo de los individuos hijos y dentro de este bucle, hay otro que itera sobre cada caraacter y mientras no sean gaps, se agregan a la variable temporal. Esta variable se convierte a una cadena de caracteres llamada *secuenciaHijo* y se inicializa una variable booleana integridad como true, que indicará si la secuencia del hijo coincide con las secuencias de la población.

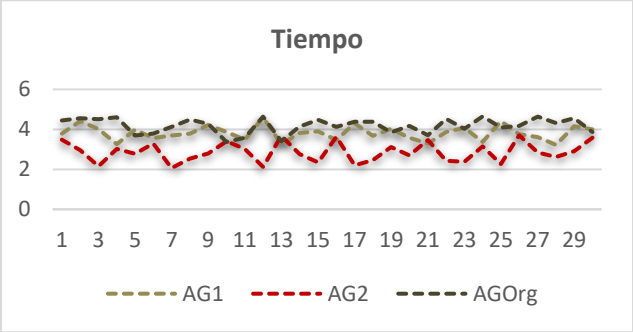


Fig. 5. Gráfica que representa y compara el tiempo de cada corrida en cada una de las versiones del segundo grupo de secuencias.

Se crea otra variable temporal s*ecuenciaOriginal* que se construye de la misma manera que la anterior pero ahora utilizando un individuo de la población inicial. Se comparan ambas variables y si son iguales, la variable integridad permanece como true, si no son iguales, la variable integridad se establece en false. Se devuelve un mensaje indicando si la verificación fue exitosa o si se encontró un error en la secuencia del hijo. Este método se manda a llamar en la clase principal para mostrar el mensaje correspondiente a cada individuo hijo debajo de su calificación.

4.2 Tercera versión del algoritmo (AG2)

Como propuesta de mejora en esta versión, surgió la idea de que el algoritmo se transportara a otro lenguaje de programación (Python) para conocer cómo sería su funcionamiento y realizar cambios para la optimización del algoritmo. Naturalmente se sabe que Python tiende a ser un lenguaje conocido por su facilidad de uso y su rápida iteración en el desarrollo, lo que puede favorecer a la diferencia de rendimiento.

Funciona igualmente con un arreglo de secuencias de caracteres, pueden ser palabras o archivos de texto. Primero almacena las secuencias en una lista de listas mediante el constructor de la clase ‘Individuo’ y procede a aplicar las mutaciones a cada secuencia, éste y los procesos de modificación de gaps, alineación y la asignación del puntaje se hacen de la misma manera que en el algoritmo original, siguiendo las mismas reglas para la calificación. En la clase *Generacion* se realizaron los cambios más significativos, a continuación se explican parte por parte:

Constructor. En él se establece un valor para el número de individuos que habrá en la población esto permite que se adapte a ‘n’ cantidad de secuencias, se crea la población inicial con una lista tomando de base las secuencias existentes de la siguiente manera: por cada índice del número de secuencias se crea un objeto de la clase ‘Individuo’ y se almacenan en la lista de población, además en el constructor se agregó un atributo que guarda las secuencias originales.

alinear_secuencias(). Se añadió este nuevo método que se encarga de mantener todas las secuencias con la misma longitud agregando gaps al final de las palabras que lo requieran dentro de la población inicial con el objetivo de facilitar sus procesos de comparación y reproducción.

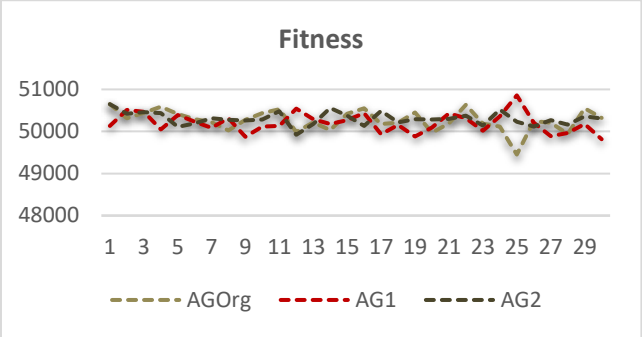


Fig. 6. Gráfica que representa y compara el puntaje del fitness de cada corrida en cada una de las versiones del tercer grupo de secuencias.

selección_torneo(). La selección de los padres ya no se hace de manera aleatoria, sino que ahora los padres se eligen con ayuda este método que selecciona un grupo de 'n' de individuos de la población inicial y los devuelve como participantes del torneo donde los que tienen mayor puntaje, tienen más posibilidades de ganar.

reproducción(). Primero se alinean las secuencias de la población, se selecciona una pareja de padres entre los 4 individuos de la población inicial seleccionando uno por uno, excluyendo el que ya fue seleccionado, genera cuatro hijos de estos padres en dos bucles de dos hijos cada uno donde el hijo1 toma los parámetros de los padres en el mismo orden que se eligieron y el hijo2 de forma invertida, se cambian los roles para introducir variabilidad en la descendencia y evitar sesgos o patrones que puedan surgir si siempre se utilizan los mismos padres en el mismo orden; la nueva población es la de los hijos.

hijo(). Sigue la lógica para reproducir los nuevos individuos: alinea los dos padres agregando gaps al final de cada secuencia de manera que ambos tengan la misma longitud, después va comparando los padres carácter por carácter para decidir de qué manera se van a estructurar a los hijos en base a algunas condiciones: a) Si ambos padres tienen el mismo carácter o también si el carácter del padre1 es un gap y en la misma posición el carácter del padre2 es alfabético, el hijo hereda el carácter del padre2 y b) En otro caso, el hijo hereda el carácter del padre1.

validar_secuencias(). Verifica que no haya modificaciones en las secuencias de los hijos, toma estas secuencias (*hijos*), las convierte en secuencias de caracteres alfabéticos únicamente (es decir, elimina sus gaps) y va comparando los elementos o secuencias de cada lista con las originales, utiliza una variable booleana que ayuda a tomar una decisión: si encuentra alguna discrepancia en ellas (False) quiere decir que las secuencias se están alterando al reproducirse, de lo contario, se mantienen integras (True). Por último, se manda a llamar este método en la clase principal 'Algoritmo' con el objeto de la clase *Generacion* dentro del bucle de reproducción tomando los elementos de las secuencias de la nueva población y comparándolas con las secuencias originales, imprime debajo de cada individuo hijo el resultado de su verificación: *True = "Individuo verificado correctamente", False = "Individuo no válido"*. La figura 1 muestra el proceso de cuatro fases para los tres algoritmos desarrollados.

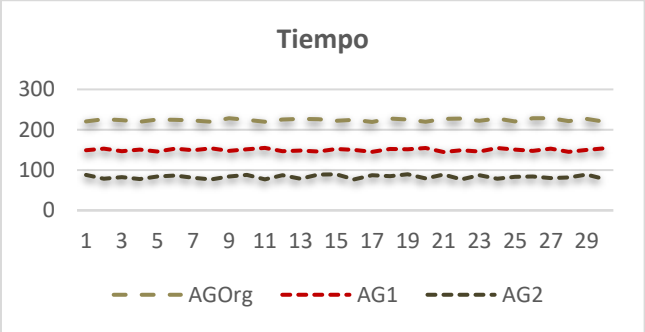


Fig. 7. Gráfica que representa y compara el tiempo de cada corrida en cada una de las versiones del tercer grupo de secuencias.

5. Experimento

Para realizar la comparación entre el funcionamiento de las tres versiones del algoritmo, se ejecutó un experimento en el cual se efectúan treinta corridas de cada uno de los algoritmos bajo los mismos parámetros en todas las versiones para que los resultados sean viables estadísticamente para comparativa. La población inicial es de 52 individuos (se debía utilizar una cifra que sea múltiplo de 4 ya que en la versión original se crean 4 individuos hijos por generación), un número máximo de gaps posibles a añadir de 50 para el proceso de mutación, así mismo 10 generaciones de reproducción. En esas treinta corridas se pretende evaluar el tiempo total de ejecución, el fitness de cada una y el promedio de los resultados (de fitness y tiempo en segundos) de cada versión.

6. Resultados

Se realizaron cálculos de promedios para los resultados de fitness y tiempo como comparativa entre los tres algoritmos en el proceso de los tres conjuntos de secuencias como se indica en la tabla 1. Así mismo se realizaron pruebas Anova de un factor para identificar diferencias significativas entre dichos resultados con el uso del software Minitab16. S describen estos resultados en gráficas comparativas de la Fig. 2 y Fig. 3. Al analizar los resultados obtenidos, podemos ver que efectivamente según las estadísticas los resultados de ambas versiones optimizadas (AG1 y AG2) mejoraron considerablemente en cuanto a puntaje y tiempo en comparación con la versión original, esto se describe en figuras 2 y 3. Respecto del tiempo, la prueba Anova indicó un valor F=23.97 con valor P=0.00 en la comparativa entre los tres algoritmos. Respecto del fitness, la prueba Anova indicó un valor F=23.97 y valor P=0.000 en la comparativa de los tres algoritmos. Por lo que aceptamos que existen diferencias con un nivel de confianza de 95.

Las figuras 4 y 5 representan los resultados de tiempo y fitness para el segundo conjunto. Respecto del tiempo en el resultado de los tres algoritmos, la prueba Anova

indicó un valor F= 78.57 y valor P=0.000 con nivel de confianza de 95, por lo que asumimos que existen diferencias. En la comparativa del fitness para los tres algoritmos, la prueba Anova indicó un valor F= 0.27 y valor P=0.763 con el mismo nivel de confianza. Por lo que asumimos que no existen diferencias.

Las figuras 6 y 7 representan los resultados de fitness y tiempo para los tres algoritmos en el tercer grupo de secuencias. Respecto del tiempo, en la comparativa de los tres algoritmos la prueba Anova indicó un valor F= 11052.68 y valor P=0.000 con nivel de confianza de 95. Por lo que asumimos diferencias significativas. Respecto del fitness, la prueba Anova indicó un valor F= 1.62 y valor P=0.203 con nivel de confianza de 95. Por lo que asumimos que no existen diferencias.

7. Conclusiones

Respecto de la métrica fitness, de acuerdo los resultados obtenidos y con los gráficos, podemos aseverar que, estadísticamente las versiones optimizadas (AG1 y AG2) mejoraron los resultados de manera exitosa en comparación con los que se obtuvieron en el algoritmo original (AGOrg) y de hecho, en ambas optimizaciones los resultaos fueron bastante similares aun estando implementados en diferentes lenguajes de programación (Java y Python), lo que quiere decir que las estrategias de selección y reproducción cumplieron con su propósito de manera exitosa aumentando la calidad de individuos en las nuevas generaciones. Con el tiempo de ejecución, ocurren cosas interesantes.

Podemos observar que el tiempo promedio de AGOrg y AG1 es casi idéntico y dichas versiones fueron desarrolladas en Java, se sabe que comparado con otros lenguajes de programación, Java es un poco más lento, concluyendo con que a pesar de ser una versión mejorada y que nos arroja mejores resultados en las generaciones (fitness), AG1 no presentó alguna ventaja en el tiempo de ejecución del algoritmo, en cambio AG2 si demostró tener ventaja si valoramos también el tiempo, esto puede atribuirse a que está desarrollado en Phyton y actualmente, Python está entre los lenguajes de programación más rápidos, superando a casi todos los demás en el mundo de la programación.

En resumen, podemos decir que definitivamente el rendimiento de ambas versiones AG1 y AG2 alcanzaron las expectativas deseadas y superaron los estándares que se obtuvieron con AGOrg, sin embargo, AG2 está por encima de AG1 ya que pese a la aleatoriedad que siempre existe en este tipo de algoritmos de ordenamiento múltiple de secuencias, éste logró perfeccionar los resultados de manera muy similar o un poco mejor, y además en menor tiempo. Como trabajo a futuro se visualiza la implementación de nuevas estrategias en el proceso de mutación para los diferentes algoritmos aquí planteados, así como incrementar el número de secuencias genéticas en conjuntos de prueba que permitan una comparativa más amplia y estadísticamente más rigurosa. También se visualiza la incorporación de nuevos recursos de hardware que permitan la experimentación con mayores niveles de complejidad para el algoritmo genético.

Agradecimientos. Los autores desean reconocer a la Facultad de Sistemas de la Universidad Autónoma de Coahuila por el apoyo en la realización de este trabajo.

Referencias

1. Hogeweg, P.: Las raíces de la bioinformática en la biología teórica. PLoS Computational Biologics, pp. 7 (2011)
2. Chowdhury, B., Garai, G.: A Review on Multiple Sequence Alignment from the Perspective of Genetic Algorithm. Genomics, vol. 109, no. 5–6, pp. 419–431 (2017) DOI: 10.1016/ j.ygeno.2017.06.007.
3. Kaya, M., Sarhan, A., Alhajj, R.: Multiple Sequence Alignment with Affine Gap by Using Multi-objective Genetic Algorithm. Computer Methods and Programs in Biomedicine, vol. 114, no. 1, pp. 38–49 (2014) DOI: 10.1016/j.cmpb.2014.01.013.
4. Kumar, M., Husian, M., Upreti, N., Gupta, D.: Genetic Algorithm: Review and Application. Electrical Engineering eJournal, pp. 2–5 (2010)
5. Alam, T., Qamar, S., Dixit, A., Benaida, M.: Genetic Algorithm: Reviews, Implementations, and Applications. CompSciRN: Computer Principles, pp. 1–9 (2020). DOI: 10.48550/arXiv.2007.12673.
6. Hanif, M.K., Talib, R., Awais, M., Saeed, M.Y., Sarwar, U.: Comparison of Bioinspired Computation and Optimization Techniques. Current Science, vol. 115, no. 3, pp. 450–453 (2018)
7. Waterman, M.S.: Introduction to Computational Biology: Maps, Sequences and Genomes. Chapman and Hall/CRC (2018)
8. Steiner, G.: On the Complexity of Dynamic Programming for Sequencing Problems with Precedence Constraints. Annals of Operations Research, vol. 26, no. 1–4, pp. 103–123 (1990). DOI: 10.1007/bf02248587.
9. Almanza-Ruiz, S.H., Chavoya, A., Duran-Limon, H.A.: Parallel Protein Multiple Sequence Alignment Approaches: A Systematic Literature Review. The Journal of Supercomputing, vol. 79, no. 2, pp. 1201–1234 (2022). DOI: 10.1007/s11227-022-04697-9.
10. Sayers, E.W., Bolton, E.E., Brister, J.R., Canese, K., Chan, J., Comeau, D.C., Connor, R., Funk, K., Kelly, C., Kim, S., Madej, T., Marchler-Bauer, A., Lanczycki, C., Lathrop, S., Lu, Z., Thibaud-Nissen, F., Murphy, T., Phan, L., Skripchenko, Y., Tse, T., Wang, J., Williams, R., Trawick, B.W., Pruitt, K.D., Sherry, S.T.: Database Resources of the National Center for Biotechnology Information. Nucleic Acids Research, vol. 50, no. D1, pp. D20–D26 (2021). DOI: 10.1093/nar/gkab1112.

ISSN 1870-4069

Un breve resumen sobre la implementación de los sistemas expertos en problemas de agricultura

Martín Laguna Estrada[1], Norma Verónica Ramírez Pérez[1], Jessica Alejandra Araujo Rodríguez[1], Norma Natalia Rubín Ramírez[2]

[1] Tecnológico Nacional de México, Guanajuato, México

[2] Tecnológico Nacional de México, Nayarit, México

{martin.laguna,norma.ramirez,d2203008}@itcelaya.edu.mx, nrubin@iitepic.edu.mx

Resumen. Los Sistemas Expertos (SE) en la agricultura, son programas informáticos diseñados para simular el conocimiento y la experiencia de expertos humanos en el ámbito agrícola. Estos sistemas utilizan bases de datos y algoritmos para toma de decisiones y ofrecen recomendaciones sobre diversas actividades agrícolas como los son: el diagnóstico de enfermedades de los cultivos, la gestión de plagas y el manejo de la producción entre otros. Su aplicación abarca desde la optimización de recursos hasta la predicción de condiciones climáticas adversas, de ahí que los SE en agricultura sean conocidos por su contribución a mejorar la eficiencia, la productividad y la sostenibilidad de las operaciones agrícolas al proporcionar soluciones rápidas y precisas a problemas complejos. Un plus que proporcionan los SE es que, además de facilitar la toma de decisiones basadas en datos y conocimientos especializados, resultan fundamentales para enfrentar los desafíos actuales del sector agrícola como la seguridad alimentaria y la gestión ambiental. El objetivo de este artículo es presentar una revisión literaria de los temas actualizados de agricultura que se pueden abordar con los Sistemas Expertos en los últimos años.

Palabras clave: Sistemas expertos, operaciones agrícolas, condiciones edafológicas, algoritmos de aprendizaje, inteligencia artificial.

A Brief Summary on the Implementation of Expert Systems in Agricultural Problems

Abstract. Expert Systems (SE) in agriculture are computer programs designed to simulate the knowledge and experience of human experts in the agricultural field. These systems use databases and algorithms for decision making and offer recommendations on various agricultural activities such as: diagnosis of crop diseases, pest management and production management, among others. Their application ranges from resource optimization to the prediction of adverse weather conditions, hence ES in agriculture are known for their contribution to improving the efficiency, productivity and sustainability of agricultural

operations by providing quick and accurate solutions to problems. complex. A plus that ES provides is that, in addition to facilitating decision-making based on data and specialized knowledge, they are essential to face the current challenges of the agricultural sector such as food security and environmental management. The objective of this article is to present a literary review of the updated agricultural topics that can be addressed with Expert Systems in recent years.

Keywords: Expert systems, agricultural operations, soil conditions, learning algorithms, artificial intelligence.

1. Introducción

En la actualidad los agricultores han deseado maximizar los rendimientos de sus cultivos que a la vez permita dar respuesta a soluciones que se presentan día a día, sobre todo en las limitaciones de terreno, insumos y recursos económicos. La tecnología ha llegado a resolver ciertas situaciones que muchos de los agricultores no se imaginaban que se podían realizar, como, por ejemplo, las mediciones, el análisis y la respuesta a variaciones que se dan en el cultivo, entre otras. Los agricultores requieren conocimientos avanzados o la asesoría de expertos para tomar decisiones durante todo el proceso de siembra.

Requieren conocimientos y toma de decisiones desde el proceso de preparación, selección de semillas, fertilizantes, pesticidas, programación del agua, gestión de malezas, cultivo, recolección y distribución de sus productos y otras variables para obtener un alto rendimiento. Los SE han sido una solución a esta problemática y pueden, además, estar diseñados para ayudar al agricultor a tomar decisiones económicamente viables y sólidas en el manejo de cultivos de inicio a fin.

2. Materiales y métodos

A continuación, se describen aplicaciones recientes de diferentes Sistemas Expertos en la agricultura descritos por autores especialistas en el área, señalando las ventajas que implica el utilizar estos sistemas.

2.1. Sistemas expertos en la agricultura

A finales de la década de los 70´s, los Sistemas Expertos comienzan a aplicarse en el ámbito agrícola y después de casi 30 años de desarrollo, su dominio de aplicación se ha extendido en cada uno de los procesos agrícolas para la toma de decisiones que impactan en la parte económica. Los SE son una de las ramas importantes de la Inteligencia Artificial orientada a aplicaciones, el enfoque de los Sistemas Expertos es intentar modelar el conocimiento del dominio de los expertos en sus respectivas áreas de especialización, por ejemplo, diagnóstico, planificación, previsión, etc., y se basan en el conocimiento que incluye no solo modelos y datos, sino que también enfatiza las experiencias de los expertos en el dominio. Un sistema experto es una aplicación

informática que resuelve problemas complicados que de otro modo requerirían una amplia experiencia humana.

Puede ser operado por una persona con menos educación o un lego en un campo de conocimiento particular. Los Sistemas Expertos Agrícolas (SEA) son programas informáticos que utilizan inteligencia artificial para emular el conocimiento agrícola humano. Utilizan datos, reglas lógicas y algoritmos para tomar decisiones en la producción y manejo de cultivos. Al replicar el razonamiento humano, mejoran la toma de decisiones, esto es, transfieren el saber humano a la computadora, resolviendo problemas de manera efectiva y pueden interactuar con usuarios expertos y no expertos, facilitando la resolución, en este caso, de problemas agrícolas. Estas herramientas optimizan la eficiencia agrícola y mejoran la calidad y productividad, reduciendo el impacto ambiental, además permiten pronosticar la producción y los precios del mercado, brindando a los agricultores acceso a información valiosa [1,2].

2.2. Toma de decisiones integradas para el manejo de cultivos

Los Sistemas Expertos Agrícolas (SEA) permiten a los agricultores tomar decisiones fundamentadas sobre siembra, riego, fertilización y control de plagas, considerando factores como clima, suelo y estado del cultivo. Ejemplo de ello es el "Sistema de Análisis de Datos Agrícolas" desarrollado por [3], que unifica servicios de socialización e informatización de operaciones para mejorar la eficiencia en la toma de decisiones. Por su parte, [4], resumen que la integración del aprendizaje automático ha mejorado la eficiencia agrícola, pero se enfrenta a desafíos como la adaptabilidad de los modelos y la seguridad de los datos. Algoritmos de aprendizaje supervisado y no supervisado, incluido el aprendizaje automático, identifican patrones y predicen rendimientos agrícolas, como señalan [5, 6], estos últimos destacan la aplicación del aprendizaje automático en diversos aspectos agrícolas, con resultados significativos.

2.3. Identificación y control de plagas y enfermedades

Los Sistemas Expertos Agrícolas (SEA) se emplean extensamente en la detección automática de plagas y enfermedades en cultivos, utilizando inteligencia artificial y tecnologías como Data Mining, Neural Networks, Expert System y Machine Learning. Este enfoque reduce el costo de los químicos aplicados, mejorando la economía agrícola y permitiendo un aumento sostenible de la producción.

En 2012 [7], desarrollaron un sistema experto para diagnosticar plagas y enfermedades en cultivos de berenjena en la región Caribe de Colombia. Utilizando Swi-Prolog, Java, XML y PostgreSQL, el sistema pudo identificar ocho plagas y nueve enfermedades considerando daños en la planta y presencia de insectos, lo que requirió de datos climáticos para mayor precisión.

Con la utilización de SWI-Prolog, [8] en (2020), presentaron un sistema experto para el diagnóstico de enfermedades y plagas en los cultivos de arroz, tabaco, tomate, pimientos, maíz, pepino y frijol. El sistema brindó diagnósticos rápidos y fiables, para prevenir y alertar a los agricultores sobre posibles plagas, ofreciendo datos y soluciones. [1] en 2020, diseñaron e implementaron un sistema experto para el diagnóstico, prevención, control de plagas y enfermedades en el cultivo de la uva; este sistema fue aceptado entre profesionales y técnicos, demostrándose su utilidad.

En plantas ornamentales, [9] en 2021, desarrollaron un sistema experto para diagnosticar plagas y enfermedades, beneficiando a usuarios no expertos y agilizando el análisis temprano. Estos autores utilizaron una metodología descriptiva y cuantitativa, y herramientas como ATOM y PhpMyAdmin, y se validó con diagnósticos expertos. Sin embargo, en este caso particular, concluyen los autores que se necesita más investigación en plagas específicas, para la obtención de más datos, y su aplicación en la práctica agrícola local para evaluar su eficacia y eficiencia.

2.4. Optimización de la gestión del suelo

[10], opinan que la implementación de prácticas de gestión del suelo, incluyendo la monitorización, análisis de datos y modelos de simulación, sobre su composición y condiciones aseguran su calidad y rendimiento agrícola. Al respecto, estos autores concluyeron que la implementación de inteligencia artificial en la agricultura, mediante técnicas como sensores remotos y riego automatizado con GPS, mejora el manejo del suelo y las prácticas agrícolas. Por su lado [11], mencionan que el modelaje en agricultura permite predecir comportamientos de plantas, animales y suelos, considerando interacciones complejas que mejoran la eficiencia en investigación y transferencia tecnológica y facilita la evaluación de innovaciones antes de su implementación, optimizando el uso de recursos.

De acuerdo con [12], se está desarrollando la predicción del rendimiento de los cultivos mediante indicadores de calidad del suelo, integrando técnicas de aprendizaje automático y procesamiento de datos de teledetección. Este enfoque implica el análisis de sus ventajas y desventajas, como la selección de datos mínimos, el uso de drones y satélites, el preprocesamiento de datos y la elección de algoritmos de aprendizaje automático. Los autores proponen un modelo basado en aprendizaje automático para estimar la calidad del suelo a nivel local, utilizando datos de teledetección. La producción del modelo se utilizaría para ajustar las prácticas agrícolas y mejorar el rendimiento de los cultivos.

En este contexto, [13], emplearon inteligencia artificial, específicamente redes neuronales, para analizar los suelos de invernaderos en el sector norte de la provincia de Cotopaxi en Ecuador. Utilizaron sensores y sistemas informáticos para recopilar datos y determinar la idoneidad del suelo para el cultivo de rosas. Los resultados indicaron que las redes neuronales son más estables que otros métodos para evaluar el estado del suelo, lo que permite tomar decisiones precisas sobre su manejo y control. Esta técnica demostró ser eficiente y se ajustó al rango establecido por el sector agrícola, optimizando así el análisis de suelos en invernaderos y facilitando la conversión de suelos dañados en fértiles.

2.5. Análisis y recomendaciones de manejo del agua

En este grupo se incluyen aquellas tecnologías que ofrecen análisis detallados y recomendaciones para el manejo eficiente del agua en la agricultura, incluyendo sistemas de riego inteligente y monitorización del uso del agua. Entre estas tecnologías se consideran los sensores y monitoreo remoto, que permiten medir el uso del agua en los cultivos y detectar fugas en los sistemas de riego, y los modelos de simulación que predicen la demanda del agua de los cultivos y optimizan la programación de riego.

[14] en 2006, desarrollaron un prototipo de sistema experto con el fin de mejorar la eficiencia del uso del agua en la irrigación del cultivo de maíz en Aguascalientes, México. Este sistema experto no solo incluía datos climáticos, sino que también proporcionaba recomendaciones específicas sobre la cantidad de agua a aplicar, el momento adecuado para el riego y los intervalos entre riegos, teniendo en cuenta las distintas etapas de desarrollo del cultivo.

Su estudio demostró la viabilidad de esta herramienta como una alternativa práctica para brindar apoyo a los agricultores, considerando las características climáticas y edáficas de cada región, así como los recursos disponibles en términos de sistemas y equipos de riego. Más recientemente, [15] en 2023, diseñaron un sistema de gestión de riego mediante red de sensores, a fin de aportar en la tecnificación del cultivo de Solanum phureja.

Para este cometido se tomaron las variables climáticas y edafológicas, los equipos a emplear y las necesidades del cultivo. Se utilizó el sistema Arduino, plataforma de código libre diseñada para dar facilidad en proyectos de electrónica, en combinación con el programa MatLab, que sirve para realizar cálculos matemáticos con vectores y matrices. La puesta en prueba del sistema demostró una precisión de 98% al momento de la medición de datos, permitiendo una gestión más precisa y eficiente del riego.

2.6. Predicción y gestión de condiciones climáticas

Los sistemas expertos para la agricultura se emplean tanto en la modelización climática como en el monitoreo para anticipar condiciones climáticas y gestionar los cultivos en consecuencia. Estos modelos pronostican eventos extremos como sequías, inundaciones y heladas, permitiendo a los agricultores tomar medidas preventivas y proteger sus cultivos.

Además, funcionan como sistemas de alerta temprana, previniendo a los agricultores sobre la inminente ocurrencia de eventos climáticos adversos, lo que les brinda tiempo para tomar las medidas necesarias y proteger sus cultivos. Con respecto al modelaje, su evaluación se basa en comparar las simulaciones con datos reales, siendo crucial su estrecha correspondencia.

Al menos en Colombia, la escasez de datos agronómicos dificulta la calibración de modelos como AquaCrop y DSSAT. [16] en 2013, menciona que se requiere información detallada sobre fenología, desarrollo y manejo de cultivos, así como variables meteorológicas.

Los modelos dinámicos como AquaCrop, Eto y DSSAT son más sólidos y ya se han utilizado y calibrado en ese país, recomendándose su implementación en el Instituto de Hidrología, Meteorología y Estudios Ambientales de Colombia (IDEAM). Por otro lado, los modelos AGROMET 2.9 y SIMPROC podrían utilizarse en el futuro, habiendo solicitado permisos de uso en México y Chile para su eventual calibración.

En el caso de los sistemas de alerta temprano, [17], presentaron AgroAlert, una herramienta que anticipa la sequía en cultivos específicos de tres a seis meses de antelación. Este sistema organiza, almacena, manipula, analiza y modela las condiciones agroclimáticas, identificando las zonas de cultivo más vulnerables y permitiendo la adaptación de criterios para analizar y predecir los riesgos.

2.7. Uso de nanomateriales

Las propiedades mecánicas, químicas, térmicas, ópticas, eléctricas y biológicas de los nanomateriales y nanopartículas hacen posible su aplicación en el área agrícola, entre otras. En este campo ofrece oportunidades para mejorar el suministro de fertilizantes, limpiar contaminantes y controlar fitopatógenos [18], (Pérez et al., 2024). Esta tecnología tiene potencial para una agricultura sustentable, pero plantea preocupaciones sobre su seguridad y sustentabilidad. Por esta razón, se requieren protocolos internacionales para la síntesis y caracterización de nanomateriales [19], en 2024.

3. Conclusiones

Los sistemas expertos en agricultura son fundamentales para la toma de decisiones integradas en el manejo de cultivos, la identificación y control de plagas, enfermedades, y la optimización de la gestión del suelo. Además, proporcionan análisis y recomendaciones para el manejo del agua, la predicción y gestión de condiciones climáticas, y exploran el uso de nanomateriales para mejorar la eficiencia agrícola. Estos sistemas permiten una gestión precisa y sostenible, mejorando la productividad y la resiliencia frente a desafíos ambientales y de producción.

Los esfuerzos de los investigadores están encaminados al desarrollo de SE que orienten a los productores a tomar decisiones sobre diferentes aspectos del manejo de cultivos en todo su proceso. Uno de los principales desafíos en la industria agrícola es transferir la información más reciente y actualizada a los agricultores, se desea que los actuales SE proporcionen información en tiempo real a agricultores e investigadores a través de tecnologías del Internet utilizando un sistema experto difuso.

Referencias

1. Yaya-Lévano, J.E.R., Angulo-Altamirano, E.D.: Diseño e implementación de un sistema experto para optimizar el control de plagas y enfermedades en el cultivo de la uva. Ñawparisun-Revista de Investigación Científica, vol. 3, no. 1, pp. 83–96 (2020). DOI: 10.47190/nric.v3i1.130.
2. Badaró, S., Ibañez, L.J., Agüero, M.J.: Sistemas expertos: Fundamentos, metodologías y aplicaciones. Dialnet, no. 13, pp. 349–364 (2013)
3. Falcón-Suárez, J.A., Betancourt-Perera, R., Liriano-González, R., Pérez-Hernández, Y.: Software para el apoyo a la toma de decisiones en el sector agrícola. Revista Ingeniería Agrícola, vol. 13, no. 3, pp. e08 (2023)
4. Araújo, S.O., Peres, R.S., Ramalho, J.C., Lidon, F., Barata, J.: Machine Learning Applications in Agriculture: Current Trends, Challenges, and Future Perspectives. Agronomy, vol. 13, no. 12, pp. 2976 (2023). DOI: 10.3390/agronomy13122976.
5. Benos, L., Tagarakis, A.C., Dolias, G., Berruto, R., Kateris, D., Bochtis, D.: Machine Learning in Agriculture: A Comprehensive Updated Review. Sensors, vol. 21, no. 11, pp. 3758 (2021). DOI: 10.3390/s21113758.
6. Tripathi, P., Kumar, N., Rai, M., Shukla, P.K., Verma, K.N.: Applications of Machine Learning in Agriculture. IGI Global Scientific Publishing, pp. 99–118 (2023). DOI: 10.4018/ 978-1-6684-6418-2.ch006.

7. Bula, H.D., Aramendiz, H., Salas, D., Vergara, W.E., Villadiego, A.L.: Sistema experto para el diagnóstico de plagas y enfermedades en los cultivos de berenjena (Solanum Melongena l.) en la región caribe de Colombia. Ingeniería e Innovación, vol. 1, no. 1 (2013). DOI: 10.21897/23460466.765.
8. Medina-Carbó, Y., Santana-Ges, I.M., Leo-González, S.: Sistema experto para el diagnóstico de enfermedades y plagas en los cultivos del arroz, tabaco, tomate, pimiento, maíz, pepino y frijol. Revista Cubana de Ciencias Informáticas, vol. 10, no. 10, pp. 1–19 (2020). DOI: 10.48550/arXiv.2007.11038.
9. Sandoval-Pillajo, A.L., Checa-Cabrera, M.A., Díaz-Vásquez, R.A., Acosta-Espinoza, J.L.: Sistema experto para el diagnóstico y tratamiento de enfermedades y plagas en plantas ornamentales. Universidad y Sociedad, vol. 13, no. 3, pp. 505–511 (2021)
10. Bonilla-Segovia, J.S., Dávila-Rojas, F.A., Villa-Quishpe, M.W.: Estudio del uso de técnicas de inteligencia artificial aplicadas para análisis de suelos para el sector agrícola. RECIMUNDO, vol. 5, no. 1, pp. 4–19 (2021). DOI: 10.26820/recimundo/5.(1) .enero.2021.4-19.
11. Martínez, B., Ruiz-Rosado, O., Gallardo-López, F., Pérez-Hernández, P., Martínez-Becerra, Á., Vargas-Villamil, L.: Aplicación de modelos de simulación en el estudio y planificación de la agricultura, una revisión. Tropical and subtropical agroecosystems, vol. 14, pp. 999–1010 (2011)
12. Tobar-Díaz, R., Gao, Y., Mas, J.F., Cambrón-Sandoval, V.H.: Clasificación de uso y cobertura del suelo a través de algoritmos de aprendizaje automático: Revisión bibliográfica. Revista de Teledetección, no. 62, pp. 1–19 (2023). DOI: 10.4995/raet.2023. 19014.
13. Escobar-Iza, R.D., Maliza-Bedon, D.S., Cadena-Moreano, J.A.: Análisis de suelos utilizando redes neuronales en las florícolas de rosas del sector norte de la provincia de cotopaxi. RECIMUNDO, vol. 5, no. 2, pp. 316–330 (2021). DOI: 10.26820/recimundo/ 5.(2).abril.2021.316-330.
14. Ramos-Gourcy, F., Macias-Luevano, J., Díaz, F., Balandrán, F., Mora, M., López, V.: Desarrollo y evaluación de un sistema experto (prototipo) que auxilie en el proceso de irrigación del cultivo de maíz (zea mays l.) en aguascalientes. Investigación y Ciencia, vol. 14, no. 36, pp. 15–24 (2006)
15. Cubides, A., Bayona-Espitia, J.D., Alejandra, J.: Diseño de un sistema de gestión de riego mediante red de sensores a fin de aportar en la tecnificación del cultivo de solanum phureja para la sostenibilidad de la vereda santa ana en el municipio de monguí. Maestría Manejo y Sostenibilidad Ambiental (2023)
16. Fernández, M.: Diagnóstico de modelos agroclimáticos. Evaluación del riesgo agroclimático por sectores. FONADE/IDEAM (2013)
17. Sosa-Escalona, Y., Peña-Casadevalls, M., Santiesteban-Toca, C.E.: Sistema para la alerta temprana de los efectos del cambio climático en la agricultura. Revista Cubana de Ciencias Informáticas, vol. 11, no. 3, pp. 64–76 (2017)
18. Hernández, H.P., López-Valdez, F., Juárez-Maldonado, A., Méndez-López, A., Sarabia-Castillo, C.R., García-Mayagoitia, S., Torres-Gómez, A.P., Valle-García, J.D., Pérez-Moreno, A.Y.: Implicaciones de los nanomateriales utilizados en la agricultura: una revisión de literatura de los beneficios y riesgos para la sustentabilidad. Mundo Nano. Revista Interdisciplinaria en Nanociencias y Nanotecnología, vol. 17, no. 32, pp. 1–50 (2023). DOI: 10.22201/ceiich.24485691e.2024.32.69720.
19. Rodríguez-González, V., Díaz-Cervantes, E.: Potencial de los nanomateriales en la agricultura: retos y oportunidades. Mundo Nano. Revista Interdisciplinaria en Nanociencias y Nanotecnología, vol. 17, no. 32, pp. 1–20 (2023). DOI: 10.22201/ceiich.24485691e. 2024.32.69802.

ISSN 1870-4069

Evaluación de técnicas de aprendizaje automático supervisado para la predicción de disponibilidad de agua subterránea en acuíferos de México

Alberto González Sánchez[1], Ronald Ernesto Ontiveros Capurata[1], Miguel Antonio Vega Castro[2]

[1] Instituto Mexicano de Tecnología del Agua, Coordinación de Seguridad Hídrica, México

[2] Universidad Politécnica del Estado de Morelos, Maestría en Tecnologías de la Información, México

{alberto_gonzalez,ronald.ontiveros}@tlaloc.imta.mx, 16090507@upemor.edu.mx

Resumen. Hoy en día, la sobreexplotación del agua subterránea es un problema global. En México, hay 653 acuíferos que representan el 39.1% del volumen destinado a usos consuntivos. La Comisión Nacional del Agua es responsable de gestionar este recurso, por lo periódicamente determina la cantidad de agua disponible para extracción considerando factores como el concesionamiento y la recarga. Sin embargo, se ha observado un número creciente de acuíferos en déficit, lo que plantea desafíos para prever su disponibilidad. Los modelos físicos pueden abordar este problema mediante simulaciones, pero requieren una gran cantidad de información, tiempo y recursos. Los algoritmos de aprendizaje supervisado representan una alternativa, ya que pueden detectar tendencias sin el conocimiento profundo que demandan los modelos físicos. Este trabajo evalúa cuatro técnicas para la predicción de disponibilidad de agua en acuíferos: regresión con máquinas de soporte vectorial, árboles de modelos M5', bosques aleatorios (*Random-Forest Regression, RFR*) y redes neuronales artificiales. Los modelos fueron entrenados con información climatológica, uso de suelo y distribución del concesionamiento en años seleccionados entre 1997 y 2015, y fueron evaluados con datos de 2018 y 2020. La comparativa mostró buen desempeño de *RFR*, con un coeficiente *r* alto y errores *RMSE* bajos. La comparativa de acuíferos en déficit mostró una coincidencia del 55.10% para 2018 y del 48.72% para el año 2020. Así, las *RFR* pueden predecir de manera adecuada la disponibilidad de agua en acuíferos a corto plazo, lo que puede ayudar a una gestión más sustentable del recurso.

Palabras clave: Sobreexplotación de acuíferos, *machine learning*, máquinas de soporte vectorial, redes neuronales artificiales, M5', bosques aleatorios.

Assessment of Supervised Machine Learning Techniques for Predicting Groundwater Availability in Aquifers of Mexico

Abstract. Today, the overexploitation of groundwater is a global problem. In Mexico, there are 653 aquifers constituting 39.1% of the volume allocated for consumptive uses. The National Water Commission is responsible for managing this resource, periodically assessing the amount of water available for extraction by considering factors such as concessions and recharge. However, an increasing number of aquifers in deficit have been observed, presenting challenges in accurately predicting their availability. While physical models can address this problem through simulations, they require extensive information, time, and resources. Supervised learning algorithms offer an alternative solution, as they can detect trends without the deep knowledge required by physical models. This work evaluates four techniques for predicting water availability in aquifers: regression with support vector machines, M5' model trees, random forests (Random-Forest Regression, RFR) and artificial neural networks. The models were trained using climatological information, land use and concession distribution data from selected years between 1997 and 2015. They were then evaluated using data from 2018 and 2020. The comparison demonstrated strong performance of RFR, exhibiting a high correlation coefficient (r) and low RMSE errors. The comparison of aquifers in deficit revealed a coincidence of 55.10% for 2018 and 48.72% for 2020. Therefore, RFR can adequately predict the availability of water in aquifers in the short term, aiding in the sustainable management of this vital resource.

Keywords: Aquifers overexploitation, machine learning, support vector machine regression, artificial neural networks, M5', random forests regression.

1. Introducción

Hoy en día, el uso desmedido del agua subterránea ha provocado una reducción en la disponibilidad de este recurso. En México existen 653 acuíferos, que aportan el 39.1% del volumen destinado para usos consuntivos [1], entre los que destacan el consumo humano (14.4%) y la agricultura (60%). La Comisión Nacional del Agua (CONAGUA) ha tratado de hacer un uso eficiente del recurso, estimando a partir del año 2001 la disponibilidad media anual (DMA) de agua por acuífero, considerando el volumen concesionado, la recarga y otras variables [2]. A la fecha, se cuenta con 5 actualizaciones de la DMA de los 653 acuíferos: 2010-2011, 2013, 2015, 2018 y 2020 [3–13]. Históricamente, los valores de DMA han mostrado un deterioro en la cantidad de agua disponible para extracción y un incremento en el número de acuíferos sobreexplotados (Tabla 1).

Un uso sustentable del recurso hídrico y una asignación más equilibrada de los títulos de concesión requieren un análisis complejo que contemple elementos geográficos, ambientales y climatológicos, con el fin de estimar con precisión los valores de DMA.

Tabla 1. Acuíferos en déficit y su disponibilidad promedio según publicaciones oficiales.

Fecha de publicación en el DOF[1]	Cantidad de acuíferos		Disponibilidad promedio total (%)
	En déficit	Con disponibilidad	
08/07/2010, 16/08/2010, 25/01/2011, 14/12/2011[2]	174	479	14.51
20/12/2013	193	460	12.50
20/04/2015	203	450	11.46
04/01/2018	245	408	-2.46
17/09/2020	275	378	-12.01

Los acuíferos tienen una naturaleza dinámica difícil de modelar; responden a cambios en el uso y cobertura del suelo, clima, volumen de recarga y extracción [14]. Predecir la recarga de los acuíferos es complicado, ya que no se puede medir directamente [15]. Un método para detectar el agotamiento de los acuíferos es mediante modelos de simulación física, que demandan gran cantidad de información y son costosos, ya que dependen de la medición directa de variables de campo para su calibración y validación [16]. Una alternativa es utilizar aprendizaje automático (*machine learning, ML*), que es capaz de construir modelos a partir de registros previamente etiquetados [17].

Estos modelos identifican tendencias sin un conocimiento profundo de los atributos subyacentes utilizados en los modelos físicos de flujo de agua subterránea [18]. Diversos algoritmos de *ML* han sido utilizados para abordar el problema de explotación de agua subterránea [19], por ejemplo, redes neuronales artificiales (RNA) [20], bosques aleatorios (*Random Forest*, *RF*) y las máquinas de soporte vectorial (*Support Vector Machines, SVM*) [21], siendo pocos los trabajos realizados en México. En este contexto, este trabajo evalúa el uso de cuatro algoritmos de *ML* (M5', *RF*, RNA y *SVM*) para predecir la disponibilidad de agua en acuíferos de este país.

2. Metodología

2.1 Construcción del conjunto de entrenamiento

El conjunto de entrenamiento se construyó a partir de fuentes de datos oficiales con información histórica de 1997 a 2021 de las variables que afectan la disponibilidad de agua subterránea, como el clima, uso del suelo y distribución de las concesiones para el aprovechamiento del agua (Tabla 2). El volumen disponible por acuífero (en hectómetros cúbicos, hm^3), es la variable de respuesta y fue obtenido de las publicaciones periódicas de la CONAGUA (Tabla 1). La DMA por acuífero en años anteriores al 2011 fue estimada con la información de la fuente FD3 (Tabla 2),

[1] Diario Oficial de la Federación. Órgano del Gobierno Constitucional Mexicano que tiene la función de publicar leyes, reglamentos, acuerdos y demás actos expedidos por los poderes de la Federación.

[2] La primera publicación de la disponibilidad media de los acuíferos se hizo en entregas parciales.

Tabla 2. Fuentes de datos utilizados para la obtención de atributos predictores.

Identificador de Fuente de datos y formato	Descripción
FD1: Uso del suelo y vegetación Formato: vectorial (shapefile)	Capas vectoriales con la clasificación del uso de suelo y vegetación. Series I-VII de uso de suelo de INEGI (1997, 2001, 2005, 2010, 2013, 2016 y 2021). https://www.inegi.org.mx/temas/ususuelo/#descargas
FD2: Clima (temperatura, precipitación y evapotranspiración) Formato: ráster	Capas ráster con los promedios anuales de temperaturas, precipitación y evapotranspiración potencial (1997 a 2021) https://www.globalclimatemonitor.org/.
FD3: Títulos de concesión y sus anexos Formato: tabular (CSV)	Datos de títulos de concesión, tipo de uso (agrícola, industrial, urbano, etc.), volumen amparado y fecha de otorgamiento. https://datos.gob.mx/busca/dataset/concesiones-asignaciones-permisos-otorgados-y-registros-de-obras-situadas-en-zonas-de-libre-alu

utilizando la fecha de registro de la concesión y la diferencia con la recarga reportada en el período 2010-2011. De esta manera se obtiene un rango histórico comparable al de las variables predictoras.

Los datos recopilados se sometieron a un proceso de limpieza e integración. En primer lugar, se verificó la consistencia de las claves, corroborando que fueran homogéneas y que permitieran relacionar las distintas fuentes involucradas.

Cuando fue necesario, se corrigieron datos erróneos y/o atípicos empleando el manejador de base de datos *MySQL*. La integración se realizó considerando la compatibilidad histórica entre los registros, seleccionando como años representativos los correspondientes a las publicaciones de la DMA de los acuíferos, usando 2010 para las entregas realizadas entre 2010-2011 (ver Tabla 1).

Así, se integró un primer conjunto de datos tabular con los atributos año, identificador del acuífero, recarga y DMA (atributo a predecir), al cual se le añadió el resto de los atributos predictores. Para relacionar este conjunto con el uso de suelo, se usó el año representativo para "emparejar" con la serie INEGI más cercana en el tiempo.

De esta forma, 2010 se asoció con la serie IV, publicada entre 2007-2010, 2013 con la serie V, publicada el mismo año, 2015 con la serie VI y 2018 y 2020 con la serie VII. Para contar con más registros históricos, se usó la estimación de disponibilidad realizada con la FD3 en años previos al 2010, asociando estos datos a las series de uso de suelo anteriores.

Así, se hicieron estimaciones para 1997 (Serie 1), 2001 (Serie II) y 2005 (Serie III). En este período no se encontró información para 8 acuíferos (1% del total), por lo que fueron eliminados de todo el conjunto, ajustando el análisis a 645 acuíferos.

Los atributos para clima (temperatura, precipitación y evapotranspiración), se obtuvieron de la FD2. Temperatura y precipitación son las principales variables utilizadas en trabajos similares [19]. La información fue extraída mediante herramientas espaciales de intersección y estadísticas de grupo del software QGIS empleando la capa vectorial de acuíferos.

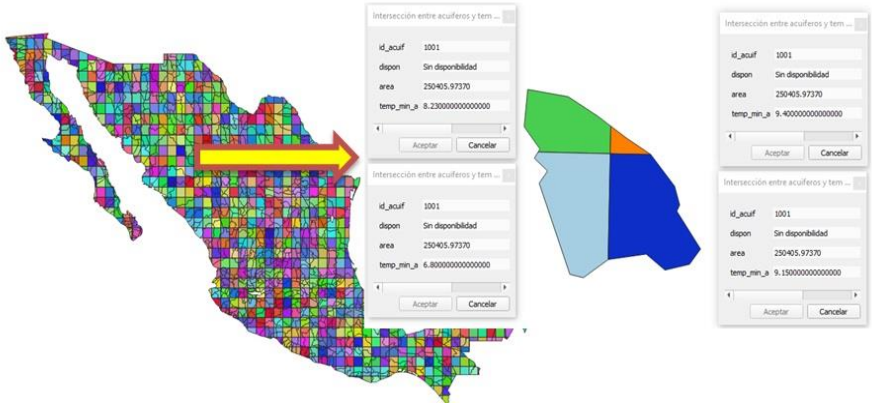


Fig. 1. Procesamiento de extracción de información climática (temperatura) a partir de la intersección del ráster con la capa vectorial de los acuíferos en QGIS.

Tabla 3. Métricas para la evaluación de los modelos. n es el total de observaciones; y_i el valor real de la observación i; $\hat{y}_i$ es el valor estimado por el modelo para i; $\overline{\hat{y}}$ es la media del conjunto de estimaciones; $\overline{y}$: la media del conjunto de observaciones; $r_i = y_i - \hat{y}_i$.

Métrica	Unidades	Cálculo
R	(adim)	$\frac{\sum_{i=1}^{n}(y_i - \bar{y}) \; - \; (\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}$
RMSE	(misma que el valor estimado y el valor real)	$\sqrt{\frac{\sum_{i=1}^{n} r_i^2}{n}}$
RRSE	%	$\sqrt{\frac{\sum_{i=1}^{n} r_i^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}} x100$

Por ejemplo, la Figura 1 muestra la extracción del dato de temperatura mínima, donde se muestra el polígono del acuífero 1001 "Valle de Santiaguillo" y los valores de las celdas circunscritas de la capa ráster. Estos valores fueron promediados por acuífero para cada año disponible.

Para obtener la información de un año representativo, se promediaron los datos de los últimos tres años, incluyendo el año de referencia.

Este procedimiento generó un conjunto de entrenamiento con 5160 registros (ocho años por 645 acuíferos) y 23 atributos (incluyendo el atributo a predecir). Los atributos seleccionados fueron los siguientes:

a) Año representativo del período al que corresponden los datos del registro (1997, 2001, 2005, 2010, 2013, 2015, 2018 y 2020) [AÑO].

b) Clave oficial del acuífero (identificador del estado y consecutivo) [ID_ACUIF].

Tabla 4. Valores de métricas obtenidas por M5' en el conjunto de prueba.

Tipo de árbol M5'	Con poda	Métrica		
		R	***RMSE***	***RRSE***
Árbol de modelos	Sí	**0.949**	69.405	39.86%
	No	0.945	74.502	42.78%
Árbol de regresión	Sí	0.000	174.141	100.00%
	No	0.939	**64.848**	**37.24%**

c) Atributos del clima. Temperatura media (°C), precipitación (hm^3) y evapotranspiración potencial (mm) promedio que se presentaron en el acuífero en los últimos tres años [TEMP, PRECIP, ET].

d) Uso de suelo y vegetación. Porcentaje de cada tipo de cobertura de suelo que presenta el acuífero en el año más cercano al representativo [S_AGR, S_ASEN, S_BOSQUE, S_AGUA, S_SELVA, S_VEG, S_OTROS].

e) Tipo de concesionamiento. Porcentaje del uso del agua concesionada para cada tipo de uso estimado para el año representativo [R_ACUA, R_AGRI, R_AGROIND, R_COM, R_DIF_USOS, R_DOMES, R_INDUS, R_OTROS, R_PECUARIO, R_PUB_URBANO, R_SERV].

f) Volumen disponible para extracción que presenta el acuífero (hm^3) (variable a predecir) [VOL_DISP].

2.2 Selección de algoritmos de aprendizaje automático

Se seleccionaron cuatro algoritmos de *ML* comúnmente empleados para predicción numérica; específicamente, las implementaciones de la *suite* para minería de datos Weka [22]. A continuación, se describe cada algoritmo y su parametrización para este trabajo.

Árboles de modelos de regresión lineal M5'. El algoritmo M5', se basa en un árbol de decisión que se construye a partir de un algoritmo recursivo, realizando la toma de decisiones de enrutado en nodos a partir de los valores de los atributos. Al final del enrutado, cada nodo hoja permite obtener el valor de una instancia mediante un modelo de regresión lineal [23], pero también tiene la opción de generar un valor numérico, por lo que en este trabajo se probaron ambas opciones.

También se contempló la opción de podar el árbol, generando cuatro combinaciones posibles: árboles de modelos con poda, árboles de modelos sin poda, árboles de valores constantes con poda y árboles de valores constantes sin poda. Se dejó un mínimo de 2 objetos en cada nodo hoja.

Bosques Aleatorios. La regresión con bosques aleatorios (*Random-Forest Regression, RFR*) tiene su fundamento en el método de *bagging* (embolsado) y los subespacios aleatorios [24]. El algoritmo comienza con la generación de *K* conjuntos obtenidos con la extracción aleatoria de ejemplos con reemplazo del conjunto de aprendizaje, y utiliza cada conjunto para crear un árbol de regresión. En el proceso de

Tabla 5. Valores de métricas obtenidas por *Random-Forest* en el conjunto de prueba.

Tamaño del bosque (árboles)	Límite en profundidad	Variables (*m*)	Métrica		
			R	***RMSE***	***RRSE***
500	10	5	0.959	52.675	30.25%
		8	0.960	54.867	31.51%
	20	5	0.958	53.128	30.51%
		8	0.960	54.752	31.44%
	Sin limitar	5	0.958	53.076	30.48%
		8	0.960	54.752	31.44%
1000	10	5	**0.961**	50.938	29.25%
		8	0.960	54.582	31.34%
	20	5	**0.961**	50.750	29.14%
		8	**0.961**	54.165	31.10%
	Sin limitar	5	**0.961**	**50.681**	**29.10%**
		8	**0.961**	54.126	31.08%

construcción de cada árbol, cada partición es producto de considerar un pequeño conjunto de las variables de entrada de forma aleatoria [25], eligiendo para dividir a la variable con el índice de Gini más bajo. Para la tarea de regresión, el resultado es el promedio de la estimación de los *K* árboles aleatorios del bosque.

De acuerdo con [26], los hiperparámetros más relevantes son el número de variables candidatas por partición (*m*) y el número de árboles (*K*). Los mismos autores sugieren un valor de *m*=*p*/3 para problemas de regresión (*p* es el número de atributos predictores, 23 en este caso), mientras que Weka utiliza *int* (log2(p) + 1)). En este trabajo se consideraron ambas opciones (5 y 8).

Estos autores también sugieren un valor de 500 o 1000 para el número de árboles. En adición, *Weka* permite especificar la profundidad máxima de los árboles, usando en este caso 5, 10 y sin límite. Por lo anterior, para esta técnica se validaron 12 combinaciones de parámetros: *m*={5,8}, *K*={500,1000} y una profundidad de árboles = {5,10,ilimitada}.

Máquinas de Soporte Vectorial para Regresión. La regresión con máquinas de soporte vectorial (*Support Vector Machines Regression, SVMR*) pertenece a un grupo de algoritmos de aprendizaje estadístico supervisado. En su forma más simple, el objetivo de la técnica es obtener una función lineal f(x)=<w,x>+b con $w \in R^N$ y $b \in R$ para un conjunto de entrenamiento $\{(x_1, y_1),...,(x_m, y_m)$. La función *f(x)* debería tener como máximo una desviación ε de los valores y_1 actuales y a la vez ser lo más plana posible. La "planitud" se puede obtener con un valor pequeño para w. El problema de optimización se puede escribir como se muestra en (1) [27]:

$$\text{Minimizar} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)\,, \tag{1}$$

Tabla 6. Valores de métricas obtenidas por las RNAs en el conjunto de prueba.

Ciclos de entrenamiento	Neuronas en la capa oculta	Métrica		
		R	**RMSE**	**RRSE**
1000	5	0.928	91.57	52.58%
	10	**0.930**	**89.63**	**51.47%**
	15	0.924	94.95	54.52%
5000	5	0.921	103.14	59.23%
	10	0.921	102.86	59.07%
	15	0.916	103.47	59.42%
10000	5	0.919	104.84	60.20%
	10	0.919	104.71	60.13%
	15	0.915	104.21	59.84%

Tabla 7. Resultados para las métricas de evaluación (todos los algoritmos).

Algoritmo	Métrica		
	r	***RMSE***	***RRSE***
M5'	0.949	69.405	39.86%
RFR	**0.961**	**50.681**	**29.10%**
RNA	0.930	89.630	51.47%
SVMR	0.920	104.675	60.11%

$$\text{Sujeto a:} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i, \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \tag{2}$$

donde ξ_i y ξ_i^* se introducen como variables de holgura para restricciones inviables.

C se denomina parámetro de regularización y determina la cantidad de desviaciones mayores que ε que son aceptadas. Si no es posible separar el conjunto de ejemplos con una función lineal, se recurre a la transformación del espacio original mediante una función no lineal denominada *kernel*. Para este trabajo, se utiliza la versión de *SVMR* implementada en Weka, que aplica una versión mejorada del algoritmo de aprendizaje de optimización mínima secuencial [28], con C=1 y un *kernel* polinomial de grado 1.

Redes neuronales artificiales (RNA). Las redes neuronales se dividen en una capa de entrada, una de salida y una o más capas ocultas. La capa de entrada consiste de neuronas que reciben las señales o datos del entorno (atributos de entrada). La capa oculta proporciona grados de libertad que le permiten presentar características más complejas. La capa de salida está compuesta de neuronas que proporcionan la respuesta de la red neuronal. Las RNA son empleadas con frecuencia para clasificación y predicción de modelos de series históricas [29].

Existen distintas formas de interconectar las neuronas en una red neuronal (topología). Para este trabajo, se utilizó la topología y esquema de entrenamiento más

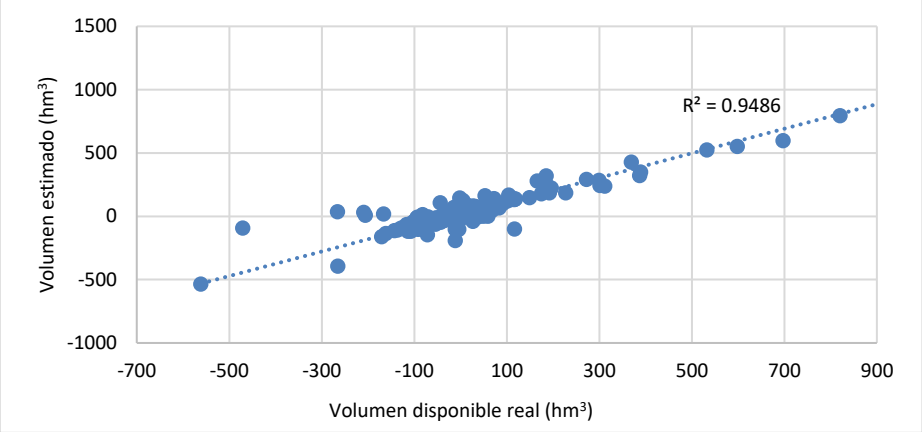


Fig. 2. Volumen disponible real versus el estimado por *RFR* para todos los acuíferos (2018).

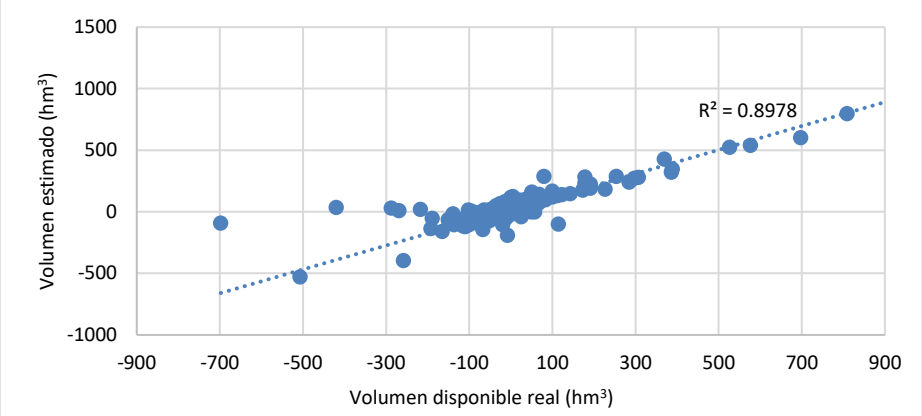


Fig. 3. Volumen disponible real versus el estimado por *RFR* para todos los acuíferos (2020).

común, que es perceptrón multicapa entrenada por retropropagación. Para la capa oculta, se probaron combinaciones de 5, 10 y 15 neuronas, con 1000, 5000 y 10000 ciclos de entrenamiento con decaimiento, ambos parámetros utilizados en trabajos similares realizados con anterioridad [30].

2.3 Evaluación de los algoritmos de aprendizaje

Los algoritmos fueron evaluados usando la técnica *percentage-split*, por lo que el conjunto de aprendizaje se dividió en dos: entrenamiento y prueba. El primer subconjunto se integró con información de los primeros seis años (1997, 2001, 2005, 2009, 2013 y 2015), utilizando 3870 muestras (75% de los registros disponibles). El subconjunto de prueba se integró con la información de los últimos 2 años (2018 y 2020), representando el 25% de los registros restantes.

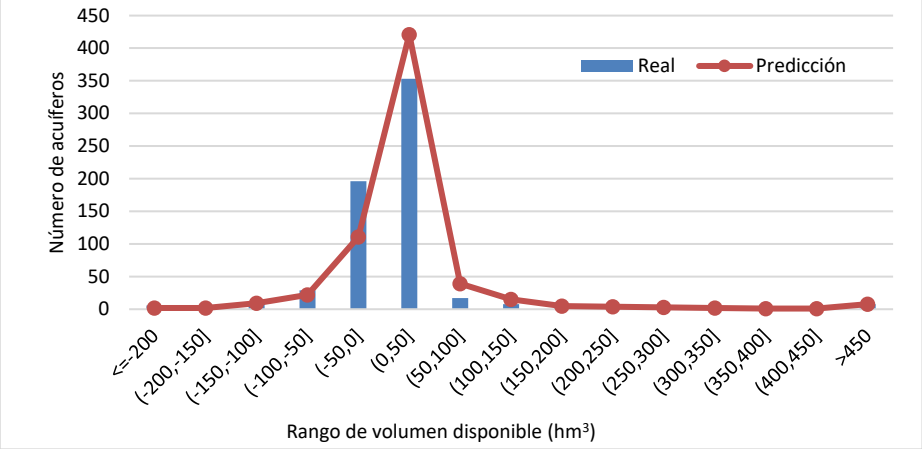


Fig. 4. Histograma de frecuencias de acuíferos por rango de volumen disponible y estimación por *RFR* (2018).

Al tratarse de modelos de predicción numérica, la evaluación del conjunto de prueba se realizó con las métricas de coeficiente de correlación (r), error cuadrático medio (*Root Mean Square Error*, *RMSE*) y el error cuadrático relativo (*Root Relative Square Error*, *RRSE*), que se describen en la Tabla 3 [23].

3. Resultados

En esta sección se muestran los resultados de la evaluación de los algoritmos para el conjunto de prueba (años 2018 y 2020). Primero, se presentan los resultados para cada técnica. Al final, se presenta una comparativa general entre todas las técnicas.

3.1 Resultados por algoritmo

La Tabla 4 muestra los resultados del algoritmo M5'. Se resaltan en negritas los mejores valores de cada métrica. Aunque los árboles de modelos con poda tuvieron una *r* más alta, los errores más bajos se obtuvieron con un árbol de regresión sin poda. La Tabla 5 muestra los resultados del algoritmo *RFR*. Los *RMSE* y *RRSE* más bajos se obtuvieron en un bosque de 1000 árboles, con m=5 y sin límite de profundidad. La Tabla 6 muestra los resultados de las RNAs.

Los mejores valores para las métricas de error se encontraron con 1000 ciclos de entrenamiento y 10 neuronas en la capa oculta. Finalmente, la técnica *SVMR* fue evaluada con los parámetros anteriormente especificados. En este caso, se trató de un único resultado, obteniendo una r=0.920, con un *RMSE*=104.675 y *RRSE*=60.11%.

3.2 Comparación entre algoritmos

La Tabla 7 concentra los mejores resultados encontrados para todos los algoritmos bajo análisis. Se observa que el algoritmo *RFR* obtuvo el valor mayor para *r* y los

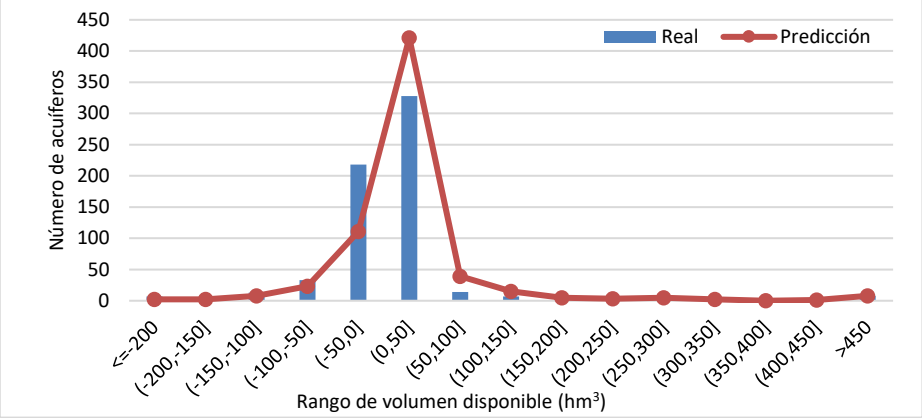


Fig. 5. Histograma de frecuencias de acuíferos por rango de volumen disponible y estimado por *RFR* (2020).

valores más bajos para *RMSE* y *RRSE*. No obstante, se debe considerar que los algoritmos de aprendizaje tienen diversos grados de sensibilidad al peso de sus parámetros. Por ejemplo, *SVM* es más sensible que *RF* [31], y la eficiencia de las RNAs dependen en gran parte de su topología y ciclos de aprendizaje [38].

El uso de alguna técnica de sintonización paramétrica podría mejorar las condiciones de comparación de los algoritmos. Una vez determinado el algoritmo que produce mejores resultados, se puede hacer un análisis más específico.

Así, las gráficas de dispersión de las Figuras 2 y 3 muestran el ajuste obtenido por *RFR* en cada año del conjunto de prueba. Dada la densidad de acuíferos, un análisis de frecuencias puede mejorar la visualización de los errores cometidos por el algoritmo.

Las Figuras 4 y 5 muestran el histograma de los acuíferos por rango de volumen disponible para cada año, sobreponiendo la frecuencia calculada con el volumen estimado por *RFR*. En la comparativa, se observa que la predicción tiene una forma similar a la distribución de probabilidad del valor real de la DMA. Sin embargo, se observa también que el algoritmo subestima la cantidad de acuíferos que tienen una disponibilidad entre el rango de -50 a 0 hm^3, y sobreestima en el rango de 0 a 50 hm^3. Esto es consistente en los dos años presentes en el conjunto de prueba.

Resulta fácil visualizar, que hay una tendencia de los acuíferos que van hacia el estado de disponibilidad bajo 0 (déficit), pero el algoritmo no logra identificar todos los casos. Finalmente, se verificaron las coincidencias entre la predicción de los acuíferos en déficit (disponibilidad negativa) para los años 2018 y 2020, contemplando también su contraparte positiva, obteniendo una coincidencia del 81.24% para 2018 y del 76.79% para 2020. Si la comparación se realiza únicamente con aquellos acuíferos que caen en déficit, la coincidencia es de 55.10% y 48.72%, respectivamente.

4. Conclusiones

En la evaluación de algoritmos de aprendizaje automático para la predicción del volumen disponible en los acuíferos, la regresión con bosques aleatorios (*RFR*) obtuvo

mejores resultados, seguido de M5', RNA y *SVMR*. La ventaja de *RFR* sobre M5' era esperada, ya que el primero se construye a partir de múltiples árboles. Por otra parte, RNA y *SVMR* tienen más combinaciones paramétricas que *RFR*. En este trabajo se utilizaron los valores paramétricos más comunes; sin embargo, los algoritmos de aprendizaje tienen diferentes niveles de sensibilidad al peso de sus parámetros, por lo que una exploración más profunda en el proceso de sintonización (como *grid search*) podría generar una comparación más equitativa.

La clasificación directa de acuíferos que caerán en déficit dada la predicción del volumen disponible realizada por *RFR* tuvo una coincidencia del 55.10% para 2018 y del 48.72% para 2020. La estimación fue consistente, pero baja. En este punto, es importante recordar que el estado de déficit ocurre cuando la disponibilidad es menor a 0, por lo que cantidades cercanas a dicho valor, pero superiores, no se consideran en este estado. Dado que las métricas de *RFR* son buenas, una ligera ampliación del rango para determinar el riesgo de déficit alrededor de la predicción numérica podría aumentar las coincidencias. Desde esta perspectiva, lo más adecuado es usar una técnica de clasificación de dos clases (en déficit/con disponibilidad), por lo que queda pendiente validar este enfoque y su comparación con los resultados de la predicción numérica.

De lo anterior, se concluye que *RFR* puede predecir de manera aceptable la disponibilidad de agua en los acuíferos en corto plazo, no así el estado final de déficit. No obstante, y dado el bajo *RRSE* obtenido, *RFR* puede ser útil para una gestión provisoria del agua subterránea, mejorando la protección y conservación del recurso. En este sentido, es importante señalar que no se sugiere dejar la responsabilidad del concesionamiento a un modelo de aprendizaje. La abstracción inherente al proceso de construcción de estos modelos puede omitir elementos sociales y ambientales importantes, lo que es especialmente crítico en la administración del agua. Depender únicamente de modelos de aprendizaje automático para su gestión plantea un dilema ético; este proceso debe ser transparente, equitativo y responsable, en beneficio de todas las partes involucradas.

Referencias

1. CONAGUA: Estadísticas del agua en México 2018. vol..303 (2018)
2. CONAGUA: Norma oficial mexicana NOM-011-CONAGUA-2000 conservación del recurso agua, que establece las especificaciones y el método para determinar la disponibilidad media anual de las aguas nacionales (2000)
3. CONAGUA: ACUERDO por el que se da a conocer la ubicación geográfica de 371 acuíferos del territorio nacional, se actualiza la disponibilidad media anual de agua subterránea de 282 acuíferos, y se modifica, para su mejor precisión, la descripción geográfica de 202 (2009)
4. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 36 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológicas que se indican (2010)
5. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 44 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológicas que se indican, (2010)

6. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 41 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológicas que se indican (2010)
7. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 50 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológicas administrativas que se indican (2011)
8. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 58 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológicas administrativas que se indican (2011)
9. CONAGUA: ACUERDO por el que se da a conocer el resultado de los estudios de disponibilidad media anual de las aguas subterráneas de 142 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológico-administrativas que se indican (2011)
10. CONAGUA: ACUERDO por el que se actualiza la disponibilidad media anual de agua subterránea de los 653 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológico-administrativas que se indican (2013)
11. CONAGUA: ACUERDO por el que se actualiza la disponibilidad media anual de agua subterránea de los 653 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológico-administrativas que se indican (2015)
12. CONAGUA: ACUERDO por el que se actualiza la disponibilidad media anual de agua subterránea de los 653 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las Regiones Hidrológico-Administrativas que se indican (2018)
13. CONAGUA: ACUERDO por el que se actualiza la disponibilidad media anual de agua subterránea de los 653 acuíferos de los Estados Unidos Mexicanos, mismos que forman parte de las regiones hidrológico-administrativas que se indican (2020)
14. Wang, X., Liu, T., Zheng, X., Peng, H., Xin, J., Zhang, B.: Short-term Prediction of Groundwater Level Using Improved Random Forest Regression with a Combination of Random Features. Applied Water Science, vol. 8, no. 5 (2018). DOI: 10.1007/s13201-018-0742-6.
15. Crosbie, R.S., Davies, P., Harrington, N., Lamontagne, S.: Ground Truthing Groundwater-Recharge Estimates Derived from Remotely Sensed Evapotranspiration: A Case in South Australia. Hydrogeology Journal, vol. 23, no. 2, pp. 335–350 (2014). DOI: 10.1007/s10040-014-1200-7.
16. Coulibaly, P., Anctil, F., Aravena, R., Bobée, B.: Artificial Neural Network Modeling of Water Table Depth Fluctuations. Water Resources Research, vol. 37, no. 4, pp. 885–896 (2001). DOI: 10.1029/2000wr900368.
17. Han, J., Kamber, M.: Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan Kaufmann (2006)
18. Steyn, M.: Short-term Stream Flow Forecasting and Downstream Gap Infilling Using Machine Learning Techniques (2018)
19. Uc-Castillo, J.L., Marín-Celestino, A.E., Martínez-Cruz, D.A., Tuxpan-Vargas, J., Ramos-Leal, J.A.: A Systematic Review and Meta-analysis of Groundwater Level Forecasting with Machine Learning Techniques: Current Status and Future Directions. Environmental Modelling & Software, vol. 168, pp. 105788 (2023). DOI: 10.1016/j.envsoft.2023.105788.
20. Daliakopoulos, I.N., Coulibaly, P., Tsanis, I.K.: Groundwater Level Forecasting Using Artificial Neural Networks. Journal of Hydrology, vol. 309, no. 1–4, pp. 229–240 (2005). DOI: 10.1016/j.jhydrol.2004.12.001.
21. Kanyama, Y., Ajoodha, R., Seyler, H., Makondo, N., Tutu, H.: Application of Machine Learning Techniques in Forecasting Groundwater Levels in the Grootfontein Aquifer. In:

2nd International Multidisciplinary Information Technology and Engineering Conference, (IMITEC´20), pp. 1–8 (2020). DOI: 10.1109/imitec50163.2020.9334142.
22. Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka. In: Proceedings of the Data Mining and Knowledge Discovery Handbook, pp. 1305–1314 (2006). DOI: 10.1007/0-387-25465-X_62.
23. Gonzalez-Sanchez, A., Frausto-Solis, J., Ojeda-Bustamante, W.: Predictive Ability of Machine Learning Methods for Massive Crop Yield Prediction. Spanish Journal of Agricultural Research, vol. 12, no. 2, pp. 313–328 (2014). DOI: 10.5424/sjar/2014122-4439.
24. N., G., Jain, P., Choudhury, A., Dutta, P., Kalita, K., Barsocchi, P.: Random Forest Regression-based Machine Learning Model for Accurate Estimation of Fluid Flow in Curved Pipes. Processes, vol. 9, no. 11, pp. 2095 (2021). DOI: 10.3390/pr9112095.
25. Wang, L., Zhou, X., Zhu, X., Dong, Z., Guo, W.: Estimation of Biomass in wheat Using Random Forest Regression Algorithm and Remote Sensing Data. The Crop Journal, vol. 4, no. 3, pp. 212–219 (2016). DOI: 10.1016/j.cj.2016.01.008.
26. Probst, P., Wright, M.N., Boulesteix, A.: Hyperparameters and Tuning Strategies for Random Forest. WIREs Data Mining and Knowledge Discovery, vol. 9, no. 3 (2019). DOI: 10.1002/widm.1301.
27. Vapnik, V., Golowich, S.E., Smola, A.: Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: Proceedings of the Advances in Neural Information Processing Systems (1997)
28. Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K.: Improvements to the SMO Algorithm for SVM Regression. IEEE Transactions on Neural Networks, vol. 11, no. 5, pp. 1188–1193 (2000). DOI: 10.1109/72.870050.
29. Maimon, O., Rokach, L.: Introduction to Knowledge Discovery and Data Mining. In: Proceedings of the Data Mining and Knowledge Discovery Handbook, pp. 1–15 (2009). DOI: 10.1007/978-0-387-09823-4_1.
30. Almuhaylan, M.R., Ghumman, A.R., Al-Salamah, I.S., Ahmad, A., Ghazaw, Y.M., Haider, H., Shafiquzzaman, M.: Evaluating the Impacts of Pumping on Aquifer Depletion in Arid Regions Using Modflow, Anfis and Ann. Water, vol. 12, no. 8, pp. 2297 (2020). DOI: 10.3390/w12082297.
31. Fang, P., Zhang, X., Wei, P., Wang, Y., Zhang, H., Liu, F., Zhao, J.: The Classification Performance and Mechanism of Machine Learning Algorithms in Winter wheat Mapping Using Sentinel-2 10 m Resolution Imagery. Applied Sciences, vol. 10, no. 15, pp. 5075 (2020). DOI: 10.3390/app10155075.

ISSN 1870-4069

Indicador de calidad del agua para acuacultura utilizando una memoria asociativa modificada

Raúl Jiménez Cruz[1], Midory Esmeralda Vigueras Velazquez[2], Miguel González Mendoza[1]

[1] Instituto Tecnológico de Estudios Superiores, Estado de México, México

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

{rjimenezc, mgonza}@tec.mx, midory.vigueras@gmail.com

Resumen. El Chirostoma estor estor (conocido coloquialmente como charal blanco) es una especie muy importante que habita en el Lago de Pátzcuaro, ubicado en Michoacán, México. Este trabajo propone el uso de una memoria asociativa modificada para evaluar la calidad del agua en estanques de cultivo intensivo con sistemas de fotoperiodo. El fotoperiodo mejora las tasas de reproducción debido a que se controla la puesta de huevos de los peces. Se estudiaron la temperatura, el pH, el oxígeno disuelto, el amoníaco no ionizado, el amoníaco total, la alcalinidad total, los sólidos suspendidos totales, los fluoruros totales, la dureza total y los coliformes fecales. Una memoria hetero asociativa tiene la funcionalidad de recordar etiquetas que pertenecen a un patrón en el conjunto de datos. En otras palabras, realiza una clasificación de patrones. Los resultados experimentales muestran un buen rendimiento del modelo propuesto en comparación con los reportados en la literatura, evitando errores de medición y proporcionando una nueva herramienta para la investigación en acuacultura.

Palabras clave: Memoria asociativa, clasificador, reconocimiento de patrones.

Water Quality Indicator for Aquaculture Using a Modified Associative Memory

Abstract. The Chirostoma estor estor (commonly known as whitefish) is a highly important species inhabiting Pátzcuaro lake, located in Michoacán, Mexico. This paper proposes the use of a modified associative memory to evaluate water quality in intensive cultivation ponds with photoperiod systems. The photoperiod enhances reproduction rates by controlling the fish's egg laying. Temperature, pH, dissolved oxygen, un-ionized ammonia, total ammonia, total alkalinity, total suspended solids, total fluorides, total hardness, and fecal coliforms were studied. A heteroassociative memory has the functionality of remembering labels that belong to a pattern in the dataset. In other words, it performs pattern classification. Experimental results show good performance of the proposed

model compared to those reported in the literature, avoiding measurement errors and providing a new tool for research in aquaculture.

Keywords: Associative memory, classifier, pattern recognition.

1. Introducción

El Lago de Pátzcuaro, ubicado en Michoacán, México, se encuentra a 63 km al oeste de la ciudad de Morelia. Este es uno de los atractivos turísticos más importantes de la entidad, formando parte de un conjunto arqueológico, histórico, recreativo y cultural. El turismo representa la mayor fuente de ingresos para los habitantes. En este lago se encuentran 12 especies de peces diferentes: ocho son endémicas y cuatro son introducidas. Las dos especies más importantes en este lugar son el Chirostoma estor estor y Algansea lacustris. Ambas especies generan altos ingresos para la población debido a su alto volumen de pesca y demanda en el mercado regional.

El fotoperiodo en la acuacultura se utiliza para modificar el ciclo reproductivo, mejorar la sincronización de la maduración sexual e inducir la puesta, solucionando así problemas de baja población. En la literatura, se han realizado varios trabajos en los últimos años que han contribuido a comprender los efectos de la luz en la pesca, lo que ha generado avances tecnológicos considerables en la acuacultura.

Por ejemplo, en [2] se propuso un sistema informatizado de monitoreo y control ambiental donde un sistema supervisa y controla el fotoperiodo y la temperatura, proporcionando un registro continuo de estos parámetros en formato digital. Además, en [3] se propuso el uso de un sistema de apoyo a la toma de decisiones para la investigación y gestión en acuacultura. Sin embargo, estos trabajos carecen de metodologías de evaluación precisa y desarrollo tecnológico para un estudio más profundo de la especie Chirostoma estor estor y para un proceso de fotoperiodo exitoso. Por lo que se propone aplicar un clasificador inteligente para ayudar a la clasificación del agua para esta especie de pez.

2. Requisitos ambientales

Para poder obtener un dataset que pueda ser utilizado es necesario que se realicen análisis semanales de nitratos, que incluyen amoníaco no ionizado y amoníaco total. La demanda bioquímica de oxígeno, la demanda química de oxígeno, la temperatura y el pH se monitorearon diariamente debido a que representan los parámetros más críticos según expertos en la acuacultura de peces blancos (véase Tabla 1). Es importante destacar que otros parámetros no fueron considerados debido a que los sistemas cultivados están controlados ecológicamente y rara vez presentan niveles problemáticos. Por lo tanto, un conjunto reducido de los parámetros más importantes puede ser modelado con éxito para proporcionar un indicador preciso y viable que pueda ser medido en un sistema computacional. La importancia de los parámetros involucrados en la cría de peces blancos se detalla en la siguiente lista:

1. La Demanda Bioquímica de Oxígeno (**DBO**): Muestra la calidad del agua desde el punto de vista orgánico, ya que mide la cantidad de oxígeno consumido por

Tabla 1. Parámetros de calidad del agua utilizados para evaluar el hábitat del pez blanco.

Parámetros Críticos	**Por requerimiento**
Demanda Bioquímica de Oxígeno (DBO)	Amoníaco Total (TAN)
Demanda Química de Oxígeno (DQO)	Dureza Total (TOT_HAR)
Temperatura (TEMP)	Alcalinidad Total (TOT_ALC)
pH	Sólidos Suspendidos Totales (TSS)

microorganismos en la oxidación química de la materia orgánica presente en la muestra de agua [4].

2. La Demanda Química de Oxígeno (**DQO**): Es una medida de la cantidad de oxígeno disuelto consumido, bajo condiciones preestablecidas, por la oxidación química de la materia orgánica biodegradable presente en el agua [5].

3. Los parámetros de DBO y DQO: Proporcionan información diferente. Normalmente, los valores de DQO son más altos que los valores de DBO, porque el oxidante químico es capaz de reaccionar con sustancias que son difíciles de biodegradar para los microorganismos [6].

4. La temperatura (**TEMP**): Esta propiedad termodinámica afecta considerablemente las características físicas, químicas y biológicas de los cuerpos de agua. Controla el desove, la incubación (etapa más crítica en el desarrollo del pez blanco); regula actividad o suprime las tasas de crecimiento y puede ser letal en condiciones extremas. La especie resiste un amplio rango de temperatura de 15 °C a 24 °C. Las tasas de crecimiento se verán afectadas en temperaturas por debajo de los 15 °C. La temperatura óptima para el desarrollo de la especie se considera que está en 22°C [7].

5. **pH**: Este parámetro está determinado por el número de iones de hidrógeno libres (H+) y nos ayuda a indicar los niveles de acidez o alcalinidad en el estanque cultivado. El valor de pH ideal para el pez blanco es en un agua completamente neutra, lo que significa un valor exacto de 7. Sin embargo, el pH puede oscilar entre 7 y 8.5. Se puede observar que la especie no acepta un amplio rango en el pH, ya que el pez blanco estaría sujeto a estrés, lo que provoca altas tasas de mortalidad y problemas de amoníaco no ionizado [8].

2.1. Niveles de parámetros

La estabilidad de los parámetros de calidad del agua es importante en cualquier sistema de acuicultura. Las variaciones en sus niveles ocurren con frecuencia a lo largo del día y medirlos permite entender la dinámica del ambiente. Los parámetros diarios son extremadamente importantes porque pueden desestabilizar considerablemente el hábitat cultivado. La Tabla 2 muestra los niveles permitidos donde el hábitat del pez blanco puede considerarse óptimo para un buen proceso de cultivo.

Tabla 2. C_l y C_u corresponden a los límites inferior y superior definidos respectivamente, mientras que *m* es el nivel óptimo y deseable para la especie.

Parámetros	Unidades	Rango	Límites Permitidos		
			C_l	C_u	m
TEMP	Celsius	15 – 24	15	24	19.5
DO	mg/l	4.5 – 9	4.5	---	6
pH	Unidad pH	7 – 8.5	7	8.5	7.75
TAN	mg/l	0.1 – 1.0	0.1	1.0	0.55
TOT_HAR	mg/l	80-120	80	120	100
TOT_ALC	mg/l	90-110	90	110	100
TSS	mg/l	50-100	50	100	75
FEC_COL	mg/l	50-200	50	200	189.5
TOT_FLUO	mg/l	0.8-1.1	0.5	1.5	1.0

3. Modelos de aprendizaje automático

3.1. Linear Associator

El modelo de clasificación consiste en el uso del Linear Associator implementándole una técnica de ortonormalización de la matriz de entrenamiento para reducir el inconveniente principal del modelo original.

El Linear Associator, es una memoria asociativa, cuyo desarrollo se atribuye a dos científicos Kohonen en Finlandia [9] y Anderson en Estados Unidos [10] en el año de 1972. Pero este modelo tiene un gran problema, requiere vectores ortonormales para funcionar correctamente, lo cual no existe en los datasets de uso habitual.

3.2. Linear Associator con SVD

En el año 2021 se creó una nueva variante de este modelo el clasificador LA-SVD para más información revisar la referencia [11], A grandes rasgos esta nueva variante del modelo original propone dos ideas muy importantes, la primera es que al realizar la predicción de los patrones el modelo suele regresar valores con punto decimal lo cual está mal, pero si las etiquetas de clase son catalogadas en números ya sea como 1, 2, 3… y así sucesivamente.

Ahora bien, el modelo rara vez dará con la etiqueta correcta debido a que es muy importante que la matriz entrenada sea ortonormal por lo que tendera a clasificar mal los patrones. Por lo que la primera propuesta consiste en aplicar un redondeo en el momento que el clasificador obtenga la pseudo etiqueta de clase.

Ya que los valores que se obtienen al final son valores aproximados numéricamente a la etiqueta de clase. La segunda propuesta que se implementó en este modelo es el uso de la Descomposición en Valores Singulares o SVD en inglés. Esta técnica puede ayudar a obtener en la matriz de entrenamiento valores aproximadamente cercanos a los ortonormales de la matriz. Por lo que mejoraría la clasificación de los patrones al ser implementado.

El algoritmo con la SVD consiste en lo siguiente: Una vez elegido el método de validación comenzaremos tomando el conjunto de entrenamiento para entrenar la memoria asociativa. Se descompondrá el conjunto de entrenamiento usando la descomposición en valores singulares reducida o SVD economy los cual nos otorgará tres matrices, U, Σ y V. Esto debido a que la matriz U es una matriz que contiene valores aproximados a los ortonormales en una matriz.

El siguiente paso consiste en aplicar la fase de entrenamiento del linear Associator es decir se va a realizar un producto externo de la siguiente manera:

$$M = \sum_{\mu=1}^{p} y^{\mu}.(x^{\mu})^{T}, \tag{1}$$

donde:

- M es la matriz entrenada.
- y^{μ} son el vector de etiquetas de clase es decir la codificación de los patrones de salida.
- x^{μ} son los patrones que se encuentran en el conjunto de entrenamiento, pero transpuestos.

Una vez obtenida la matriz M se finaliza la fase de entrenamiento del modelo. El siguiente paso es la fase de recuperación la cual consiste en tomar el conjunto de entrenamiento y transformarlo en un conjunto ortonormal de la siguiente manera:

$$x_{\text{test\_new}} = x_{\text{test}} \cdot (V^{+})^{T} \cdot S^{+}, \tag{2}$$

donde:

- $x_{\text{test\_new}}$ es el conjunto de prueba transformado.
- x_{test} es el conjunto de entrenamiento original.
- $(V^{+})^{T}$ es una de las matrices obtenidas en la SVD antes de entrenar se le aplico la pseudo inversa de Moore Penrose además de la transpuesta.
- S^{+} es una de las matrices ob tenidas en la SVD antes de entrenar se le aplico la pseudo inversa de Moore Penrose.

Una vez transformado el conjunto de prueba nótese que jamás interactúa con el conjunto de entrenamiento se aplica la fase de recuperación del linear Associator la cual consiste en lo siguiente:

$$\text{round}(M \cdot x_{\text{test\_new}}). \tag{3}$$

Al aplicar el redondeo se obtendrá la etiqueta de clase predicha.

4. Descripción del dataset

El conjunto de datos que se analizó en el presente trabajo fue proporcionado por el repositorio de datos de CONAGUA [12], que desafortunadamente contenía muchos valores faltantes pero se le aplico un preprocesamiento de imputación para que este fuese más fácil de implementar. La base de datos de CONAGUA tiene las siguientes

Tabla 3. Accuracy.

SMO	Naive Bayes	Perceptrón Multicapa	Bosques Aleatorios	LA SVD	LA SVD Polar
94.96	92.44	95.68	95.68	96.55	94.96

Tabla 4. Balance accuracy.

SMO	Naive Bayes	Perceptrón Multicapa	Bosques Aleatorios	LA SVD	LA SVD Polar
0.9520	0.9315	0.9610	0.9590	0.9619	0.9478

características: Contiene 1390 patrones y diez características, todas numéricas, que son las siguientes: TEMP, pH, DBO, DQO, TOT_ALC, TAN, TSS, TOT_FLUO, TOT_HAR y FEC_COL.

Las clases pertenecientes a esta base de datos corresponden a un semáforo que CONAGUA estableció por medio de la RENAMECA, de la siguiente manera:

- **Verde**: El agua del sitio se encuentra dentro de los rangos de calidad excelente, buena calidad y aceptable, en todos los indicadores. Esta clase tiene 564 patrones.
- **Rojo**: El agua del sitio está contaminada con Demanda Bioquímica de Oxígeno (DBO) o con Demanda Química de Oxígeno (DQO) y además el agua del sitio está contaminada con Sólidos Suspendidos Totales (TSS). Esta clase tiene 826 patrones.

Como se puede observar el dataset se encuentra balanceado ya que tiene un IR (Imbalance Ratio) [13] de 1.46, siendo la clase Verde la clase minoritaria y la clase Rojo la mayoritaria. Cabe aclarar que este dataset extraído es para todo el país, con alrededor de 5000 datos y en este caso se aplicó un filtro en los rasgos que no usaban las características necesarias para un cultivo de peces Chirostoma, así como un filtro en los patrones ya que el caso de estudio fue para el lago de Pátzcuaro donde se realiza acuacultura del Chirostoma reduciéndolo a 1390 patrones que representan zonas de acuacultura, además los datos son del año 2016 pero aplicando el mismo filtro se puede implementar con datos más actuales. La razón de usar un dataset del 2016 es porque se encuentra más completo que los actuales.

5. Resultados experimentales y discusión

Como el conjunto de datos maneja datos completamente numéricos, se utilizó un método de imputación, que consiste en obtener la media de cada característica para cada clase del conjunto de datos y así eliminar la problemática de los valores perdidos. Posteriormente, se programó el algoritmo LA-SVD y a su vez se le realizó una pequeña variante donde se implementa un cambio de coordenadas cartesianas a esféricas para n dimensiones, esto con el fin de cambiar el espacio muestral bajo la hipótesis de que cada patrón puede alejarse o acercar de la clase a la que pertenece, todo esto se realizó utilizando las siguientes ecuaciones:

$$x_n = r \cos \alpha_{n-1} \prod_{n=1}^{n-2} \sin \alpha_n, \tag{4}$$

donde x_n es el enésimo patrón, r es el radio de una esfera n dimensional y α_n es el enésimo ángulo del plano esférico en n dimensiones. Para entender la siguiente ecuación es necesario segmentarla en partes como sigue:

$$r = \sqrt{\sum_{n=1}^{n}}. \tag{5}$$

La ecuación 2 se refiere al cálculo del radio n-dimensional de la esfera. Las siguientes ecuaciones son útiles para determinar el ángulo alfa (α) de los $n-1$ componentes:

$$\alpha_1 = \cos^{-1}\left(\frac{x_1}{\sqrt{x_n^2 + x_{n-1}^2 + \cdots + x_1^2}}\right), \tag{6}$$

$$\alpha_3 = \cos^{-1}\left(\frac{x_2}{\sqrt{x_n^2 + x_{n-1}^2 + \cdots + x_2^2}}\right), \tag{7}$$

$$\alpha_{n-2} = \cos^{-1}\left(\frac{x_{n-2}}{\sqrt{x_n^2 + x_{n-1}^2 + x_{n-2}^2}}\right), \tag{8}$$

$$\alpha_{n-1} = \cos^{-1}\left(\frac{x_{n-1}}{\sqrt{x_n^2 + x_{n-1}^2}}\right). \tag{9}$$

Una vez realizado este cambio, se le aplicó el método de validación leave-one-out para este clasificador con la razón de que los resultados son deterministas al implementarlo por lo que no hay variabilidad en los mismos. Finalmente, se implementó el clasificador LA SVD con coordenadas esféricas, el cual mostró un porcentaje de accuracy del 81.22%, pero debido a que el data set se encuentra desbalanceado se procedió también a usar la medida de desempeño balance accuracy la cual dio un resultado de 96.55%.

El siguiente paso fue comparar el resultado del clasificador propuesto con algunos clasificadores del estado del arte utilizando la plataforma WEKA. Los resultados de precisión se muestran en la Tabla 3. Debido a que el conjunto de datos está desequilibrado, es necesario analizar la matriz de confusión y verificar que no haya sesgo en las clases con más patrones:

- **SMO**: A pesar de tener un buen accuracy, clasificó correctamente 564 patrones en la primera clase es decir todos, pero solo clasificó 771 de 826 en la segunda clase.
- **Naive Bayes**: Para la primera clase solo se clasificaron correctamente 546 patrones de 564, para la segunda clase se clasificaron correctamente 739 patrones de los 826.
- **Perceptrón multicapa**: La primera clase obtuvo 555 patrones bien clasificados de 564. La segunda clase obtuvo 7775 patrones bien clasificados.

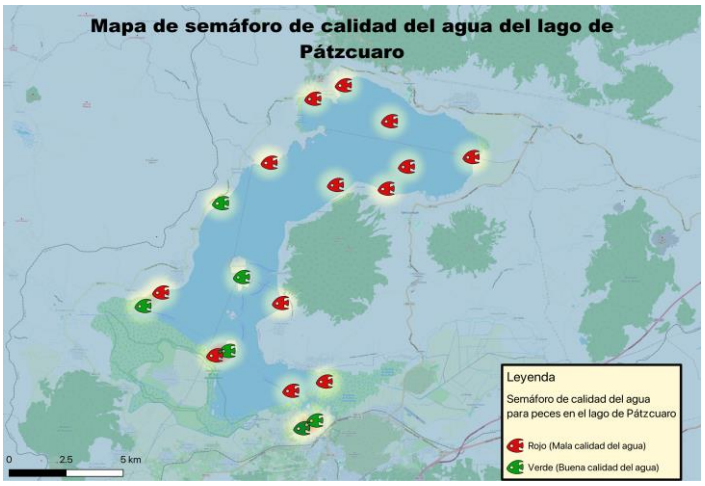


Fig. 1. Mapa que representa en un semáforo la calidad del agua en el Lago de Pátzcuaro.

- **Bosques Aleatorios**: Obtuvo 547 aciertos en la primera clase y 783 en la segunda clase.
- **LA SVD**: El algoritmo propuesto obtuvo 532 patrones bien clasificados en la primera clase, la segunda clase obtuvo 810 patrones bien clasificados.
- **LA SVD Polar**: La variante del algoritmo obtuvo 529 patrones bien clasificados para la primera clase, mientras que en la segunda clase se obtuvieron 791 patrones bien clasificados.

Debido a que se notaron algunos sesgos en la clasificación se procedió a realizar el cálculo del Balance Accuracy para verificar si existe dicho sesgo esto puede verse en la Tabla 3. Accuracy.

SMO	Naive Bayes	Perceptrón Multicapa	Bosques Aleatorios	LA SVD	LA SVD Polar
94.96	92.44	95.68	95.68	96.55	94.96

Tabla *4*. Finalmente, como puede apreciarse en la tabla anterior el modelo más estable es el LA SVD en cuanto a una clasificación más balanceada de los patrones de este dataset además de ser el mejor en accuracy.

Posteriormente la clasificación puede ser representada en un mapa debido a que los datos de Conagua tienen asociados cada punto un par de coordenadas por lo que una vez clasificado se puede mostrar en un sistema de información geoespacial (véase Fig. **1**). La figura anterior es un ejemplo de cómo expandir este proyecto a más que clasificar donde podemos ver lugares donde el agua tiene una mala calidad y que se puedan atender de forma inmediata en cuanto se detecte un estanque con calidad mala.

6. Conclusiones

Este trabajo se desarrolló con el fin de crear una herramienta especializada para monitorear los estanques de Chirostoma estor estor, la cual es una especie endémica

mexicana en peligro de extinción. Se debe destacar que su principal ventaja radica en el análisis de los parámetros específicos cruciales para su reproducción.

El uso de un clasificador se revela como un elemento fundamental para evaluar la calidad del agua necesaria para asegurar el bienestar de este ejemplar en su hábitat, es decir, se puede tomar como un recurso esencial para la gestión y mejora del ecosistema acuático en caso de ser necesario. Como trabajo futuro, se planea fortalecer el modelo de evaluación, incorporando más parámetros para una evaluación exhaustiva del fotoperíodo, lo que potencialmente mejoraría la reproducción de la especie. Además, se considera la implementación de nuevas técnicas de preprocesamiento que minimicen el sesgo en la clasificación, asegurando una evaluación más precisa de la calidad del agua.

Referencias

1. Martínez Palacios, C.A., Toledo Cuevas, M.E.C., Blanco Michoacán. *Rev. Digit. Univ.*, vol. 10, pp. 42–44 (2005)
2. Fabbrocini, A., Di-Stasio, M., D'Adamo, R.: Computerized Sperm Motility Analysis in Toxicity Bioassays: A New Approach to Pore Water Quality Assessment. Ecotoxicology and Environmental Safety, vol. 73, no. 7, pp. 1588–1595 (2010). DOI: 10.1016/j.ecoenv. 2010.05.003.
3. Mølmann, J.A., Steindal, A.L., Bengtsson, G.B., Seljåsen, R., Lea, P., Skaret, J., Johansen, T.J.: Effects of Temperature and Photoperiod on Sensory Quality and Contents of Glucosinolates, Flavonols and Vitamin C in Broccoli Florets. Food Chemistry, vol. 172, pp. 47–55 (2015) DOI: 10.1016/j.foodchem.2014.09.015.
4. Tchinda, D., Henkanatte-Gedera, S., Abeysiriwardana-Arachchige, I., Delanka-Pedige, H., Munasinghe-Arachchige, S., Zhang, Y., Nirmalakhandan, N.: Single-step Treatment of Primary Effluent by Galdieria Sulphuraria: Removal of Biochemical Oxygen Demand, Nutrients, and Pathogens. Algal Research, vol. 42, pp. 101578 (2019). DOI: 10.1016/j.algal.2019.101578.
5. Kabir, H., Zhu, H., Lopez, R., Nicholas, N.W., McIlroy, D.N., Echeverria, E., May, J., Cheng, I.F.: Electrochemical Determination of Chemical Oxygen Demand on Functionalized Pseudo-Graphite Electrode. Journal of Electroanalytical Chemistry, vol. 851, pp. 113448 (2019). DOI: 10.1016/j.jelechem.2019.113448.
6. Nguyen, L.A.T., Ward, A.J., Lewis, D.: Utilisation of Turbidity as an Indicator for Biochemical and Chemical Oxygen Demand. Journal of Water Process Engineering, vol. 4, pp. 137–142 (2014). DOI: 10.1016/j.jwpe.2014.09.009.
7. Zheng, G., Bao, A., Li, J., Zhang, G., Xie, H., Guo, H., Jiang, L., Chen, T., Chang, C., Chen, W.: Sustained Growth of High Mountain Lakes in the Headwaters of the Syr Darya River, Central Asia. Global and Planetary Change, vol. 176, pp. 84–99 (2019). DOI: 10.1016/j.gloplacha.2019.03.004.
8. Qin, Y., Alam, A. U., Pan, S., Howlader, M.M., Ghosh, R., Hu, N., Jin, H., Dong, S., Chen, C., Deen, M.J.: Integrated Water Quality Monitoring System with ph, Free Chlorine, and Temperature Sensors. Sensors and Actuators B: Chemical, vol. 255, pp. 781–790 (2018). DOI: 10.1016/j.snb.2017.07.188.
9. Kohonen, T.: Correlation Matrix Memories. IEEE Transactions on Computers, vol. C–21, no. 4, pp. 353–359 (1972). DOI: 10.1109/tc.1972.5008975.
10. Anderson, J.A.: A Simple Neural Network Generating an Interactive Memory. Mathematical Biosciences, vol. 14, no. 3–4, pp. 197–220 (1972). DOI: 10.1016/0025-5564(72)90075-2.
11. Jiménez-Cruz, R., Velázquez-Rodríguez, J., López-Yáñez, I., Villuendas-Rey, Y., Yáñez-Márquez, C.: Supervised Classification of Diseases based on an Improved Associative Algorithm. Mathematics, vol. 9, no. 13, pp. 1458 (2021). DOI: 10.3390/math9131458.

12. CONAGUA: Indicadores de la calidad del agua superficial y subterránea. Red nacional de medición de la calidad del agua. http://files.conagua.gob.mx/Ica20/Contenido/Documentos/PresentaciondeIndicadoresdelaCalidaddelAgua.pdf (2022)
13. KEEL: KEEL-dataset, data set repository. http://sci2s.ugr.es/keel/imbalanced.php (2005)

ISSN 1870-4069

Fuzzy Control of a Self-Balancing System: An Approach for Satellite Attitude Determination and Control System Testbed

A. de J. Pablo-Sotelo[1,2], María Elena Aguilar-Jáuregui[3], A. Luviano Juárez[1], Cuauhtemoc Peredo-Macías[3], J. J. Hernández Gómez[2]

[1] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
Ciudad de México, Mexico

[2] Instituto Politécnico Nacional, Centro de Desarrollo Aeroespacial,
Ciudad de México, Mexico

[3] Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, Mexico

abraham_pablo_11@hotmail.com

Abstract. In the development of satellite systems, rigorous validation of constituent subsystems is imperative. Among the various subsystems that compose a satellite, the Attitude Determination and Control Sys- tem (ADCS) plays a crucial role in maintaining satellite orientation and stability. The validation process for these subsystems traditionally employs test benches capable of simulating space environment conditions. However, simulating such an environment presents numerous challenges, a significant one being the imbalance caused by the discrepancy between the satellite's center of mass and its geometric center, which can substantially affect test conditions and results. This work presents a simplified design of a single-axis balancing system for a 1U CubeSat ADCS testbed. The system utilizes a fuzzy controller designed to operate a sliding mass, which corrects internal system perturbations for the initial configurations of satellite tests.

Keywords: Fuzzy control, satellite control, attitude determination and control system, self-balancing, hardware-in-the-loop test.

1 Introduction

Attitude Determination and Control System (ADCS) is a critical subsystem responsible for managing a satellite's orientation in space. It plays a crucial role in ensuring the success of space missions. The ADCS performs several vital functions, such as orienting solar panels to maximize energy collection, aligning satellite antennas with ground stations to facilitate communication, and directing scientific instruments towards specific celestial bodies or regions of interest. In the context of CubeSats, a class of small satellites, the implementation of ADCS

varies. While not all CubeSats incorporate an ADCS, the majority do include this subsystem, with only a few exceptions. The decision to include an ADCS in a CubeSat depends on the specific mission objectives, power constraints, and complexity of the satellite.

1.1 Verification of Attitude Determination and Control Systems of CubeSats

Satellite orientation and stability control systems require rigorous testing before deployment. To address this problem, advanced testing equipment has been created that simulates the conditions in space [13, 15, 17, 19, 24].These platforms, integrate various mechanical, electrical, and control components to recreate the challenges a satellite will face in space. The primary function of these testbeds is to evaluate the effectiveness of a satellite's ADCS. By simulating space conditions, these platforms enable to:

1. Assess the ADCS performance,
2. Detect potential malfunctions,
3. Implement necessary adjustments.

This process is crucial for ensuring the satellite's proper functioning once in orbit. For CubeSats test benches these typically incorporate three key elements [14]:

- An air bearing system for frictionless rotation,
- A mechanism to generate simulated disturbances,
- A Helmholtz cage for magnetic field simulation.

The versatility of these test benches allows for a range of verification procedures. These are generally categorized into two main types of testing:

- Hardware-in-the-Loop (HIL) Test: Evaluates real hardware components in a simulated environment, combining physical and virtual elements to test system performance under realistic conditions [1, 21, 23].
- Software-in-the-Loop (SIL) Test: Assesses control algorithms in a fully virtual environment, allowing for rapid iteration and debugging of software without physical hardware constraints [6, 7, 10].

One of the inherent challenges in developing testbeds for ADCS systems is balancing the testbed itself, as discussed in [4]. The testbed must initially perform a balancing procedure to establish the initial test conditions. While various solutions have been proposed in the literature, the most widely accepted method is manual balancing [9, 12, 20]. However, this approach is inherently susceptible to multiple human-related issues.

1.2 Fuzzy Control

Fuzzy control, based on fuzzy logic introduced by Zadeh in 1965 [27], is a control and decision-making approach that allows working with imprecise and vague information, similar to human reasoning.

This approach is fundamental in situations where strict data precision is not possible or necessary, and a more flexible and adaptable interpretation of information is required. Fuzzy sets employ membership functions that assign a degree of membership (usually between 0 and 1) to each element of the set, allowing for the representation of vague or imprecise concepts [16 , 26].

Unlike conventional control methods, fuzzy control does not require a precise mathematical model of the system, making it particularly useful for nonlinear systems with a high degree of uncertainty [18, 22].

This approach allows for the incorporation of expert knowledge in the form of linguistic rules, facilitating the implementation of control strategies based on human experience. Furthermore, fuzzy control can efficiently manage multiple input and output variables, making it suitable for complex multivariable systems.

In the context of satellite test systems, specifically for ADCS, fuzzy control offers significant advantages, as demonstrated in [3, 5, 8]. The nonlinear nature of the balancing system (as shown in [4]), coupled with the need to handle multiple variables such as inclination angle and angular velocity, makes fuzzy control an attractive option.

The ability to incorporate expert knowledge about system behavior can lead to more robust and adaptable control.

Compared to classical control methods like PID, fuzzy control can offer better performance in nonlinear systems and may be easier to adjust in situations where the exact mathematical model of the system is difficult to obtain or changes over time. It is worth noting that the combination of classical controllers such as PID and fuzzy controllers has been extensively studied, as shown in the literature [2, 11, 25].

2 Fuzzy Controller

The following section presents a conceptual model that closely approximates a real ADCS testbed, addressing the single-axis balancing problems described in Section 1.1. This model serves as a foundation for understanding and analyzing the dynamics of the system in a controlled environment.

2.1 Description of the system

The system used to study mass balancing systems in ADCS testbeds is shown in Figure 1.

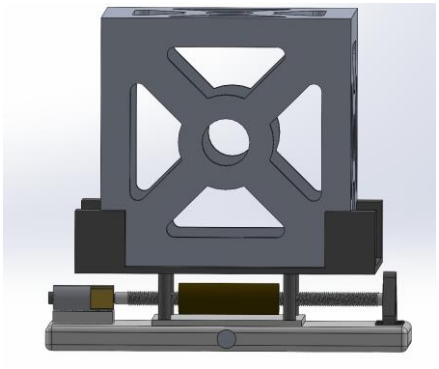


Fig. 1. A frontal view of the system with a 1U CubeSat mounted.

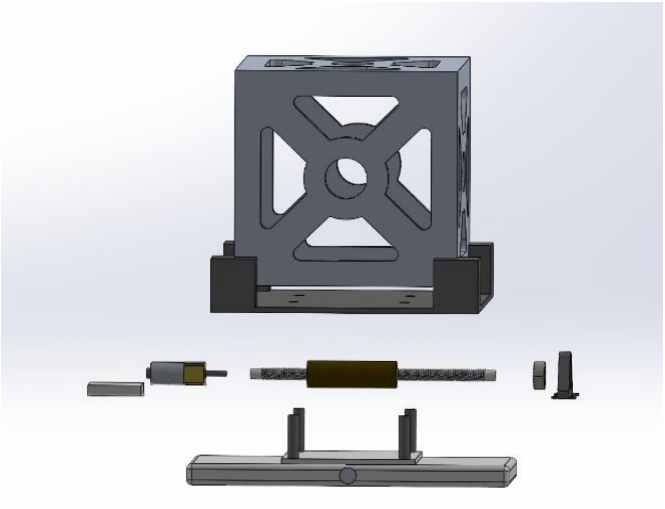


Fig. 2. Exploded view of the system.

It is important to note that while this design deviates from the characteristic configuration of manual balancing systems for verification purposes, it retains similar functional attributes.

Specifically, it incorporates a mass that traverses beneath the satellite, aiding in the compensation of the discrepancy between the center of mass and the geometric center of the CubeSat.

At this stage, it should be observed that low-friction environmental conditions have not been incorporated into the model. Figure 2 presents an exploded view of the system, offering a comprehensive visualization of all constituent components.

Table 1. System Components (see Figure 2)

Component	**Function**	**Technical Specification**
Lead Screw	Displaces the mass to correct centre of mass	M5-0.8×100 worm screw
Motor	Actuates the corrective mass	Pololu Micro Metal Gearmotor MP 6V with 12 CPR Encoder 298:1
Sliding Mass	31.46g corrective mass	Galvanised steel
Encoder	Measures system inclination via rotational displacement at the base pivot	5000-pulse incremental magnetic encoder

Table 2. Signals of the fuzzy system.

Name	**Type**	**Description**
Inclination Angle (θ)	Input	Represents the beam's inclination relative to an inertial frame located at the system's pivot, measured by an incremental encoder
Angular Velocity $\dot{\theta}$	Input	Rate of change of the inclination angle with respect to time t
Mass Displacement (μ_1)	Output	Required displacement to generate a balancing torque for the system

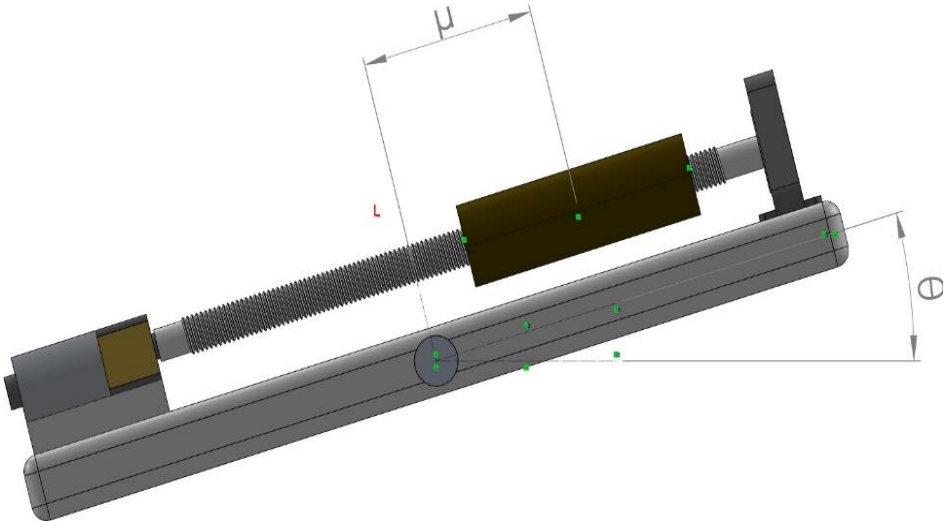


Fig. 3. System without CubeSat.

Table 3. Sets for the error variable.

Inclination Angle [degrees]	Set
-90 to -0.5	Very Low Error (*V LE*)
-1 to 0	Low Error (*LE*)
-0.5 to 0.5	Zero Error (*ZE*)
0 to 1	High Error (*HE*)
0.5 to 90	Very High Error (*V HE*)

Table 4. Sets for the angular velocity variable.

Angular velocity [degrees/sec]	Set
-2 to 0	Low Velocity (*LV*)
-0.5 to 0.5	Zero Velocity (*ZV*)
0 to 2	High Velocity (*HV*)

Table 5. Sets for the mass displacement variable.

Displacement [meters]	Set
-0.05 to -0.01	Very Low Position (VLP)
-0.02 to 0	Low Position (LP)
-0.01 to 0.01	Zero Position (ZP)
0 to 0.02	High Position (HP)
0.01 to 0.05	Very High Position (VHP)

Table 6. Control Rule Matrix.

		Angular Velocity		
		HV	ZV	LV
Error	VLE	HP	VHP	VHP
	LE	HP	HP	VHP
	ZE	LP	ZP	HP
	HE	LP	LP	VLP
	VHE	LP	VLP	VLP

Furthermore, Table 1 provides a detailed description of the most salient components, highlighting their roles and significance within the overall system architecture. Given the encoder specifications described in Table 1, the system achieves an angular resolution of 0.072°.

Additionally, the maximum displacement velocity of the mass, determined by the worm screw and motor characteristics, is approximately 0.001467 m/s. For the purposes of this study, Figure 3 shows the essential elements crucial to the design of the fuzzy controller.

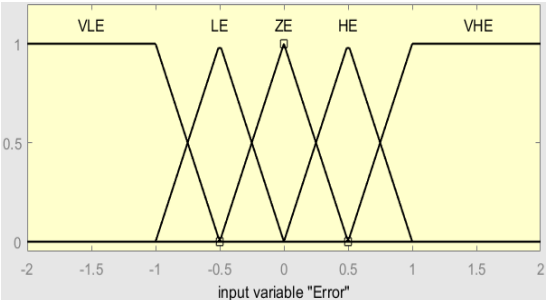


Fig. 4. Membership function of the error variable.

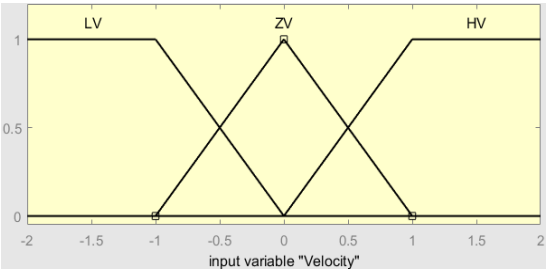


Fig. 5. Membership function of the angular velocity variable.

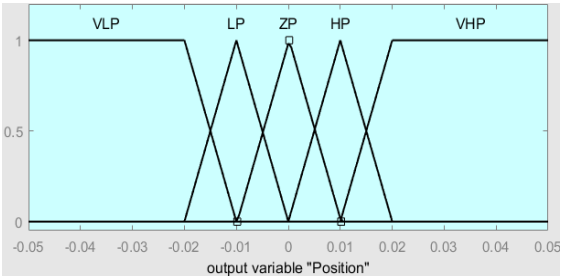


Fig. 6. Membership function of the mass displacement variable.

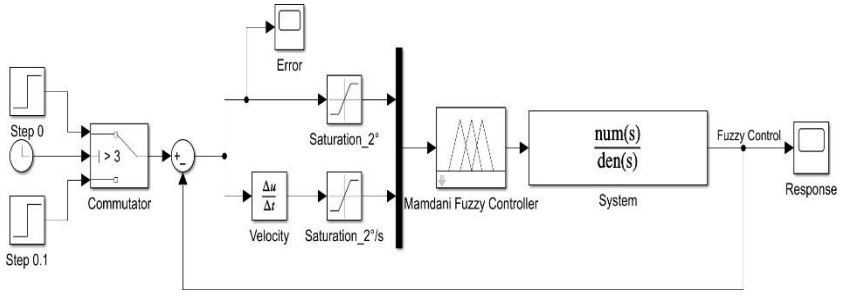


Fig. 7. System model and fuzzy control.

2.2 Fuzzy Control Design

The following section details the design of the fuzzy control system for the setup showed in Figure 1.

2.3 Linguistic Variables of the System

Table 2 presents the linguistic variables of the system, along with a concise description of their role within the fuzzy controller.

2.4 Functional Description of the System

The following presents a brief and simplified functional description (See Table 6 to observe the control behavior) of the system:

If the system's inclination is very high and this inclination is rapid, the mass displacement is high. If the system's inclination is very high and this inclination is slow, the mass displacement is low. If the system's inclination is very high and the velocity is zero, the mass displacement is low. If the system's inclination is very low and the velocity is very high, the displacement is high.

2.5 Definition of the Sets

The error variable is derived from the incremental encoder located at the system's pivot. For this linguistic variable, the universe of discourse is defined from -90° to 90°, and the sets are represented in Table 3.
The selection of sets for the angular velocity variable follows a similar approach to the previous set. However, it is described by the angular velocity in degrees per second, as shown in Table 4.

Finally, Table 5 describes the set for the mass displacement variable, which is defined by the dimensions of the lead screw.

2.6 Control Rule Sets

The control rules are entirely dependent on the experience of the control system designer. Due to the number of variables, a two-dimensional matrix is generated, derived from the functional description of the system. Given the behavior of the system's linguistic variables, it is possible to describe the fuzzy control through the matrix in Table 6.

Based on the control matrix, the following structure is used for the propositions: ***IF*** *premise 1* ***AND*** *premise 2* ***THEN*** *consequent.*

Here the consequent is the output variable (mass displacement variable). Below are all the compound propositions of the system:

If $E = VLE$ **and** $V = HV$ **then** $P = HP$
If $E = VLE$ **and** $V = ZV$ **then** $P = VHP$
If $E = VLE$ **and** $V = LV$ **then** $P = VHP$

If $E = LE$ **and** $V = HV$ **then** $P = HP$
If $E = LE$ **and** $V = ZV$ **then** $P = HP$
If $E = LE$ **and** $V = LV$ **then** $P = VHP$
If $E = ZE$ **and** $V = HV$ **then** $P = LP$
If $E = ZE$ **and** $V = ZV$ **then** $P = ZP$
If $E = ZE$ **and** $V = LV$ **then** $P = HP$
If $E = HE$ **and** $V = HV$ **then** $P = LP$
If $E = HE$ **and** $V = ZV$ **then** $P = LP$
If $E = HE$ **and** $V = LV$ **then** $P = VLP$
If $E = VHE$ **and** $V = HV$ **then** $P = LP$
If $E = VHE$ **and** $V = ZV$ **then** $P = VLP$
If $E = VHE$ **and** $V = LV$ **then** $P = VLP$

2.7 Membership Functions

Finally, the following membership functions are established for each set of linguistic variables in the system. Figure 4 shows the membership function corresponding to the error variable.

This distribution for the membership functions was constructed based on the operating ranges shown in the literature [9, 12, 20]. Note that the system's efficiency may vary depending on the type of function; this will be addressed in depth in the conclusions of this work. For the membership function of the angular velocity, only three velocities will be considered. Efficient results were shown in the simulations presented in Section 2.3 within the context of this work. Lastly, due to the dimensions of the mobile bar and the moment-generating mass, the following membership function is established (see Figure 6).

2.8 Simulations

The fuzzy controller is validated through numerical simulation using Matlab ® Simulink tool, version R2023b. The Simulink model of the fuzzy controller is shown in Figure 7.

The system response under these conditions is shown in the graph in Figure 8. Finally, the corresponding modification is made in the Simulink model to obtain the system's response to an initial condition of 2° and its response to a reference of 0°, which is showed in Figure 9.

3 Results and Conclusions

The fuzzy controller, as shown in Figure 8, was subjected to tracking two references (the same ones illustrated in the Simulink model in Figure 7). To reach the position of 0.1° inclination, the system achieved its settling time in approximately 1 second, and exhibited an overshoot of 0.024° for the 0.1° reference.

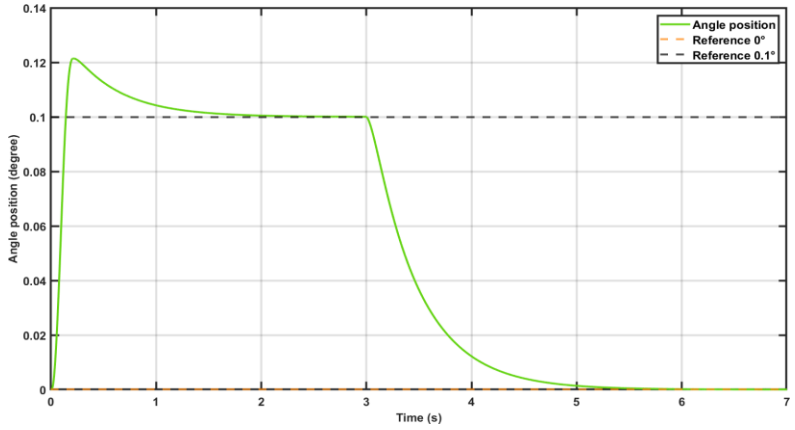


Fig. 8. Response of the fuzzy controller to references of 0.1° and 0°.

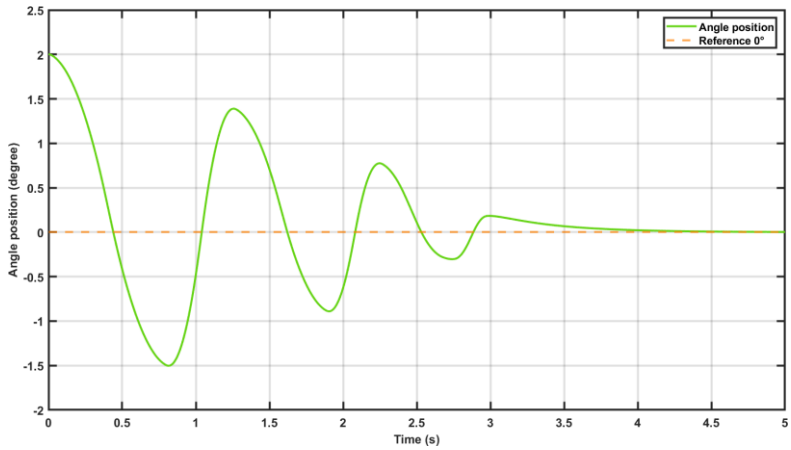


Fig. 9. Response of the fuzzy controller to reference of 0°.

On the other hand, to reach the 0° position starting from a 0.1° deviation, the settling time is achieved in approximately 1.2 seconds with an almost imperceptible overshoot of 0.001°. Finally, the case closest to the real application was presented, which starts from a system with a 2° deviation (see Figure 9). In this case, a settling time of 3.4 seconds was observed, and an overshoot above the 0° reference level of approximately 1.5°, which, although significantly large, does not affect the model's objectives.

The fuzzy control system was successfully designed for a single axis to balance an ADCS test system. This design effectively addresses the issues arising from human manipulation and calibration inherent in all manual systems presented in the literature. While not optimal, the performance metrics of the controller are sufficient to meet the requirements for balancing systems in ADCS testbed, considering the scope of this work. The system's linguistic variables, their sets, and membership functions were defined, along with the control rules. This approach allows for the incorporation of expert knowledge and the handling of system nonlinearities. Future work could

explore the optimization of the fuzzy controller using techniques such as genetic algorithms to further enhance its performance. Additionally, we could consider systems with a greater number of membership functions or different geometries (e.g., sigmoidal) to potentially enhance system performance. However, it is important to avoid too many functions near 0° to prevent exceeding the capabilities of the selected encoder. These improvements might help reduce the overshoot and settling time, potentially leading to better system performance.

Acknowledgments. The authors acknowledge partial economical support by projects 20242752, 20240894, 20241163, 20241077 and 20240811, as well as EDI and PIFI grants, provided by Secretaría de Investigación y Posgrado, Instituto Politécnico Nacional.

References

1. Carletta, S., Teofilatto, P., Farissi, M.S.: A magnetometer-only attitude determination strategy for small satellites: Design of the algorithm and hardware-in-the-loop testing. Aerospace (2020). https://doi.org/10.3390/aerospace7010003
2. Carvajal, J., Chen, G., Öğmen, H.: Fuzzy PID controller: Design, performance evaluation, and stability analysis. Inf. Sci. 123, 249–270 (2000). https://doi.org/10.1016/S0020-0255(99)00127-9
3. Cheng, C.H., Shu, S.L., Cheng, P.: Attitude control of a satellite using fuzzy controllers. Expert Syst. Appl. 36, 6613–6620 (2009). https://doi.org/10.1016/j.eswa.2008.08.053
4. da Silva, R.C., Borges, R.A., Battistini, S., Cappelletti, C.: A review of balancing methods for satellite simulators. Acta Astronautica 187, 537–545 (2021). https://doi.org/10.1016/j.actaastro.2021.05.037
5. Dizadji, M.R., Yousefi-Koma, A., Gharehnazifam, Z.: 3-axis attitude control of satellite using adaptive direct fuzzy controller. In: 6th RSI International Conference on Robotics and Mechatronics (IcRoM) pp. 1–5 (2018). https://doi.org/10.1109/ICROM.2018.8657560
6. Gaber, K., El-Mashade, M., Aziz, G.A.A.: High-precision attitude determination and control system design and real-time verification for Cubesats. International Journal of Communication Systems 33 (2020). https://doi.org/10.1002/dac.4311
7. Garcia, C.B., Vale, S.R.C., Martins-Filho, L., Duarte, R.O., Kuga, H.K., Carrara, V.: Validation tests of attitude determination software for nanosatellite embedded systems. Measurement 116, 391–401 (2018). https://doi.org/10.1016/J.MEASUREMENT.2017.11.040
8. Guan, P., Liu, X., Liu, X.: Adaptive fuzzy control for satellite. In: 6th World Congress on Intelligent Control and Automation 1, 124–128 (2006). https://doi.org/10.1109/WCICA.2006.1712375
9. Jovanovic, N., Pearce, J.M., Praks, J.: Design and testing of a low-cost, open source, 3-d printed air-bearing-based attitude simulator for cubesat satellites. Journal of Small Satellites 8(2), 859–880 (2019)
10. Kiesbye, J., Messmann, D., Preisinger, M., Reina, G., Nagy, D., Schummer, F., Mostad, M., Kale, T., Langer, M.: Hardware-in-the-loop and software-in-the-loop testing of the move-ii cubesat. Aerospace (2019). https://doi.org/10.3390/aerospace6120130
11. Kim, J.H., Kim, K.C., Chong, E.: Fuzzy precompensated pid controllers. IEEE Trans. Control. Syst. Technol. 2, 406–411 (1994). https://doi.org/10.1109/87.338660

12. Kwan, T.H., Lee, K.M.B., Yan, J., Wu, X.: An air bearing table for satellite attitude control simulation. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA). pp. 1420–1425 (2015). https://doi.org/10.1109/ICIEA.2015.7334330
13. Lavezzi, G., Stang, N.J., Ciarcià, M.: Start: A satellite three axis rotation testbed. Micromachines 13(2) (2022). https://doi.org/10.3390/mi13020165
14. Ley, W., Wittmann, K., Hallmann, W., et al.: Handbook of Space Technology. Wiley Press, United Kingdom, 1 edn. (2009)
15. Meissner, D.: A Three Degrees of Freedom Test-Bed for Nanosatellite and CubeSat Attitude Dynamics, Determination, and Control. Master's thesis, Naval Postgraduate School, Monterey, California, United States (December 2009)
16. Mendel, J.: Fuzzy logic systems for engineering: a tutorial. Proc. IEEE 83, 345–377 (1995). https://doi.org/10.1109/5.364485
17. Mendoza-Bárcenas, M., Vicente-Vivas, E., Rodríguez-Cortés, H.: Mechatronic design, dynamic modeling and results of a satellite flight simulator for experimental validation of satellite attitude determination and control schemes in 3-axis. Journal of Applied Research and Technology 12(3), 370–383 (2014). https://doi.org/https://doi.org/10.1016/S1665-6423(14)71619-0
18. Nguyen, A., Taniguchi, T., Eciolaza, L., Campos, V.C.S., Palhares, R., Sugeno, M.: Fuzzy control systems: Past, present and future. IEEE Computational Intelligence Magazine 14, 56–68 (2019). https://doi.org/10.1109/MCI.2018.2881644
19. Papakonstantinou, C., Moraitis, G., Lappas, V., Kostopoulos, V.: Design of a low-cost air bearing testbed for nano cmg maneuvers. Aerospace 9(2) (2022). https://doi.org/10.3390/aerospace9020095
20. Saulnier, K., Pérez, D., Huang, R., Gallardo, D., Tilton, G., Bevilacqua, R.: A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft. Acta Astronautica 105, 444–462 (2014)
21. Shim, H., Kim, O., Park, M., Woo Choi, M., Kee, C.: Development of hardware-in-the-loop simulation for cubesat platform: Focusing on magnetometer and magnetorquer. IEEE Access 11, 73164–73179 (2023). https://doi.org/10.1109/ACCESS.2023.3294565
22. Sugeno, M.: An introductory survey of fuzzy control. Inf. Sci. 36, 59–83 (1985). https://doi.org/10.1016/0020-0255(85)90026-X
23. Tapsawat, W., Sangpet, T., Kuntanapreeda, S.: Development of a hardware-in-loop attitude control simulator for a cubesat satellite. IOP Conference Series: Materials Science and Engineering 297 (2018). https://doi.org/10.1088/1757- 899X/297/1/012010
24. Ustrzycki, T.: Spherical air bearing testbed for nanosatellite attitude control development. Master's thesis, York University, Toronto, Ontario (2011)
25. Xu, J.X., Hang, C.C., Liu, C.: Parallel structure and tuning of a fuzzy pid controller. Automatica 36(5), 673–684 (2000). https://doi.org/10.1016/S0005-1098(99)00192-2
26. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning - I. Inf. Sci. 8, 199–249 (1975). https://doi.org/10.1016/0020- 0255(75)90036-5
27. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965). https://doi.org/10.1016/S0019-9958(65)90241-X

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación