EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 151 No. 7
July 2022

Research in Computing Science

Research in Computing Science, Año 21, Volumen 151, No. 7, julio de 2022, 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 junio de 2022.

Research in Computing Science, year 21, Volume 151, No. 7, July 2022, is published monthly by the Center for Computing Research of IPN.

Volume 151(7)

Advances in Artificial Intelligence

Iris Iddaly Méndez-Gurrola (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2022

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

Seguridad infantil en el interior de automóviles usando un sistema en tiempo real

José-Sergio Ruiz-Castilla[1], Adrián Trueba-Espinosa[1], José Hernández-Santiago[2]

[1] Universidad Autónoma del Estado de México, México

[2] Tecnológico de Estudios Superiores de Chimalhuacán, México

jsergioruizc@gmail.com, atruebae@uaemex.mx, jose_hernandez_santiago@teschi.edu.mx

Resumen. En el mundo existen 1450 millones de vehículos, mientras que en México existen 50 millones. La mayoría son vehículos familiares de hasta 5 pasajeros. Por otro lado, en México existen 10 millones de niños entre 0 y 4 años. En múltiples eventos han dejado niños dentro del automóvil. Los infantes han sufrido choques de calor, deshidratación e incluso la muerte. Por ejemplo, en Estados Unidos se encontraron 231 muertes entre 1999 y 2007 en el interior de vehículos. Por lo anterior, se propone un Sistema de Seguridad para Infantes en el Interior de automóviles (SSIIA). Dicho sistema iniciará su operación cuando el conductor sale del automóvil y deja dentro a un infante. El sistema detecta al infante y activa un monitoreo de temperatura y del nivel de oxígeno. Cuando el calor o el nivel de oxigeno se vuelven un peligro se inicia una ventilación desde el exterior para mitigar el calor y mantener un nivel de oxígeno suficiente. El sistema deja de funcionar cuando el conductor abre la puerta del automóvil. Para lo anterior, se ha propuesto un dispositivo que lee sensores de la ocupación de los asientos, incluyendo el del bebé. Además de, otros sensores de temperatura, nivel de oxígeno y de puerta cerrada o abierta. En los resultados experimentales es posible probar su eficacia. Finalmente, se concluye que el SSIIA puede salvar la vida a niños dejados en el interior del automóvil bajo los rayos del sol.

Palabras clave: Seguridad infantil, automóvil, sensores, sistema en tiempo real.

Child Safety Inside Cars Using a Real-Time System

Abstract. There are 1.45 billion vehicles in the world, while in Mexico there are 50 million. Most are family vehicles with up to 5 passengers. On the other hand, in Mexico there are 10 million children between 0 and 4 years old. In multiple events it has happened that children have been left inside the car. Infants have

suffered heat shock, dehydration and even death. For example, in the United States, 231 deaths were found between 1999 and 2007 inside vehicles. Therefore, we propose a Safety System for Infants Inside Cars (SSIIA). Said system will start its operation when the driver leaves the car and leaves an infant inside. The system detects the infant and activates temperature and oxygen level monitoring. When the heat or the oxygen level becomes dangerous, ventilation is started from the outside to mitigate the heat and maintain a sufficient oxygen level. The system stops working when the driver opens a door of car. For the above, a device has been proposed that reads seat occupancy sensors, including that of the baby. In addition to other sensors for temperature, oxygen level and open or closed door. In the experimental results it is possible to test its effectiveness. Finally, it is concluded that the SSIIA can save the lives of children left inside the car under the sun's rays.

Keywords: Child safety, automobile, sensors, real-time system.

1. Introducción

En el mundo existen 1,450 millones de vehículos [1]. En México existen 50 millones según INEGI. De los cuales 34 millones de automóviles son familiares [2]. En México, existen 10 millones de niños entre 0 a 4 años [3]. Por lo que, en este trabajo de investigación se propone un sistema de protección a los infantes, entre 0 y 4 años, dejados dentro del automóvil. Dicho sistema, busca detectar y proteger al niño de choques de calor y muerte. El objetivo es crear una red de sensores para detectar un infante dentro del automóvil y cuidar su integridad. A medida que permanezca uno o más ocupantes en al automóvil cerrado habrá una concentración mayor de Dióxido de carbono (CO2) y otros gases. Por lo que, se propone un sistema de ventilación automatizada para regular la temperatura y mantener el nivel adecuado de oxígeno. Para lograra el objetivo, se plantea una red de sensores para detectar la presencia de un infante y medir la temperatura y el nivel de oxígeno. Si las condiciones internas representan un peligro para el infante se activaría una ventilación desde el exterior para mitigar la temperatura y oxigenar el interior.

2. Trabajos relacionados

Existen esfuerzos para evitar accidentes de niños dentro de automóviles. Como en el trabajo de [4] que busca la aplicación de Tecnologías de última generación para prevenir la muerte accidental de niños atrapados en vehículos estacionados. En dicho trabajo, se propone un dispositivo de seguridad desde un arreglo de sensores que permitan mitigar o salvar la vida de niños dejados dentro de un automóvil en un estacionamiento. Según el autor en los Estados unidos alrededor de 38 niños mueren al año por choques de calor después de haber sido dejados dentro de un automóvil.

Mientras que, [5] propone mantener ventilada la cabina del automóvil con el fin de mitigar el efecto invernadero y evitar la elevación de la temperatura.

En este caso se propone que la energía sea tomada desde un panel solar para que el sistema de ventilación sea totalmente independiente del sistema de energía del automóvil. No se considera el escenario de cuando el infante es dejado en la noche y el nivel de oxigeno bajara peligrosamente.

Por otro lado, [6] estudió el caso de muertes de niños dejados dentro de un automóvil. Los infantes murieron por hipotermia o bien por choque de calor. Durante la investigación post mortem se concluyó que algunos infantes murieron después de 6 horas dentro del automóvil. Sin embargo, el tiempo es variable de acuerdo con las condiciones climáticas y características del automóvil

Otro caso raro pero documentado es el trabajo de [7] en el cual se relata cuando un niño de tres años entró al automóvil y cerró la puerta. El niño murió asfixiado, porque se quedó dormido.

Otra causa de muerte en niños dentro de un automóvil es la muerte por hipotermia. En el trabajo de [8] se encontraron 231 muertes en los Estados Unidos desde 1999 hasta 2007. Las muertes fueron por hipotermia en coches estacionados con los infantes dentro. En este caso, el tiempo promedio de sobrevivencia fue de 4.6, horas.

La temperatura dentro de la cabina del automóvil cambia de acuerdo con la estación del año y de las condiciones climáticas. El autor [9] encontró que la temperatura mayor fue de 61, 68 y 76 grados centígrados mientras que la más baja fue de 10 grados, en este caso en Atenas, Grecia. El problema se genera cuando existe alguien dentro y el nivel de oxígeno va bajando.

En el trabajo de [10] se propuso la emisión de una alerta vía celular para alertar al padre que ha dejado a su hijo en el interior del automóvil, Sin embargo, se considera que esta medida es insuficiente debido a que depende de la distancia a la que se encuentre el padre del automóvil.

Por otro lado, podría no funcionar el teléfono por falta de energía o señal. Lo anterior, sigue poniendo en riesgo al infante.

Se añade que, en casos investigados el infante se quedó dormido durante el viaje y al llegar al destino el o los padres descienden dejando al infante dentro del automóvil bajo el sol por horas.

En el trabajo de [11] se encontró que la incidencia más alta de niños que murieron dentro del automóvil fue en menores de 4 años. En este caso, el 95% de niños fueron menores de 4 años y el 5% mayores de 4 años. En relación a las causas, el 27% de los casos sucedió porque los niños estaban jugando y se quedaron atrapados dentro del automóvil, el otro 73% porque los dejaron dentro de automóvil. De este último porcentaje, el 2% de los niños dejados dentro del automóvil fue de manera intencional. Las muertes sucedieron por golpes de calor o por asfixia.

En el trabajo de [12] se encontró que existe una baja conciencia de los padres al dejar a sus hijos dentro del automóvil. En este caso al menos el 24.85% de padres dejaron al menos una vez a sus hijos dentro del automóvil a pesar del riesgo.

En la Tabla 1 se muestran algunas características de los trabajos revisados, también, se agregan características de la propuesta. La tabla compara algunas ventajas de los trabajos revisados y de la propuesta de este trabajo de investigación.

Tabla 1. Tabla comparativa de trabajos revisados y la propuesta.

Características de las soluciones de los trabajos revisados	Características del método propuesto
Se propone una ventilación usando un panel solar [5]. No funciona de noche y lugares cerrados.	Se propone un sistema con una batería. Puede funcionar de día y de noche.
Se encontró que los niños permanecieron vicos hasta seis horas [6].	Se activará la ventilación cuando la temperatura sea alta y cuando falte oxígeno.
En el caso de [7] en niño se quedó dormido.	El niño puede seguir dormido y aun así el sistema ventilará el espacio
En el caso de [8], los niños murieron por hipotermia.	No se propone solución para este escenario.
En el trabajo de [10], se propone un mensaje por el celular a la madre o tutor. Depende del funcionamiento del celular.	El sistema es autónomo. No se requiere activar por el usuario.

3. Cabina del automóvil

El habitáculo del automóvil se denomina cabina. Así nos referiremos en adelante. La cabina puede ser para dos pasajeros y hasta 7 pasajeros en camionetas familiares. Sin embargo, en este trabajo nos enfocaremos a automóviles que tienen hasta 5 pasajeros que son la gran mayoría. El automóvil puede quedarse con los vidrios abiertos o cerrados.

3.1. Riesgos dentro del automóvil

- Asfixia. Cuando el nivel de oxígeno es muy bajo, puede causar la pérdida del conocimiento y la Muerte,
- Insolación. Cuando la temperatura es muy alta y deshidrata al infante,
- Muerte por asfixia. Cuando el nivel de oxígeno es insuficiente para mantener a la persona consciente y con vida,
- Hipotermia. El infante puede sufrir hipotermia si la temperatura es muy baja durante mucho tiempo.

3.2. Sensores para la detección

Existen dos soluciones para evitar daños al infante. La primera consiste en mantener la temperatura y nivel de oxígeno óptimo todo el tiempo. Esta solución es 100% segura, pero se requiere de energía para mantener los niveles. La segunda consiste en detectar

Tabla 1. Sensores para detectar la presencia del infante en el interior del automóvil.

Sensor	Función	Desventaja
Sensor en los asientos	Detecta si el asiento está ocupado (incluido el asiento de bebes)	Podría haber algún objeto de peso similar en el asiento
Sensor de movimiento	Detecta si alguien se mueve dentro de la cabina del automóvil	Con temperaturas altas se puede detectar movimiento
Sensor de temperatura	Puede detectar cuerpos con temperatura corporal	Dentro de la cabina puede haber temperaturas superiores a los 36 grados centígrados
Sensor de gases tóxicos	Detecta gases tóxicos como dióxido de carbono y monóxido de carbono	

la presencia del infante y mantener la temperatura y el nivel de oxígeno, mientras dure su estancia. Para detectar la presencia del infante se requieren sensores, con cierta redundancia para evitar fallas posibles. Ver la Tabla 2.

Lo anterior indica que podría ser necesario más de un sensor para estar seguros de que hay un infante dentro del automóvil cerrado. En la solución propuesta se han incluido los sensores de los asientos.

4. Método propuesto

Se propone el Sistema de Seguridad para Infantes en el Interior de Automóviles (SSIIA), el cual se activa cuando el conductor sale del automóvil y lo cierra con un infante dentro.

Para lograr la propuesta de solución se presenta un diagrama de flujo que permite mostrar los pasos a seguir. Ver la Fig. 1.

Se trata de un sistema en tiempo real que debe ser autónomo. Requiere de una fuente de alimentación, en este caso se propone una batería recargable mientras el automóvil está en funcionamiento. El sistema deberá estar en espera de datos de los sensores de los asientos para activar a Oc. Una vez activado Oc se activará el VAIA. El VAIA monitorea las entradas de TIA y NOIA. Si TIA o NOIA indican un riesgo de peligro se enciende el VAIA.

4.1. Red de sensores

4.1.1. Sensores de asientos

Existen diversos sensores para asientos de automóviles incluyendo sensores para detectar infantes. En este caso se proponen sensores comerciales que podrían adquirirse en el mercado. Si el automóvil tiene sensores desde la armadora pueden conectarse al SSIIA. Los sensores que se proponen se muestran en la Fig. 2.

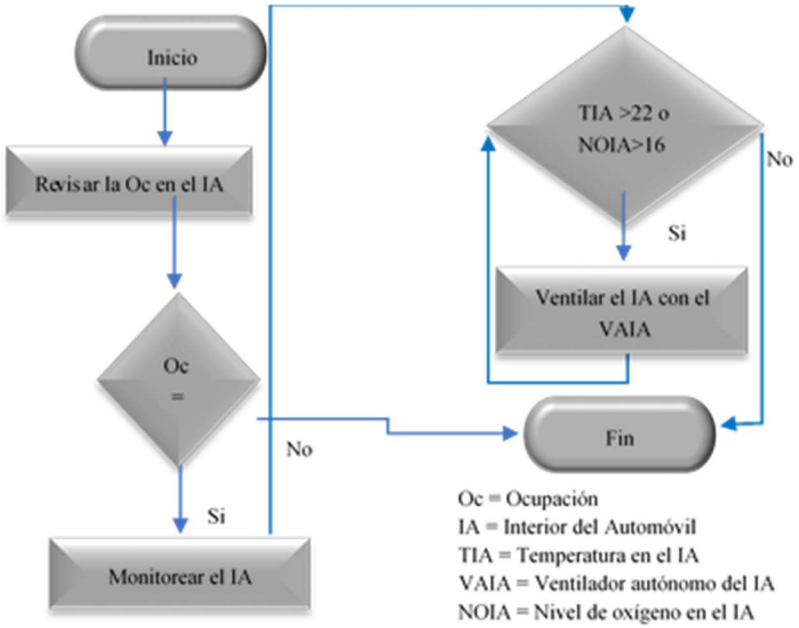


Fig. 1. Diagrama de flujo del método propuesto.

Este sensor mide el peso desde 5 kg. Por lo que detecta desde un niño de un año o más. En este caso si existen de 430 a 480 ohmios significa "Asiento no ocupado" y si la señal es de 120 ohmios o menos significa "Asiento ocupado". Dichos sensores son comerciales y pueden implementarse a cada asiento.

4.1.2. Sensores de temperatura

Se requiere también un sensor de temperatura para el interior del automóvil. La temperatura depende del exterior. Sin embargo, la temperatura puede ser mayor en el interior por el efecto invernadero. El calor es mayor debido a que las partículas de luz ingresan, pero no pueden salir y se acumulan dentro de la cabina, generando muy altas temperaturas. El sensor de temperatura tiene la función de enviar una señal cuando la temperatura podría provocar un choque de calor al infante. Se propone el sensor de la Fig. 3.

Este sensor tiene las siguientes características. Sensor de temperatura interior, Peso [kg]: 0,05, Fabricante del sensor: VALEO. Este sensor es requerido aun cuando los automóviles tienen un termómetro para el motor y otro para el medio ambiente exterior.

4.1.3. Sensores de nivel de oxígeno

A medida que respiramos dentro de la cabina cerrada del automóvil la cantidad de oxígeno va disminuyendo gradualmente. Si el infante se queda por horas dentro de la cabina el nivel de oxígeno bajará hasta volverse un riesgo de asfixia. El sensor enviará

Fig. 2. Sensor de asiento de pasajeros para automóviles [13].

Fig. 3. Sensor de temperatura para interior de vehículos [14].

una señal cuando el nivel de oxígeno se convierte en un riesgo. Se propone el sensor de la Fig. 4.

4.1.4. Sensores de puertas del automóvil

Los automóviles detectan cuando una puerta está abierta o cerrada. En este caso, se pueden usar los sensores instalados desde la fabricación o bien añadir algún sensor adicional a cada puerta. El sensor es importante porque sirve para que el SSIIA deje de funcionar cuando se abre la puerta del automóvil. Se supone que al abrir la puerta entrará aire más fresco y se disipará el calor.

Por lo que, al abrir la puerta se enviará una señal para que sistema deje de funcionar. Los sensores de las puertas son como se muestra en la Fig. 5.

Es necesario conectar el SSIIA a los sensores de las puertas, toda vez que la mayoría de los vehículos tienen instalados los sensores desde su fabricación. En este caso, no importa cual puerta se abra, eso garantiza que el infante es salvado o bien la cabina es ventilada y ya no es necesaria una ventilación automatizada.

Fig. 4. Sensor de nivel de oxígeno [15].

Fig. 5. Sensor de puerta abierta de automóvil [16].

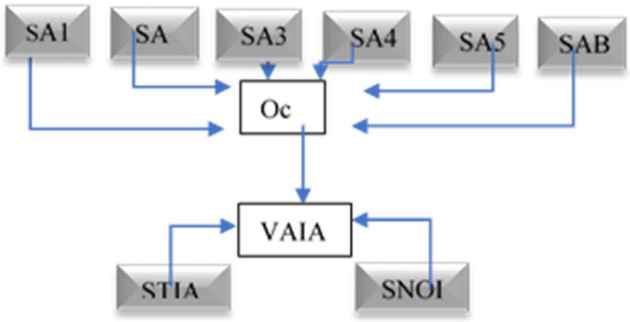


Fig. 6. Diagrama de sensores dentro del automóvil.

La red de sensores debe instalarse dentro de la cabina del automóvil de acuerdo con el siguiente diagrama. Ver la Fig. 6.

Los sensores corresponden a: SA1 Sensor del asiento 1, SA2 Sensor del asiento 2, SA3 Sensor de asiento 3, SA4 Sensor de asiento 4, SA5 sensor de asiento 5, SAB Sensor de asiento del bebé, STIA Sensor de la temperatura en el interior del automóvil y el SNOIA Sensor del nivel de oxígeno en el interior del automóvil

```
Leer SA1, SA2, SA3, SA4, SA5, SAB
```

```
Si Oc = SA1+ SA2 + SA3 + SA4 + SA5 + SAB
   Si Oc >= 1 Entonces Oc = True
   Si Oc = False
      Fin
   Sino
      VAIA = False
      Mientras VAIA = False
         Leer TIA
         Leer NOIA
         Leer SPC
         Si VAIA >= 1
            VAIA = True
            Ventilar el IA
         Sino
            VAIA = False
      Fin de Mientras
   Fin de Sino
Leer SA1, SA2, SA3, SA4, SA5, SAB
Fin de Si
```

Código 1. Código para el funcionamiento del SSIIA.

En este caso lo sensores SA1, SA2, SA3, SA4, SA5 y SBA corresponden a los sensores de cada uno de los asientos incluyendo el asiento para bebes. Si se detecta algún ocupante en cualquier asiento se activa Oc. Al activarse Oc se activa el VAIA para iniciar la medición de temperatura y nivel de oxígeno, a través de los sensores STIA y SNOIA. El VAIA iniciará la ventilación hasta que la temperatura o el nivel de oxigeno dejen de ser un peligro. Como es un sistema de tiempo real podrá volver a activarse si es necesario.

4.2. Comportamiento del sistema

El comportamiento del sistema basado en sensores se muestra en un diagrama de estados. Ver la Fig. 7.

Como sistema real está en ejecución todo el tiempo. Cuando el automóvil está ocupado se activa OC, pero no se activa el VAIA. El VAIA solo funciona si TIA es mayor de 22 grados o el nivel de Oxígeno es menor a 16 por ciento. El VAIA deja de funcionar cuando una puerta se abre.

Para lograr el funcionamiento del SSIIA se requiere un algoritmo implementado en algún lenguaje de programación. El programa reside en una tarjeta Arduino. Ver el código 1.

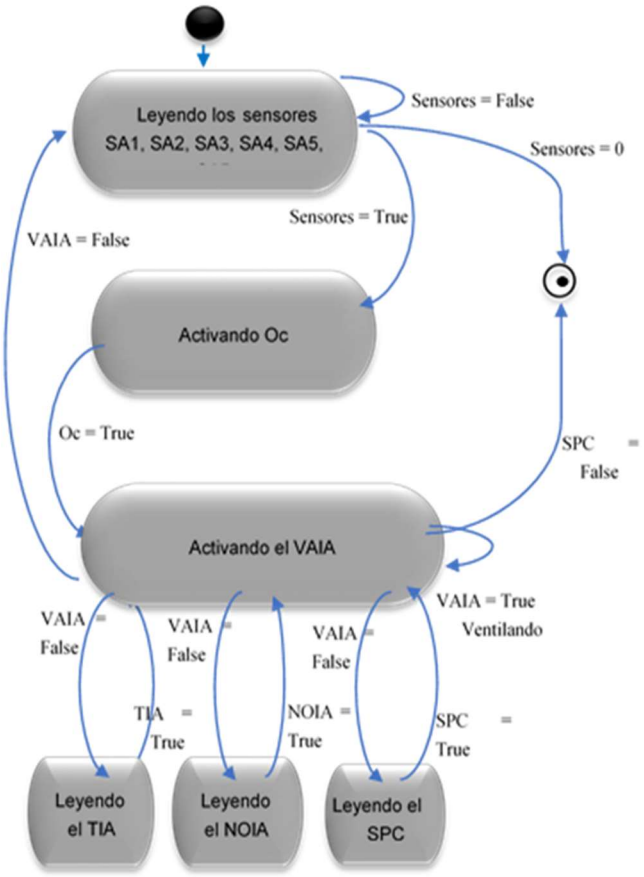


Fig. 7. Diagrama de estados del SSIIA.

5. Resultados experimentales

La Tabla 3 muestra los escenarios posibles de las lecturas de los sensores SA1, SA2, SA3, SA4, SA5, y SAB. Si todos los sensores emiten 0, entonces el sistema puede dejar de funcionar. Si uno o más sensores de asientos envían 1, entonces se activa Oc.

Como se puede observar cambia el estado cuando al menos un asiento está ocupado por el infante. En este caso Oc es True. Si todos los sensores son False el estado es False. Una vez que Oc es True el estado del VAIA es True y comenzará un ciclo que se repetirá ventilando el IA hasta que la puerta sea abierta. Ver la Tabla 4.

El VAIA ventilará el IA mientras los sensores TIA y NOIA envíen el valor de 1. El VAIA podría detenerse si los valores son 0, pero se vuelve a activar si los sensores en cuanto se envían un 1 de nuevo.

Tabla 2. Escenarios de la lectura de los sensores de los asientos.

Escenario	SA1	SA2	SA3	SA4	SA5	SAB	Oc activado
1	0	0	0	0	0	0	False
2	1	0	0	0	0	0	True
3	0	1	0	0	0	0	True
4	0	0	1	0	0	0	True
5	0	0	0	1	0	0	True
6	0	0	0	0	1	0	True
7	0	0	0	0	0	1	True

Tabla 3. Escenarios de las lecturas de los sensores de temperatura, oxígeno y puerta cerrada.

Escenario	TIA	NOIA	SPA	VAIA
1	0	0	0	False
2	1	0	0	True
3	0	1	0	True
4	0	0	1	True
5	1	1	1	True

6. Conclusiones y trabajo futuro

Se concluye que un sistema en tiempo real puede ser capaz de ventilar el interior de un automóvil y salvar a infantes dejado en el interior bajo los rayos del sol. Se requiere que el sistema sea autónomo porque el automóvil está detenido y apagado en algún estacionamiento u otro lugar. Que la energía puede tomarse de una batería recargable desde el propio automóvil. Que es necesario instalar sensores en los asientos, incluyendo el asiento para bebés para detectar al infante. Que una vez detectado el infante el sistema iniciará la medición de la temperatura y del nivel de oxígeno. Que el sistema puede ventilar el interior manteniendo condiciones favorables para mitigar el riesgo para el infante. Que al abrir la puerta del automóvil el sistema puede dejar de funcionar. Que el sistema puede instalarse desde la fabricación del automóvil, pero que también podría adoptarse a automóviles usados.

Referencias

1. Amadoz, S.: ¿Cuántos coches hay en el mundo en circulación? El Motor. Disponible en: https://motor.elpais.com/actualidad/cuantos-coches-hay-en-el-mundoencirculacion/ (2022)
2. INEGI.: Parque vehicular. Instituto Nacional de Estadística y Geografía. https://www.inegi.org.mx/temas/vehiculos/ (2022)
3. Amadoz, S.: Población de niños entre 0 y 4 años. El Motor. Disponible en: https://motor.elpais.com/actualidad/cuantos-coches-hay-en-el-mundo en circulacion/ (2022)
4. Aiello, V., Borazjani, P. N., Battista, E., Albanese, M.: Next-generation technologies for preventing accidental death of children trapped in parked vehicles. In: Proceedings of the

2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI), pp. 508–513 (2014) doi: 10.1109/IRI.2014.7051931
5. Vishweshwara, S. C., Al-Dhali, J. M.: Study of Excessive Cabin Temperatures of the Car Parked in Oman and its Mitigation. International Journal of Multidisplinary Sciences and Engineering, vol. 4, no. 9, pp. 18–22 (2013)
6. Krous, H. F., Nadeau, J. M., Fukumoto, R. I., Blackbourne, B., Byard, R. W.: Environmental Hyperthermic Infant and Early Childhood Death: Circumstances, Pathologic Changes, and Manner of Death. The American Journal of Forensic Medicine and Pathology, vol. 22, no. 4, pp. 374–382 (2021)
7. Byard, R. W., James, R. A.: Car window entrapment and accidental childhood asphyxia. Journal of Paediatrics and Child Health, vol. 37, no. 2, pp. 201–202 (2001) doi: 10.104v6/j.1440-1754.2001.00586.x
8. Booth, J. N., Davis, G. G., Waterbor, J., McGwin, J. G.: Hyperthermia deaths among children in parked vehicles: an analysis of 231 fatalities in the United States, 1999–2007. Forensic Science, Medicine, and Pathology, vol. 6, no. 2, pp. 99–105 (2010) doi: 10.1007/s12024-010-9149-x
9. Grundstein, A., Meentemeyer, V., Dowd, J.: Maximum vehicle cabin temperatures under different meteorological conditions. International Journal Biometeorology, vol. 53, no. 3, pp. 255–261 (2009) doi: 10.1007/s00484-009-0211-x
10. Hairulnizam, M., Abdul, H. O., Siti, S. Y., Shahreen, K., Mohd, F.: Minimizing heatstroke incidents for young children left inside vehicle. In: IOP Conference Series: Materials Science and Engineering, vol. 160 (2016) doi:10.1088/1757-899X/160/1/012094
11. Guard, A., Gallagher, S. S.: Heat related deaths to young children in parked cars: an analysis of 171 fatalities in the United States, 1995–2002. Injury Prevention, vol. 11, no. 1, pp. 33–37 (2005) doi: 10.1136/ip.2003.004044
12. Alowirdi, F. S., Al-harbi, S. A., Abid, O., Aldibasi, O. S., Jamil, S. F.: Assessing parental awareness and attitudes toward leaving children unattended inside locked cars and the risk of vehicular heat strokes. International Journal of Pediatrics and Adolescent Medicine, vol. 7, no. 2, pp. 93–97 (2020) doi: 10.1016/j.ijpam.2019.11.004
13. Ingeniería y Mecánica Automotriz: ¿Qué es el sensor de peso de asiento y cómo funciona? Disponible en: https://www.ingenieriaymecanicaautomotriz.com/que-es-el-sensor-de-peso-de-asiento-y-como-funciona/ (2022)
14. AUTODOC: Valeo 508793 Sensor, temperatura interior. TecAlliance. Disponible en: https://www.autodoc.es/valeo/7927015 (2022)
15. UbiBot: Embedded RJ45*, worry-free network connection. UBIBOT. Disponible en: https://www.ubibot.com/ubibot-gs1/ (2022)
16. AliExpress: Sensores de puertas abiertas de automóviles. Disponible en: https://es.aliexpress.com/ (2022)

ISSN 1870-4069

Evaluación comparativa de algoritmos de predicción aplicados al conteo de homicidios dolosos en México

Jorge Homero García-Gómez, Sergio Ivvan Valdez,
Hugo Carlos Martínez

Centro de Investigación en Ciencias de Información Geoespacial,
Ciudad de México,
México

hom.garcia@gmail.com,
{svaldez,hcarlos}@centrogeo.edu.mx

Resumen. En los últimos años, los homicidios dolosos se han incrementado drásticamente en México. Esta escalada de violencia sólo ha sido analizada considerando métodos estadísticos descriptivos cuando son aplicables a todo el país, o métodos de inteligencia computacional en zonas limitadas debido a la falta de datos de alta resolución. En contraste, este trabajo usa la base de datos del Secretariado Ejecutivo de Seguridad Pública de incidencia delictiva para todos los municipios de México, para realizar una evaluación comparativa entre los modelos Autoregressive Integrated Moving Average, la red neuronal recurrente Long Short-term Memory y Neural Prophet, para la predicción del número de homicidios dolosos en los 16 municipios de Guanajuato con mayor incidencia. Los resultados muestran la diferencia de rendimiento entre los modelos mencionados, los parámetros óptimos para cada uno, los casos en los que destaca cada modelo, y permite inferir el alcance de su utilidad. La comparación se realizó mediante el error absoluto medio, el error cuadrático medio y con pruebas estadísticas de hipótesis usando bootstrap. Los modelos son competitivos entre ellos, y aunque hay una diferencia en la media del error absoluto medio y el error cuadrático medio, estos no son estadísticamente significativos, por lo que se recomienda el uso de Autoregressive Integrated Moving Average debido a su menor complejidad computacional.

Palabras clave: ARIMA, LSTM, neuralProphet, predicción temporal, series de tiempo, homicidios dolosos.

Comparative Evaluation of Prediction Algorithms Applied to Counting Intentional Homicide in Mexico

Abstract. In recent years, intentional homicides have increased drastically in Mexico. This increase of violence has only been analyzed considering descriptive statistical methods when they are applicable to the entire country, or computational intelligence methods in limited areas due to the lack of high-resolution data. In contrast, this work uses the database of the "Secretariado Ejecutivo de Seguridad Pública" of criminal incidence for all the towns of Mexico, to

carry out a comparative evaluation between the Autoregressive Integrated Mobile Average models, the Long recurrent Short-term Memory neural network and Neural Prophet, for the prediction of the number of intentional homicides in the 16 towns of Guanajuato with higher incidence. The results show the difference in performance between the mentioned models, the optimal parameters for each one, the cases in which each model stands out, and allow inferring the scope of its usefulness. The comparison was made using the mean absolute error, the mean square error, and with statistical tests of hypotheses using bootstrap. The models are competitive among them, and although there is a difference in the mean of the mean absolute error and the mean square error, these are not statistically significant, so the use of Autoregressive Integrated Moving Average is recommended due to its lower computational complexity.

Keywords: ARIMA, LSTM, neuralProphet, temporal prediction, time series, intentional homicides.

1. Introducción

En México el número de homicidios dolosos (HD) ha tenido variaciones significativas en las últimas décadas, pues de acuerdo con [12] entre 1977 y 1992 la tendencia en los homicidios osciló entre los 17 y 21 homicidios por cada 100 000 habitantes, mientras que de 1992 hasta 2007 hubó una disminución monótona, pasando de los 20 a los 8 homicidios por cada 100 000 habitantes. Del 2007 al 2014 la tendencia osciló pasando de los 9 a los 13 homicidios por cada 100 000 habitantes [31].

Las tendencia observadas a partir del 2014 por el INEGI hasta el 2021 muestran un incremento significativo, por ejemplo en el 2020, [16] reporta 29 homicidios por cada 100 000 habitantes. Así mismo, se ha observado una disminución en el número de sentencias condenatorias para estos delitos [32]. Mientras que el aumento de la criminalidad y la ausencia del estado de derecho elevan la inestabilidad social, disminuyen la inversión privada y disminuyen el gasto público y privado, en consecuencia se reduce la actividad económica [17].

El fenómeno observado en nuestro país no es un problema que pueda atajarse o confrontarse de manera global, ya que los métodos y análisis que puedan aplicarse en algunas regiones no pueden aplicarse particularmente a otras, dada la complejidad del fenómeno [28]. Por ejemplo, el fenómeno involucra diferentes factores como son la espacialidad del territorio, la economía, la educación, la cultura, la infraestructura [28], entre otros factores que difieren en sus dinámicas y relaciones.

Otro factor que se debe considerar, se relaciona con la naturaleza, disponibilidad y calidad de los datos que los gobiernos ponen a disposición, pues estos difieren no sólo entre los países, sino que también difieren entre las jurisdicciones regionales internas de cada país [9]. Diferentes estudios que se han realizado en otras regiones han mostrado resultados convenientes no solo al caracterizar, sino al momento de combatir el fenómeno desde la realidad de cada país o ciudad.

Por ejemplo, [10] señala como las predicciones de las acciones criminales favorecen a los gobiernos, instituciones de seguridad y ciudadanía, ya que al conocer con antelación los patrones temporales delictivos, las patrullas vecinales y cuerpos de seguridad

se anteponen a los momentos y lugares de alta probabilidad de que se produzca un delito. En la literatura pueden encontrarse diversos trabajos con diferentes metodologías que buscan anteponerse al crimen mediante la predicción de los delitos, por ejemplo, [23] utiliza el método ARIMA para el pronóstico del crimen en los municipios de Colombia, [18] es otro trabajo que para la predicción de delitos violentos en el estado de Misisipi emplea el software WEKA de código abierto, en su estudio encontraron que el algoritmo de regresión lineal mostraba el mejor desempeño.

En el caso mexicano sólo se ha encontrado un estudio de predicción de delitos que emplea datos ajenos a la capital del país, [29] realiza la predicción de los homicidios dolosos para los 15 municipios mexicanos con mayor número de casos registrados, en el estudio emplean diferentes algoritmos de aprendizaje automático, este último estudio es el más cercano al aquí realizado, sin embargo, se diferencia del mismo por los modelos y métodos de comparación empleados.

Este trabajo utiliza diferentes modelos de series de tiempo para la predicción de HD, a saber un modelo estadístico, uno de red neuronal y otro que es una combinación de estadístico y red neuronal.

Se plantea buscar los hiper-parámetros (HP) óptimos de una red neuronal recurrente de tipo LSTM (Long-short term memory, memoria a largo-corto plazo), así como la ventana de tiempo (VT) óptima del algoritmo NP (Neural Prophet, Profeta neuronal), y los parámetros óptimos del modelo ARIMA (autoregressive integrated moving average, media móvil integrada autorregresiva), para realizar el pronóstico del número de homicidios en los 16 municipios del estado de Guanajuato con mayor incidencia (en la figura 1 se muestran los municipios del estado y puede apreciarse los 16 municipios seleccionados para esta investigación), utilizando bases de datos oficiales del Secretariado Ejecutivo del Sistema Nacional de Seguridad Pública (SESNSP), una instancia es el número de homicidios dolosos de un mes, en el periodo de tiempo comprendido entre enero del 2015 a diciembre del 2021.

Los resultados obtenidos por los mejores modelos LSTM, NP y ARIMA son evaluados mediante la comparación de su MAE (mean absolute error, error absoluto medio), su MSE (mean squared error, error cuadratico medio) y una prueba de hipótesis bootstrap, con el objetivo de encontrar el modelo que presenté el mejor desempeño en la predicción.

En la Sección 2 se describe lacónicamente trabajos relacionados con el pronostico de delitos. En la Sección 3 se introduce cada uno de los modelos: ARIMA, LSTM, y NP, finalmente se describe la comparación en rendimiento que se realiza entre los diferentes modelos. En la sección 4 se muestran los resultados y se discuten los mismos. Se finaliza presentando las conclusiones en la sección 5.

2. Trabajo relacionado

Existen diferentes estudios que sirven para la identificación de patrones, relaciones o tendencias en los datos relacionados a la criminalidad, los estudios son principalmente de dos tipos: estudios por visualización y estudios por modelación. de acuerdo con [26], los estudios por visualización más comunes son los análisis espaciales, entre otros, estos son:

– Análisis de frecuencia de un delito en una ciudad,
– Visualización por tipo de delito por área,
– Visualización de “puntos calientes” por tipo de delito,
– Gráficos descriptivos para la identificación de relaciones:
  – Tipos de delito durante un periodo de tiempo determinado,
  – Delitos cometidos en diferentes localidades,
  – Número de delitos por hora,
  – Delitos con mayor incidencia en una ciudad.

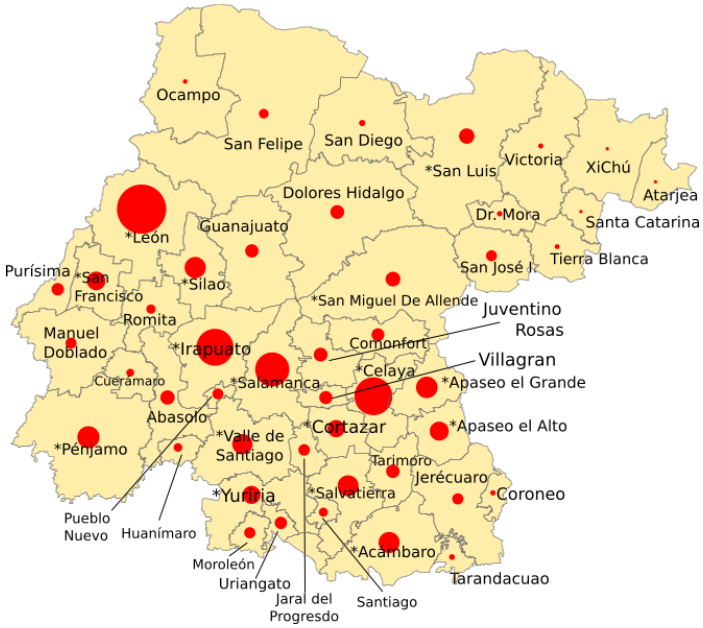


Fig. 1. División política del estado de Guanajuato donde se representa, con un circulo rojo de tamaño proporcional, el número de HD acumulados del 2015 al 2021. Los 16 municipios con mayor número de HD acumulado son selecionados para este estudio, su nombre comienza con un asterisco.

Los estudios por modelación pueden dividirse a su vez en dos subtipos, en estadísticos y de aprendizaje automático (machine learning). Uno de los modelos estadísticos usualmente empleados para el pronóstico de delitos es el modelo ARIMA, por ejemplo, [7] pronosticó los delitos en Chicago en un año utilizando los datos de los últimos 5 años con un 84 % de precisión.

Los estudios que emplean métodos de machine learning, suelen ser métodos supervisados [30], por ejemplo, para clasificación de tipos de crimenes, se utilizaron redes neuronales, máquina de soporte vectorial, k vecinos más cercanos y bosques aleatorios

[20]. Sugiriendo lo mismo que [30], quien reporta que los métodos de machine learning, más comúnmente utilizados son:

– K-Nearest Neighbours,
– Redes Neuronales,
– Boosting,
– Árboles de Decisión,
– Bosques Aleatorios.

En México se cuentan con diferentes estudios relacionados a la criminalidad que presenta el país, estudios geoestadísticos relacionados a los homicidios [13], estudios puntuales en la ciudad de México que emplean el método ARIMA para predecir delitos [21], estudios de clasificación y descubrimiento de patrones delictivos [5] o estudios que buscan optimizar rutas de patrullaje [8].

Sin embargo, la mayor parte de los estudios se concentran en la ciudad de México dado que los datos para el resto del país no son datos de alta resolución, en nuestra investigación solo se ha encontrado un estudio que emplea diferentes métodos de machine learning para el pronóstico de los homicidios que es aplicable a todo el país [29].

3. Metodología

La metodología seguida en esta investigación se describe en la Figura 2. Siguiendo el esquema de la Figura 2 de izquierda a derecha, se comienza por descargar los datos dispuestos por el SESNSP, que van de enero del 2015 a diciembre del 2021, el periodo seleccionado se debe a la nueva metodología de presentar los datos por parte del SESNSP, donde la información se encuentra más desagregada y específica respecto a la incidencia delictiva [14], esta nueva metodología se empezó a implementar a partir del 2015.

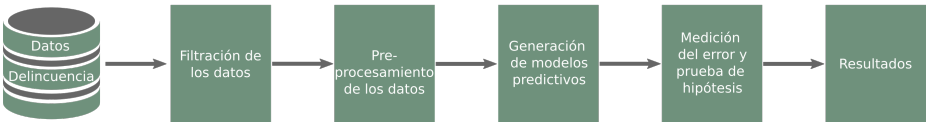


Fig. 2. Esquema general de la metodología empleada para este proyecto.

En el siguiente paso se filtran los datos, seleccionando los 16 municipios con mayor número de homicidios dolosos en el estado de Guanajuato. En el tercer paso se preprocesan los datos filtrados, transformándolos a una forma adecuada para el entrenamiento de cada modelo. Los modelos de predicción empleados son ARIMA, LSTM y NP.

Todos los modelos utilizarán 80 registros para el entrenamiento y 4 para prueba. En las siguientes subsecciones se describe cada uno de los modelos y los procedimientos seguidos en cada caso (cuarto paso en el esquema de la Figura 2), la Subsección 3.4 explica el protocolo utilizado para la evaluación y comparación de los resultados de los modelos (quinto paso en el esquema de la Figura 2).

3.1. Modelo ARIMA

El modelo ARIMA propuesto por [4] consiste en la combinación de dos modelos, el modelo AR (auto-regresivo) y el modelo MA (promedio movil), la expresión del modelo ARMA se presenta en la ecuación (1):

$$y_t = -\sum_{j=1}^{p} \alpha_j y_{t-j} + \sum_{j=1}^{q} \beta_j \epsilon_{t-j}, \tag{1}$$

donde comúnmente $\beta_0 = 1$, mientras que el termino y_t refiere al valor a predecir, α_j concierne a los términos autoregresivos, β_j son los parámetros de las medias móbiles y ϵ es el termino de perturbación estocástica, también conocido como el termino de error [19]. El modelo ARIMA incluye la diferenciación (d) de la serie con el propósito de transformarla a estacionaria, así pues, la I en la sigla ARIMA refiere a la palabra integrado (sumado) [19].

Para la obtención de los parámetros necesarios en la creación del modelo, se utilizó la biblioteca **pmdarima** desarrollada por [24]. La biblioteca **pmdarima** prueba varios conjuntos de parámetros p y q, seleccionando el modelo que minimiza el criterio de Información de Akaike (AIC) propuesto por [2]. Mientras que para la selección del término de diferenciación d, utiliza la prueba de Dickey-Fuller aumentada de [22].

El procedimiento para la obtención del modelo ARIMA para cada uno de los 16 municipios seleccionados, inició por la conversión de los datos, a una estructura de datos de serie de tiempo, posteriormente se seleccionaron los parámetros p, d, q, y se realizó el ajuste del modelo a cada una de las series de tiempo. A continuación se realizó la predicción del modelo, y finalmente se almacenaron los resultados de cada municipio.

3.2. Modelo LSTM

El algoritmo LSTM propuesto por [15] se caracteriza por no presentar el problema de las estructuras RNN conocidos como “desaparición del gradiente”, especialmente observados en secuencias de entradas grandes [1].

Además presentan un mejor rendimiento de convergencia [11] y son convenientes (al igual que RNN) al trabajar con bases de datos con componente temporal o secuencial. Las redes LSTM se componen de múltiples funciones en comparación de una RNN, dichas funciones pretenden olvidar los datos innecesarios de las entradas y procuran recordar o conservar los datos de utilidad a la predicción. Las funciones que

integran una red LSTM se presentan a continuación:

$$f_t = g(W_f y_t + U_f h_{t-1} + b_f), \tag{2}$$

$$k_t = \tanh(W_k y_t + U_k h_{t-1} + b_k), \tag{3}$$

$$c_t = f_t c_{t-1} + i_t k_t, \tag{4}$$

$$o_t = g(W_o y_t + U_o h_{t-1} + b_o), \tag{5}$$

$$h_t = o_t \tanh(t_h), \tag{6}$$

donde y_t es la entrada de datos en el tiempo t y g es una función de activación no lineal como la función $\mathrm{sigmoide}$ o $ReLU$. W, U son matrices de ponderaciones y b es el vector conocido como sesgo. h_t y c_t son la salida y vector de estado de celda en el tiempo t. f_t es utilizado para recordar información antigua y sirve para obtener información nueva [1].

Para la predicción se utiliza el concepto de ventana móvil temporal (VT), el cual consiste en utilizar h pasos de tiempo y posteriormente ir desplazando i pasos la ventana temporal. En este caso el objetivo es predecir un mes a la vez, por lo que la ventana a utilizar queda determinada como $k = t - h, t - h + 1, \ldots, t - 1$.

Para este experimento se realizaron las predicciones utilizando únicamente la variable endógena del municipio a predecir (al igual que en ARIMA). Los hiper-parámetros de los modelos LSTM se buscaron exhaustivamente en el conjunto de las combinaciones siguientes:

- Se probaron VT desde 2 hasta 12 meses,
- Se probó dropout en 0 y 0.2,
- Se probaron las funciones de activación sigmoide y tanh,
- Se probaron números de neuronas de 20, 40 y 60.

Las diferentes combinaciones producen 132 resultados en total. Todos los modelos se desarrollaron con un arreglo de dos capas donde la segunda capa tenía la mitad del número de neuronas de la primera capa. Antes de aplicar la LSTM se les quito la tendencia a los datos restandoles un modelo de regresión lineal, y se normaliaron, restando el promedio y dividiendo entre la desviación estándar.

3.3. Neural Prophet

NP es un modelo reciente propuesto en noviembre de 2021 por [27], y es una extensión del algoritmo Facebook Prophet propuesto en 2018 por [25]. El modelo NP esta integrado por seis componentes tal como se muestra en la siguiente ecuación:

$$Y_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(t), \tag{7}$$

donde $T(t)$ corresponde a la tendencia de la serie temporal después de eliminar la estacionalidad, $S(t)$ corresponde a la estacionalidad y $E(t)$ corresponde a los eventos "especiales" preestablecidos. Estas primeras tres componentes se corresponden a la parte estadística del algoritmo NP, cada una de sus componentes se ajusta individualmente.

Las componentes $L(t)$ son las covariables actuales retrasadas n periodos de tiempo, mientras que el pronóstico para estas covariables se modela con el término $F(t)$. La componente $A(t)$ conocida como término autorregresivo aplica el concepto de observación de valores pasados, y usa los mismos para la predicción de valores futuros.

Las ultimas tres componentes del modelo NP se corresponden a la parte de red neuronal, por lo que NP es un algoritmo con una parte estadística y una parte de red neuronal, de acuerdo con [6] esta combinación presenta la ventaja de proporcionar una ganancia en eficiencia al tiempo que se limita la pérdida de interpretabilidad.

Las pruebas realizadas con NP se realizaron de manera análoga a las realizadas con el modelo LSTM, utilizando las variables endógenas correspondientes al municipio que se predice. Mencionar que en cuanto al VT se realizó el mismo número de combinaciones que en los modelos LSTM, por que que se generaron en este caso 11 resultados con este modelo.

3.4. Evaluación de modelos

Una vez obtenidas las predicciones por los diferentes modelos, se evalua su desempeño empleando el MAE y MSE, y una prueba de hipótesis de medias, para la media del MAE, usando el método bootstrap.

El método bootstrap, sirve, entre otras cosas, para hacer pruebas de hipótesis [3] no-paramétricas, y consiste en tomar repetidas muestras de una población con reemplazo, estas muestras son observaciones de la distribución empírica, con ellas se pueden calcular estimadores que se aproximan a los de la distribución subyacente de los datos. En este caso estimamos la probabilidad de que la media del MAE producido por un algoritmo sea menor que la media de otro, el resultado es un estimador del $p - \text{valor}$ de la pruena de hipótesis de medias.

Para comparar los modelos se calcula el MAE y MSE sobre los datos de prueba para los 16 municipios con mayor número de homicidios del estado de Guanajuato. Todos usan la misma cantidad de datos de prueba, que son 4 por municipio, 80 instancias de entrenamiento.

Ya que el número de combinaciones de parámetros que se probaron es exhaustiva, se presenta los resultados para el modelo ARIMA con la mayor AIC, usando para cada municipio los valores de d, p y q que maximizan el AICc. También se comparan los 3 modelos NP y los 11 de LSTM con el MAE menor. Finalmente, se realizaron pruebas de hipótesis que comparan las medias de cada modelo contra todos los demás, de la siguiente forma:

- Hipótesis nula (H_0): Indica que el rendimiento de un modelo A es igual a un modelo B,
- Hipótesis alternativa (H_1): Indica que la media del error del modelo A es menor que la media del error del modelo B.

4. Resultados y discusión

El objetivo de esta sección es comparar el desempeño en el MAE y MSE de los diferentes algoritmos así como reportar los parámetros óptimos, dentro del conjunto de

búsqueda, que se encontraron mediante pruebas exhaustivas. El Cuadro 1 muestra los modelos ordenados por MAE.

Tabla 1. Resultados de los modelos con mejor desempeño, seleccionados para compararlos con la prueba de hipótesis donde ninguno resultó significativamente diferente. Se muestran los hiperparámetros del modelo y su respectivo MAE y MSE.

				MAE	**MSE**
ARIMA				3.210557	21.003021
		NP			
VT	parámetros				
11				3.258145	26.683529
2	por omisión			3.297269	24.408899
3				3.44949	25.374554
		LSTM			
VT	nn	$f_{\text{activación}}$	dropout		
8	60	tangh	0	3.580727	24.947718
7	40	tangh	0.2	3.602632	25.467693
8	20	tangh	0.2	3.61437	23.493635
12	60	tangh	0.2	3.629066	25.112785
2	40	sigmoid	0.2	3.641926	23.37434
11	40	sigmoid	0.2	3.665473	26.507914
4	60	sigmoid	0.2	3.681434	26.659206
6	20	sigmoid	0.2	3.692786	27.279583
5	40	tangh	0.2	3.706302	26.986547
3	20	sigmoid	0.2	3.724117	26.862549
2	60	tangh	0.2	3.733833	24.569363

Usualmente se observa que existe una correspondencia entre el MAE y el MSE, sin embargo, se advierte en el Cuadro 1 que algunos modelos arrojan un mejor resultado en el MAE mientras que otros en el MSE, esto se debe a que algunos errores están entre 0 y 1, y cuando se elevan al cuadrado se hacen menores.

De los resultados puede observarse, en el Cuadro 1, que el modelo ARIMA arroja el mejor resultado, sin embargo, la prueba de hipótesis no permitió rechazar la hipótesis nula, esto es, se confirmó que no existe diferencia estadística entre los modelos. Los desempeños observados en los modelos LSTM y NP, puede deberse a que los registros dispuestos por SESNSP son insuficientes para el correcto entrenamiento de los modelos, pues estos modelos fueron ideados para la predicción de grandes conjuntos de datos.

Otro punto es que para ARIMA los parámetros q, p y d se buscan para cada municipio, mientras que los errores reportados para LSTM y NP usan los mismos hiper-parámetros para todos los municipios. Finalmente, una búsqueda en un conjunto mayor de valores y que incluya más hiperparámetros podría mejorar los resultados, pero dados los resultados actuales, no se espera una gran mejora.

La Figura 3 muestra las predicciones para el municipio de Irapuato con ARIMA en la izquierda y LSTM a la derecha. Es evidente que ARIMA tiene una predicción con

menos error para los 4 meses de prueba, sin embargo, LSTM parece seguir la tendencia de los últimos meses para la predicción.

Por lo tanto, LSTM no puede capturar el patrón de la red, se puede deber a hiper-parámetros incorrectos, por ejemplo, el VT de 12 puede ser el mejor en promedio, pero no el mejor para el caso específico de Irapuato, otra razón puede ser que la cantidad de datos es insuficiente, y por último, que existen factores del modelo como otros valores de hiper-parámetros o normalización que pueden mejorar los resultados, esta última se considera la menos probable, ya que se realizó una búsqueda exhaustiva en ciertas combinaciones de hiper-parámetros, por lo que una búsqueda más fina se espera que sólo mejore marginalmente, en este sentido, la via más admisible sería buscar entre hiper-parámetros que no se optimizaron en este trabajo u otros factores como la estandarización o normalización de los datos.

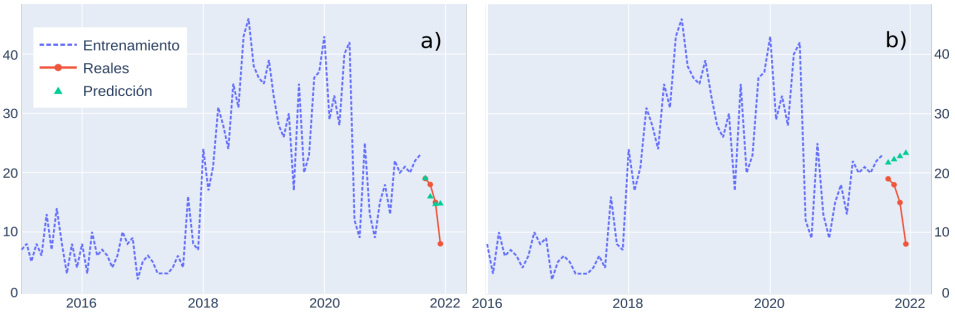


Fig. 3. La imagen de la izquierda (a) muestra los resultados del modelo ARIMA en la predicción de los HD en el municipio de Irapuato, mientras que la imágen de la derecha (b) muestra los mismos resultados obtenidos con el modelo LSTM con una VT de 12, 60 neuronas en la primer capa y sin dropout. El conjunto de entrenamiento se representa en color azul, las lineas rojas son los valores reales, mientras que la linea verde muestra la predicciones realizadas por los modelos.

5. Conclusiones

Este trabajo explora tres modelos, ARIMA, LSTM y Neural Prophet para la predicción mensual del número de homicidios dolosos en los 16 municipios con mayor incidencia del estado de Guanajuato, México. Los resultados obtenidos muestran que el modelo ARIMA presenta el menor MAE y MSE, sin embargo, se encontró que no existe una diferencia estadística significativa entre los modelos, Por lo que se recomienda el empleo del modelo con menor complejidad y coste computacional, el modelo ARIMA.

En contraste con los resultados encontrados, hay un gran auge en el uso de LSTM y Neural Prophet para la predicción de series de tiempo, por lo que se considera que aún se deben explorar modificaciones a los experimentos presentados en este estudio con el fin de intentar mejorar su desempeño. A partir de los resultados obtenidos, se consideran 3 posibles explicaciones por las que ARIMA resulto con mejor desempeño:

1. Que la cantidad de datos para entrenar los modelos LSTM y NP es insuficiente.

2. Que se pueden buscar hiper-parámetros y otras variables como normalización o estandarización diferente de los datos, e información de los municipios adyacentes.
3. Que cada municipio requiere hiper-parámetros diferentes, ARIMA utiliza un conjunto $\{p, q, d\}$ para cada municipio, mientras que el modelo de LSTM y NP se reporta con los mismos hiperparámetros para todos los municipios.

A partir de estos argumentos, el trabajo futuro explorará :

– Incorporar los datos anteriores a la nueva metodología del SESNSP, lo que incrementaría en cuatro años los datos,
– Incluir en los modelos LSTM y NP las serie temporal concerniente al municipio que se desea realizar la predicción y la serie del municipio adyacente con mayor número de HD,
– Reportar los errores de predicción por municipio. Aunque una de las razones para no hacerlo así es que la cantidad de datos es poca, y un estadístico sobre 4 datos de prueba no es suficiente para inferir diferencias estadśiticas.

Referencias

1. Aggarwal, C. C.: Neural networks and deep learning. Springer, vol. 10, pp. 419–458 (2018) doi: 10.1007/978-3-319-94463-0
2. Akaike, H.: Stochastic theory of minimal realization. IEEE Transactions on Automatic Control, vol. 19, no. 6, pp. 667–674 (1974) doi: 10.1109/TAC.1974.1100707
3. Bartlett, J.: The miracle of the bootstrap The Stats Geek
4. Box, G., Jekins, G.: Time series analysis: forecasting and control. Water Science and Technology, (1976)
5. Campedelli, G. M.: Explainable machine learning for predicting homicide clearance in the United States. Journal of criminal justice, vol. 79 (2022) doi: 10.1016/j.jcrimjus.2022.101898
6. Catherine, V.: In-depth understanding of NeuralProphet through a complete example. Towards data science (2022)
7. Cesario, E., Catlett, C., Talia, D.: Forecasting crimes using autoregressive models. In: IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress. pp. 795–802. IEEE (2016) doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.138
8. Chainey, S., Matias, J., Junior, F., Silva, T., Macedo, J., Pires Magalhaes, R., de Queiroz Neto, J. F., Silva, W.: Improving the creation of hot spot policing patrol routes: Comparing cognitive heuristic performance to an automated spatial computation approach. ISPRS International Journal of Geo-Information, vol. 10, no. 8, pp. 560 (2021) doi: 10.3390/ijgi10080560
9. Dávila, E. A. S.: Rentabilidad de los recursos que la políticas públicas municipales ponen a disposición de las PYME´S considerando su cultura organizacional. Academia Jornals, vol. 8, no. 3, pp. 1882–1887 (2016)
10. Devarakonda, D. S.: Time series analysis and forecasting of crime data. Ph.D. thesis, Master Dissertation Thesis, California State University (2019)
11. DiPietro, R., Hager, G. D.: Deep learning: RNNs and LSTM. In: Handbook of medical image computing and computer assisted intervention, pp. 503–519. Elsevier (2020)

12. Escalante, G. F.: Panorama del homicidio en México. Esquema de análisis territorial 1990-2007. Los grandes problemas de México. Seguridad nacional y seguridad interior, pp. 301–330 (2010)
13. Fuerte-Celis, M. P., Sánchez-Castañeda, M. D.: Clusters of violence in mexico: An analysis of homicide rates from 2000 to 2012. Journal of Latin American Geography, vol. 20, no. 1, pp. 99–133 (2021)
14. Gobierno de México: Incidencia delictiva del secretariado ejecutivo del sistema nacional de seguridad pública (2022)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, vol. 9, no. 8, pp. 1735–1780 (1997) doi: 10.1162/neco.1997.9.8.1735
16. INEGI: Datos preliminares revelan que en 2020 se registraron 36 579 homicidios (2021)
17. Loría, E.: Impacto de secuestros y homicidios en la inversión extranjera directa en México. Contaduría y administración, vol. 65, no. 3 (2020) doi: 10.22201/fca.24488410e.2020.2246
18. McClendon, L., Meghanathan, N.: Using machine learning algorithms to analyze crime data. Machine Learning and Applications: An International Journal (MLAIJ), vol. 2, no. 1, pp. 1–12 (2015) doi: 10.5121/mlaij.2015.2101
19. Montenegro, A.: Análisis de series de tiempo. Bogotá: Pontificia Universidad Javeriana, (2010)
20. Mosquera Cabra, J.: Trabajo de grado sobre predicción de los tipos de delitos en Medellín. Tesis de Licenciatura, Universidad de Antioquia, Facultad de Ciencias Económicas, (2021)
21. Pambabay-Calero, J., Bauz-Olvera, S., Flores-González, R., Piña-García, C.: Multivariate analysis and characterization of low impact crime in Mexico City. F1000Research, vol. 10, no. 1299, pp. 1–22 (2021) doi: 10.12688/f1000research.72990.1
22. Phillips, P. C., Perron, P.: Testing for a unit root in time series regression. Biometrika, vol. 75, no. 2, pp. 335–346 (1988)
23. Santos-Marquez, F.: Spatial beta-convergence forecasting models: Evidence from municipal homicide rates in Colombia. Journal of Forecasting, vol. 41, no. 2, pp. 294–302 (2022)
24. Smith, T. G.: ARIMA estimators for Python, about the project (2017)
25. Taylor, S. J., Letham, B.: Forecasting at scale. The american statistician, vol. 72, no. 1, pp. 37–45 (2018) doi: 10.1080/00031305.2017.1380080
26. ToppiReddy, H. K. R., Saini, B., Mahajan, G.: Crime prediction & monitoring framework based on spatial analysis. Procedia computer science, vol. 132, pp. 696–705 (2018) doi: 10.1016/j.procs.2018.05.075
27. Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., Rajagopal, R.: NeuralProphet: Explainable forecasting at scale. arXiv preprint arXiv:2111.15397, (2021) doi: 10.48550/arXiv.2111.15397
28. Valasik, M., Brault, E. E., Martinez, S. M.: Forecasting homicide in the red stick: Risk terrain modeling and the spatial influence of urban blight on lethal violence in Baton Rouge, Louisiana. Social Science Research, vol. 80, pp. 186–201 (2019) doi: 10.1016/j.ssresearch.2018.12.023
29. Valdez, S. I., Hernández-Baena, A.: On the best-performed time window size for homicide count forecasting. In: 2021 Mexican International Conference on Computer Science (ENC). pp. 1–8. IEEE (2021) doi: 10.1109/ENC53357.2021.9534830
30. Vaquero-Barnadas, M.: Machine learning applied to crime prediction. B.S. thesis, Universitat Politècnica de Catalunya (2016)
31. Zepeda Lecuona, G. R., Jiménez Rodriguez, P.: Impunidad frente al homicidio doloso en México. Este País, vol. 308, pp. 13–22 (2016)
32. Zepeda Lecuona, G. R., Jiménez Rodriguez, P.: Impunidad en homicidio doloso en México: reporte 2019. Impunidad Cero, (2019)

ISSN 1870-4069

Un framework basado en ROS para la navegación de robots móviles autónomos

Rosa-L. Villarreal-O., Abraham Sánchez-L., Martin Estrada-A., Rogelio González-V.

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{rlauravillarreal, rogelio.gzzvzz}@gmail.com
{abraham.sanchez, martin.estrada}@correo.buap.mx

Resumen. La necesidad de incrementar la autonomía en las aplicaciones robóticas ha motivado la creación y desarrollo de robots móviles. El propósito es limitar en todo lo posible la intervención humana. La autonomía de un robot móvil se basa principalmente en el sistema de navegación autónoma. La propuesta de este trabajo es un framework para la navegación autónoma de robots móviles basado en el sistema operativo robótico (ROS), lo que permitirá implementar las diversas tareas relacionadas en una serie de pasos lógicos y sencillos para integrar el sistema de navegación en un robot móvil. Mediante la construcción de un solo ecosistema, se busca reducir el tiempo empleado para configurar las actividades y requisitos para la navegación, al mismo tiempo que se simplifica el proceso. Se presentan ejemplos prácticos simulados para validar la propuesta.

Palabras clave: Robot móvil, navegación, ROS, arquitectura, simulación.

A ROS-Based Framework for Autonomous Mobile Robot Navigation

Abstract. The need to increase autonomy in robotic applications has motivated the creation and development of mobile robots. The purpose is to limit human intervention as much as possible. The autonomy of a mobile robot is mainly based on the autonomous navigation system. The proposal of this work is a framework for autonomous navigation of mobile robots based on the robotic operating system (ROS), which allows implementing the various related tasks in a series of logical and simple steps to integrate the navigation system in a mobile robot. By building a single ecosystem, it seeks to reduce the time spent configuring the activities and requirements for navigation, while simplifying the process. Practical simulated examples are presented to validate the proposal.

Keywords: Mobile robot, navigation, ROS, architecture, simulation.

1. Introducción

Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y el guiado del vehículo a través de la referencia construida [1, 9]. Realizar una tarea de navegación para un robot móvil significa recorrer un camino que lo conduzca desde una posición inicial hasta otra final, pasando por ciertas posiciones intermedias o submetas. El problema de la navegación se divide en las siguientes cuatro etapas [6, 9]:

- Percepción del mundo: Mediante el uso de sensores externos, creación de un mapa o modelo del entorno donde se desarrollará la tarea de navegación [2],
- Planificación de la ruta: Crear una secuencia ordenada de objetivos o submetas que deben ser alcanzadas por el vehículo. Esta secuencia se calcula utilizando el modelo o mapa de entorno, la descripción de la tarea que debe realizar y algún tipo de procedimiento estratégico [6],
- Generación del camino: En primer lugar, define una función continua que interpola la secuencia de objetivos construida por el planificador. Posteriormente procede a la discretización de la función, con el fin de generar el camino,
- Seguimiento del camino: Efectúa el desplazamiento del vehículo, de acuerdo al camino generado mediante el control adecuado de los actuadores del vehículo [2].

Estas tareas pueden llevarse a cabo de forma separada, aunque en el orden especificado. La interrelación existente entre cada una de estas tareas conforma la estructura de control de navegación básica en un robot móvil. La complejidad del sistema necesario para desarrollar esta tarea de navegación, depende principalmente del conocimiento que se posee del entorno de trabajo. En general, para resolver el problema de navegación de robots, se necesita encontrar respuesta a las preguntas: ¿Dónde está el robot?, ¿A dónde se dirige?, ¿Cómo llegará ahí? [9, 10]. La principal complejidad en la resolución de tareas propias de la robótica, es justamente la implementación de algoritmos para diferentes funciones que proveen en mayor medida el grado de autonomía al robot móvil.

ROS, por sus siglas Robotic Operating System, es un framework para el desarrollo de software para robots, que provee la funcionalidad de un sistema operativo [7]. Proporciona servicios estándar como la abstracción del hardware, el control de dispositivos de bajo nivel, la implementación de funcionalidad de uso común, el paso de mensajes entre procesos y el mantenimiento de paquetes. Además, es de código abierto lo que permite realizar colaboraciones de mejora a través de su comunidad.

Este framework pone a disposición una serie de herramientas y librerías para obtener, construir, escribir, y ejecutar código a través de múltiples computadoras [8]. Como sabemos, la navegación de robots móviles conlleva una serie de pasos para funcionar, de hecho, implementar un sistema de navegación para un robot desde cero definitivamente consumiría mucho tiempo y esfuerzo. Afortunadamente, ROS provee esta funcionalidad a través del paquete Navigation Stack lo que facilita la tarea

principal, sin embargo, esta pila de trabajo es descentralizada, lo que significa que cada paso de la navegación se debe realizar de forma separada y como un proceso independiente, lo que puede dificultar la labor sobre todo a personas con poca experiencia en el área de la robótica o en el uso de ROS.

La propuesta que se realiza entonces, es construir un framework que incluya todas las funcionalidades de los nodos de la pila de navegación en una única ventana, con la posibilidad de escoger distintos mapas y configurar parámetros. Efectivamente hay libros que detallan aspectos de ROS, pero estos no son fáciles de entender y menos de llevar a cabo, sino se tiene experiencia en la robótica móvil, es una de nuestras aportaciones importantes.

Se presentan al final del presente trabajo, una serie de simulaciones realizadas con este framework, aclarando que no fue posible probar con el robot real, un robot Pioneer 3DX con el que cuenta el grupo de trabajo, debido a las restricciones impuestas por la pandemia de Covid19. Si bien, no se presentan resultados realizados con el robot real, ello no significa que no se realizarán; de hecho, antes de liberar el código del framework, realizaremos experimentos con nuestro robot Pionner P3DX.

2. Sistema operativo robótico

Robot Operating System (ROS), es un conjunto de herramientas informáticas en forma de software libre (el sistema está bajo licencia BSD) de código abierto, que permite el desarrollo de software para robótica. Fue desarrollado originalmente en 2007 por la empresa estadounidense Willow Garage, para su robot PR2 (Personal Robot 2) [7]. Las principales ventajas de utilizar el conjunto de herramientas y librerías que proporciona ROS se pueden resumir a continuación:

- Mecanismo de comunicación entre programas: Es un estándar que permite la comunicación entre diversos programas de un mismo sistema, ya sea en una computadora o en varias,
- Reusabilidad de código: Los paquetes estándar proporcionados en las distribuciones de ROS, implementan muchos de los algoritmos comúnmente usados en robótica que ya han sido depurados y usados de forma estable. Es decir, no es necesario reinventar la rueda, en muchos de los casos ROS ya cuenta con un paquete que proporciona la o las funcionalidades que se buscan para un robot,
- Testeado rápido: Puesto que ROS implementa un diseño de comunicación por paso de mensajes, se pueden realizar simulaciones para aislar la funcionalidad del sistema y así crear conjuntos de prueba.

Para utilizar la propuesta del framework, se utilizaron algunas herramientas adicionales, las cuales se mencionan a continuación:

Rviz: Es una herramienta de visualización 3D para ROS. Proporciona una vista del modelo del robot que se carga en el servidor de parámetros de ROS, y que es provisto como un archivo urdf con la descripción exacta de la constitución del robot. Captura la información de sus sensores, si existen, y reproduce estos datos para su visualización. Es importante señalar que rviz no es una herramienta o programa de simulación, para este caso existen otras opciones como Gazebo.

Rviz es ampliamente utilizado ya que le permite al usuario visualizar diversos elementos como el modelo del robot, sus ejes, el láser del escáner, entre muchos otros que pueden ser agregados al panel principal.

Gmapping: Es un paquete que incluye un contenedor ROS para Gmapping de OpenSlam. Proporciona SLAM basado en laser que permite crear mapas de rejilla de ocupación 2D a partir de los datos recopilados por el láser y de pose del robot móvil.

Este paquete contiene un nodo llamado slam_gmapping que es comúnmente usado para construir mapas. El requisito básico de hardware para hacer SLAM es un escáner láser que esté montado horizontalmente en la parte superior del robot, así como los datos de odometría del mismo. El mapa que se crea se publica en el tema/map durante todo el proceso de mapeo.

AMCL: Es un sistema de localización probabilística para un robot móvil en 2D. Implementa el enfoque de localización de Monte Carlo adaptativo (o muestreo KLD), como lo describe Dieter Fox, que utiliza un filtro de partículas para rastrear la pose de un robot en un mapa conocido. Toma un mapa basado en láser, escaneos láser y mensajes de transformación y genera estimaciones de pose. Al inicio, amcl inicializa su filtro de partículas de acuerdo con los parámetros proporcionados, en caso de no ingresar ningún parámetro, toma los valores por defecto definidos en su documentación.

Navigation stack: La pila de navegación 2D que ROS pone a disposición toma la información de la odometría, flujos de sensores y posición meta con el objetivo de generar comandos de velocidad que son enviados a una base móvil. Como requisito previo para el uso de la pila de navegación, el robot debe ejecutar ROS, tener un árbol de transformación tf en su lugar y publicar los datos del sensor utilizando los tipos de mensajes ROS correctos. Además, la pila de navegación debe configurarse para que la forma y la dinámica de un robot se desempeñe a un alto nivel.

Si bien esta pila está diseñada para ser de propósito general, es cierto que existen ciertos requerimientos referentes al hardware que deben cumplirse para funcionar de forma óptima:

1 Está diseñado para robots con ruedas de accionamiento diferencial y holonómicos únicamente. Se asume que la base móvil se controla enviando los comandos de velocidad deseados para lograrlo en forma de: velocidad x, velocidad y, velocidad theta,

2 Requiere un láser plano montado en algún lugar de la base móvil. Este láser se utiliza para la construcción y localización de mapas,

3 La pila de navegación se desarrolló en un robot cuadrado, por lo que su rendimiento será mejor en robots que sean casi cuadrados o circulares.

Funciona en robots de formas y tamaños arbitrarios, pero puede tener dificultades con robots rectangulares grandes en espacios estrechos como puertas.

Dada la naturaleza de la solución implementada, esta pila de navegación es perfecta dado que el robot principal que será utilizado para llevar a cabo las pruebas es un robot de tipo diferencial, holonómico y de forma cuadrada.

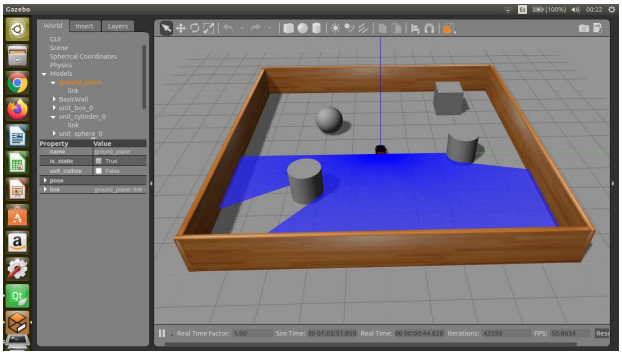


Fig. 1. Ambiente 3D construido en el simulador de Gazebo.

A continuación, se presenta la propuesta del framework, el punto central de nuestra aportación es el manejo eficiente y novedoso de la pila de navegación, que se resaltará en la sección de los experimentos realizados con el framework propuesto.

3. Propuesta del framework

Para poder implementar la pila de navegación en un robot con ROS se suele seguir una serie de pasos generales haciendo uso de diversos nodos y paquetes y que en varios casos requieren configuraciones específicas de parámetros, de acuerdo con el modelo del robot al que se le desea integrar el sistema de navegación. Se asume que el modelo del robot ya se encuentra en el servidor de parámetros de ROS.

El primer paso y el más fundamental antes de querer hacer cualquier cosa, es iniciar el proceso principal que administra el sistema ROS. Siempre se debe estar ejecutando el comando roscore en una consola independiente.

La pila de navegación funciona con o sin un mapa, sin embargo, la propuesta del framework requiere de un mapa a priori para realizar la tarea de navegación. Es aquí cuando se usa el paquete Gmapping y que a través del nodo slam_gmapping permite crear un mapa basado en las lecturas recibidas desde un sensor láser.

Suponiendo que se tiene el ambiente simulado mostrado en la Fig. 1, se requiere obtener un mapa de cuadrículas de ocupación 2D a partir de este, dicho mapa contendrá la información necesaria para realizar las tareas consecuentes de la navegación.

Para lograr la construcción del mapa, se debe contar con 3 terminales: la primera debe ejecutar el nodo slam_gmapping pasándole como parámetro el nombre del sensor laser montado en el robot, lo que permitirá empezar con el registro de los datos que se perciben con el sensor. Este nodo también pone a disposición el tema/map, lo que permite visualizar cómo es que se va creando el mapa mientras el robot se mueve.

La segunda terminal debe ejecutar un nodo que permita mover el robot a través del ambiente, suele ocuparse código personalizado para lograr esta tarea.

En este caso específico, se cuenta con un programa que permite controlar el movimiento del robot con las flechas del teclado, el cual se desarrolló como parte del trabajo.

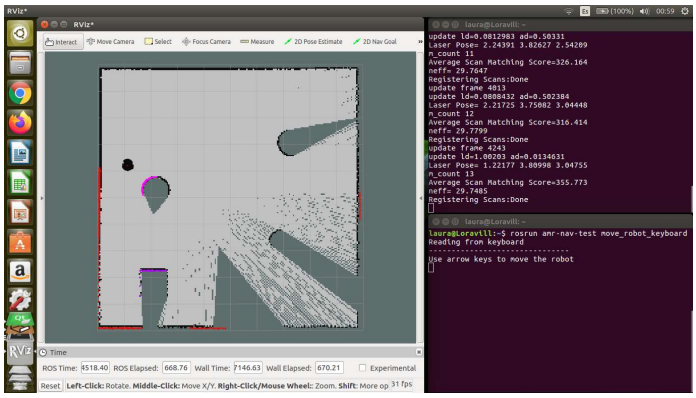


Fig. 2. Construcción de un mapa de rejillas 2D.

Antes de continuar con la tercera terminal, hasta este punto y con las dos consolas ejecutando sus nodos correspondientes se puede empezar a construir el mapa, la Fig. 2 muestra el panorama completo y la interacción entre Rviz y las consolas ya descritas. A medida que se mueve el robot a lo largo del mapa, este va coloreando, las áreas grises que representan el espacio ya explorado. Cuando el área completa se encuentre en color gris se debe guardar el mapa generado. Aquí entra otro nodo que deberá ser ejecutado en una nueva terminal, la tercera. Para poder guardar el mapa, se debe ejecutar el nodo map_saver del paquete map_server, pasándole como parámetro la ubicación y nombre del nuevo mapa.

Lo anterior genera dos archivos, uno con extensión pgm, la imagen del mapa y un archivo yaml relacionado que contiene metadatos del mapa. Una vez obtenida la imagen del mapa, los procesos que se encontraban ejecutando pueden ser detenidos, su propósito ya ha sido cumplido.Habiendo cumplido con los requisitos básicos, el primer paso para implementar el sistema de navegación en el robot es crear un paquete que contenga todas las configuraciones y archivos launch. Los archivos launch son, en esencia, archivos XML con las órdenes y parámetros para ejecutar uno o varios nodos en un solo paso, evitando así escribir los comandos directamente en las ventanas de línea de consola.

El primer archivo launch deberá lanzar los nodos necesarios para obtener los datos del sensor, la odometría del robot y la configuración de transformación del árbol tf, ya que, sin estos, el sistema de navegación simplemente no funcionará.

Dentro de este paquete, también es necesario contar con los archivos de configuración de mapa de costos, entonces se determinan cuáles serán los planificadores global y local para utilizar. Estos archivos son de extensión yaml y contienen los parámetros configurados más adecuados al robot en uso.

De forma general se ocupan un total de 3 archivos, el primero con las configuraciones comunes de los mapas de costos, los otros corresponden a la configuración de costos global y local. Además, se tiene también un archivo con la configuración del base_local_planner, que es responsable de calcular los comandos de velocidad para enviar a la base móvil del robot dado un plan de alto nivel.

3.1. Ejecutar la pila de navegación

Este paso es el resultado de todas las acciones realizadas con anterioridad. Recapitulando, se cuenta con un mapa de ocupación de rejilla 2D con las características del ambiente en el que se encuentra el robot y se cuenta con los archivos con las configuraciones requeridas para el nodo move_base. Lo que sigue es cargar el mapa al servidor, ejecutar el nodo amcl para la localización y finalmente ejecutar el nodo move_base.

Entonces, se debe cargar el mapa del ambiente en el que se desea navegar, esto se realiza mediante el nodo map_server del paquete map_server, la ubicación del mapa a mostrar es un parámetro que debe ser establecido antes de ejecutar el nodo.

Lo siguiente es iniciar la localización a través del nodo amcl, al hacerlo también se habilitará el frame map, que es la referencia en la que se basa la odometría del robot con respecto a un mapa. Este nodo recibe las transformaciones necesarias desde el frame base para poder mantenerse localizado correctamente. En este sentido, amcl se mantiene publicando en el tema/particlecloud el conjunto de estimaciones de pose que mantiene el filtro.

Al haber realizado la anterior, es posible agregar un elemento PoseArray en Rviz que permitirá subscribirse al tema/particlecloud, con esto se habilitarán flechas direccionales. Para saber si la localización se está realizando de forma correcta, basta con ver la dispersión de las flechas, si se concentran en un lugar cerca del robot quiere decir que la precisión es bastante alta, de lo contrario es una señal de que el robot no se encuentra bien localizado dentro del mapa, lo que derivará en errores de cálculo para la navegación. La Fig. 3 muestra la interacción de la ventana Rviz con las ventanas de comandos que ejecutan los diversos nodos.

Para finalizar, el último nodo que se debe ejecutar es el nodo move_base, es el más importante ya que carga con las configuraciones de los mapas de costos y los parámetros del planificador base local. Dado que, move_base requiere de diversos parámetros, los cuales están declarados en los archivos yaml, lo mejor es ejecutar este nodo desde un archivo launch. Una vez iniciado el nodo move_base, nuevos temas se ponen a disposición, por lo tanto, es posible configurar la vista de dos elementos de tipo mapa en Rviz.

El primero será un mapa estático definido dentro del mapa de costos global. El segundo mapa enfatiza las lecturas del sensor y se basa sobre el mapa de costos local, de hecho, es posible visualizar de manera más robusta los distintos obstáculos en el mapa. La Fig. 4 muestra la interacción de la ventana de comandos que se encuentra ejecutando el archivo launch y la ventana Rviz con sus diversos elementos.

A partir de este momento, ya es posible definir puntos en el mapa como metas para que la pila de navegación calcule un camino y envíe los comandos de velocidad a la base para realizar el recorrido hacia el objetivo. Para visualizar el camino calculado se debe agregar un elemento de tipo Camino (Path) que se suscribe al tema con el nombre del planificador base local. En la Fig. 5 se observa la adición de este elemento al panel de Rvix, como se nota, cuando se señale un punto meta y se genere un camino, este se mostrará como una línea de color azul.

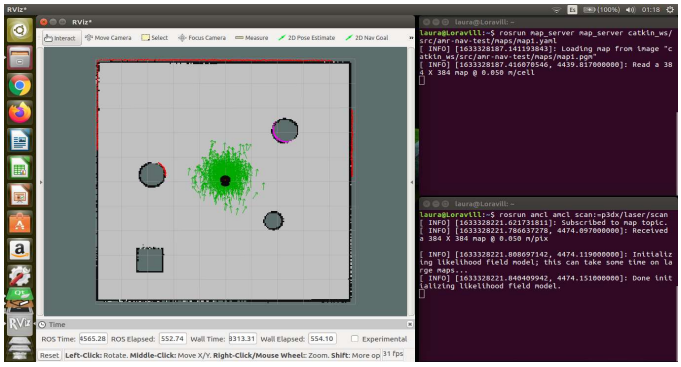


Fig. 3. Ejecución del nodo map_server y amcl.

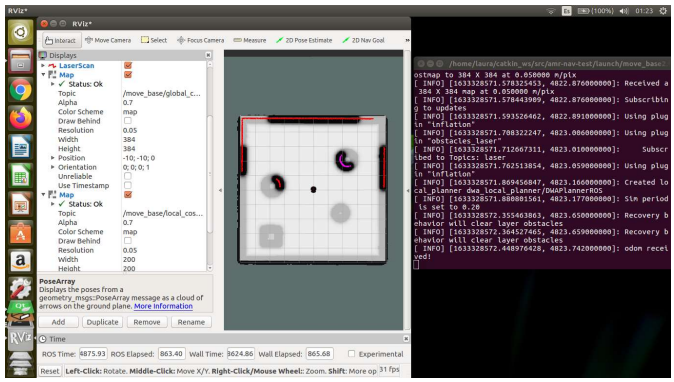


Fig. 4. Interacción del modo move_base con Rviz.

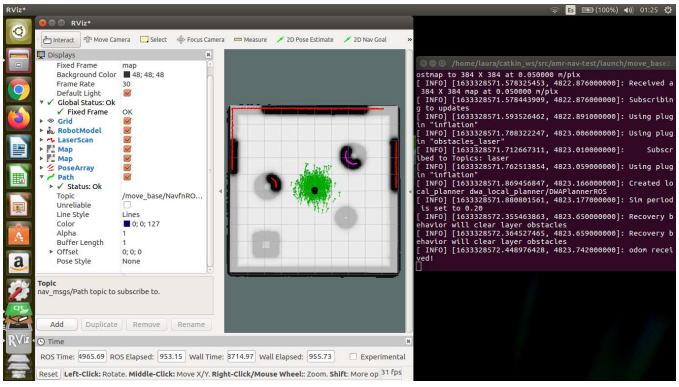


Fig. 5. Adición del elemento Path en Rviz.

Rviz incluye la funcionalidad de establecer un punto meta en el mapa, lo que envía la señal para calcular el camino que conecta los puntos de partida y final.

Lo que ocurre a continuación es la generación del camino, y el envío de los comandos de velocidad a la base móvil hasta que el robot llega a la posición meta.

Como se puede observar se ha implementado satisfactoriamente la pila de navegación en el robot Pioneer 3DX. Sin embargo, es claro que para lograrlo se tuvieron que ejecutar diversos nodos de forma separada en un total de 7 terminales, sin mencionar la instancia de Rviz presente a la cual se le tuvieron que ir agregando diversos elementos conforme se fue avanzado en el proceso. Pese a que ROS provee un sistema de navegación para robots diferenciales, es un hecho que reina el caos cuando se trata de implementar una pila de navegación desde cero, más aún, para personas con poco conocimiento en ROS, ¡es todo un reto! Esto es parte de nuestra contribución.

Es por ello, que la propuesta que se presenta en este trabajo toma todos los procesos, ejecuciones, y demás, y las embebe en una sola ventana, intuitiva y con configuraciones y modelos precargados para evitar todo el proceso que conlleva la pila de navegación y centrarse únicamente en observar el funcionamiento mediante él envió de distintas posiciones meta.

4. Resultados experimentales

Se propone la creación de un framework, AMR NAV (Autonomous Mobile Robot Navigation), basado en ROS para la navegación autónoma de robots móviles, tomando como punto de partida el paquete navigation stack. Este framework busca reunir todas las actividades que se deben llevar a cabo, como parte de la pila de navegación, en un solo ecosistema que minimice el tiempo necesario para configurar un sistema de este tipo en un robot, al mismo tiempo que muestra la relación directa de estos pasos en una visualización 3D. Además. Busca ser una herramienta de aprendizaje para los nuevos y antiguos interesados en el mundo de la robótica y de ROS.

AMR NAV interactúa con los nodos que provee ROS como parte de su pila de navegación, los nodos propios que le permiten gestionar los procesos descritos anteriomente y la simulación de Gazebo.

De forma general, la Fig. 6 muestra la arquitectura de alto nivel de AMR NAV y su interacción con los sistemas antes mencionados. Como se puede observar, la propuesta realizada provee un sistema para controlar los procesos detrás de cada paso para la navegación a través de una interfaz de usuario.

Las Fig. 7 y Fig. 8 muestran el entorno de la ventana principal de este framework. En la primera se muestran los dos primeros pasos del navigation stack, en la otra los dos faltantes. En esta parte se presentan los resultados obtenidos al utilizar AMR NAV para ejecutar una pila de navegación en un robot, las características que se buscan comparar con respecto al modo tradicional son principalmente el tiempo para completar los pasos y la facilidad de estos. También, se probará la efectividad en la obtención del camino desde un punto inicial y hasta un punto final, y si los comandos de velocidad generados logran completar el recorrido.

TurtleBot es un kit de robot personal de bajo costo con software de código abierto. Fue creado en Willow Garage por Melonee Wise y Tully Foote en noviembre de 2010. El kit TurtleBot consta de una base móvil, un sensor 3D, una computadora portátil y el kit de hardware de montaje TurtleBot.

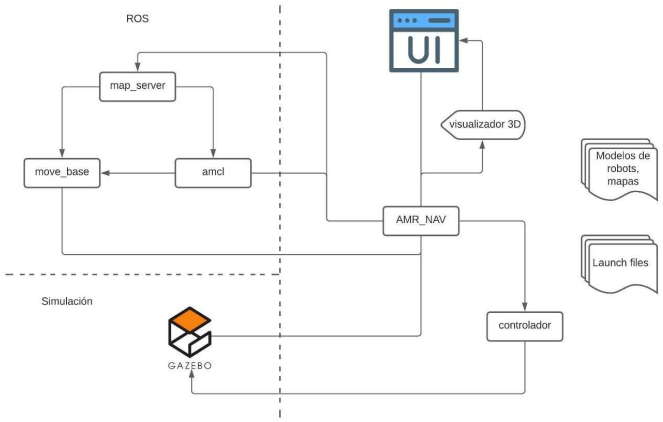


Fig. 6. Arquitectura de alto nivel de AMR NAV.

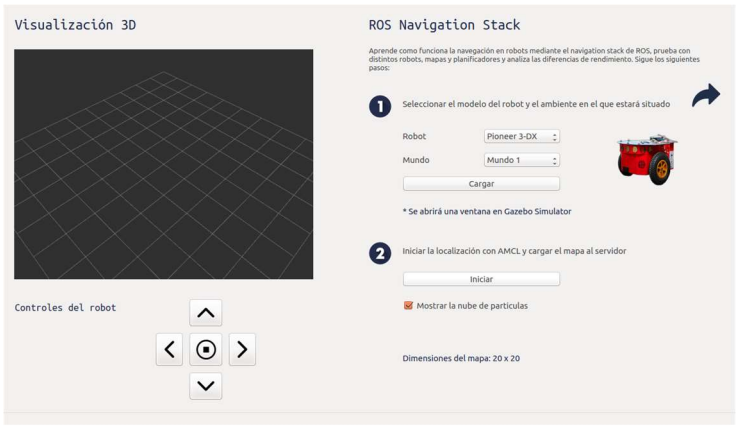


Fig. 7. Ventana principal de AMR NAV, parte 1.

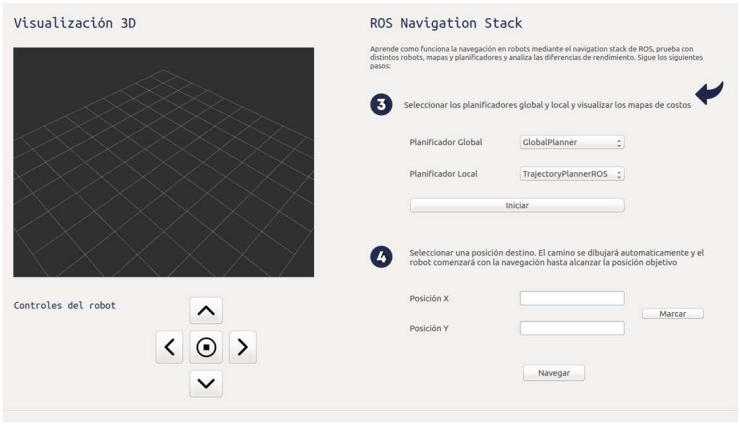


Fig. 8. Ventana principal de AMR NAV, parte 2.

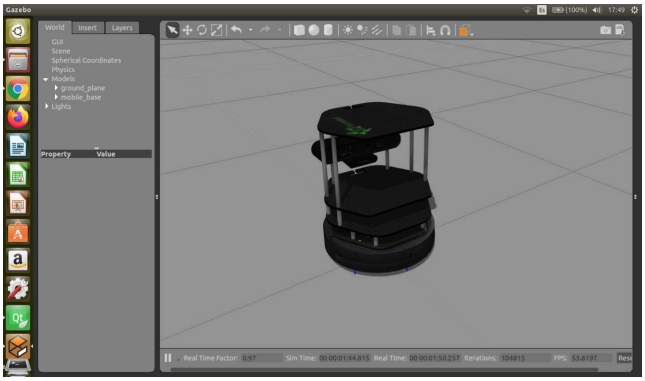


Fig. 9. Modelo del robot TurtleBot en Gazebo.

La Fig. 9 muestra el modelo de este robot en el Gazebo. AMR NAV pone a disposición el uso de este robot para la aplicación de la pila robótica. Para este robot se diseñaron 4 casos de prueba con distintas configuraciones, por motivos de brevedad se presentarán solo los resultados mediante imágenes. Lo importante a destacar, son los diferentes tipos de mapas utilizados, desde los fáciles (con pocos obstáculos) hasta los complicados (con obstáculos estrechos, por ejemplo); igualmente otro punto a resaltar fue el cambio de los planificadores tanto local como global, esto con la finalidad de evaluar el desempeño del robot en relación con su ambiente de trabajo.

Para los casos de prueba propuestos, usamos los siguientes planificadores local y global. NavfnROS (global) y DWAPlannerROS (local) para el primer caso de prueba. GlobalPlanner (global) y TrajectoryPlannerROS (local) para el segundo caso de prueba. NavfnROS (global) y DWAPlannerROS (local) para el tercer caso de prueba. GlobalPlanner (global) y TrajectoryPlannerROS (local) para el cuarto caso de prueba. Ver la Fig. 10 para los resultados obtenidos con dos mapas diferentes.

El robot Pioneer 3DX, es un robot pequeño y ligero de tipo diferencial con dos ruedas y dos motores. Este robot es de los más populares para su uso en el área educativa y de investigación en laboratorios. La Fig. 11 muestra el modelo del Pioneer 3DX en Gazebo. Dado que AMR NAV permite estandarizar lo más posible alguna de las configuraciones básicas de la pila de navegación, es interesante ver cómo estas configuraciones pueden interferir en los resultados al utilizar un robot diferente. Por ello, se crearon dos casos de prueba que permitan ilustrar las diferencias e identificar las posibles causas de una falla.

Para los casos de prueba propuestos, usamos los siguientes planificadores local y global. NavfnROS (global) y DWAPlannerROS (local) para el primer caso de prueba. GlobalPlanner (global) y TrajectoryPlannerROS (local) para el segundo caso de prueba. En la Fig. 11 se muestran los resultados de esta segunda prueba con el AMR NAV.

Los resultados obtenidos sugieren que, al menos para los dos modelos de robots utilizados, las configuraciones incorporadas en AMR NAV permiten ejecutar una pila de navegación sin mayor problema. Esto refuerza la viabilidad de la propuesta presentada, pues en efecto, logró simplificar y generalizar parámetros básicos requeridos por los distintos nodos ejecutados.

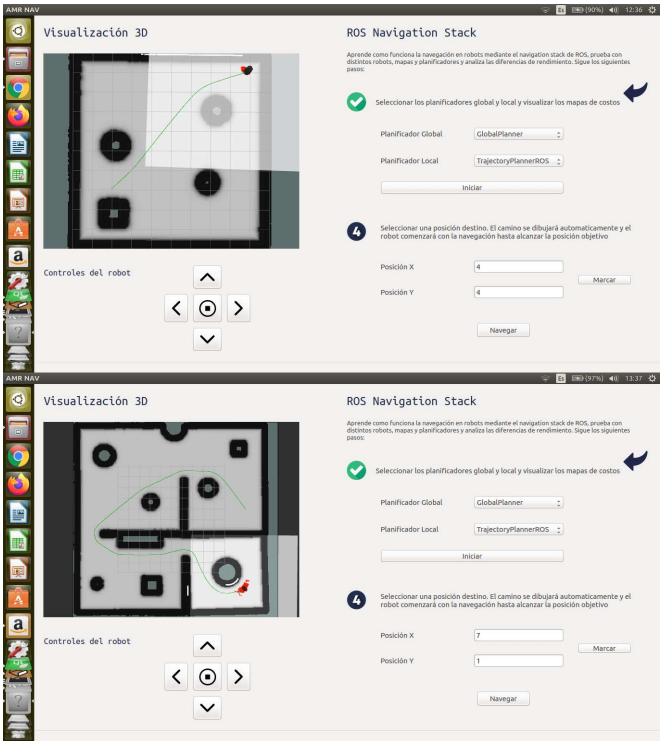


Fig. 10. Resultados obtenidos en AMR NAV con el robot TurtleBot.

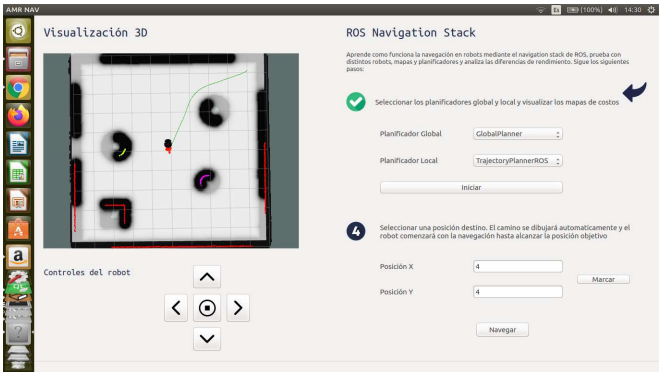


Fig. 11. Resultados obtenidos en AMR NAV con el robot Pioneer 3DX.

De lo anterior, también se demuestra que el planificador global NavfnROS y el planificador local DWAPlannerROS tienen un mejor rendimiento comparado con los otros planificadores utilizados.

Es importante señalar que las configuraciones en los mapas de costos pudieron haber interferido notablemente en el desempeño del planificador local, lo cual provocó errores en algunos casos.

Los resultados son de hecho favorables, pues comprueban que AMR NAV logró centralizar y extender los procesos necesarios para utilizar una pila de navegación. Además, el hecho de contar con los modelos de los robots, mapas y archivos de configuración para su uso inmediato, logran reducir notablemente el esfuerzo y tiempos que serían requeridos de hacerlo de forma tradicional.

Para el desarrollo y pruebas del framework propuesto en este trabajo, se utilize una computadora PC basada en el procesador AMD Ryzen 5 2500 U, con tarjeta gráfica AMD Radeon Vega 8, utilizando el Sistema operative Linux, distribución Ubuntu 14.04. Es claro que, como parte de un siguiente trabajo, se presentarían las ejecuciones en tiempo real, utilizando un robot real. Y es ahí cuando valdrá la pena realizar un estudio comparativo más detallado.

5. Conclusiones y trabajo futuro

En este trabajo se propuso un framework para la navegación de robots autónomos basado en ROS, considerando que el proceso tradicional puede ser complicado de lograr, en especial para aquellos que son nuevos en el uso del Sistema Operativo Robótico. Y es que, a pesar de contar con una comunidad con suficientes miembros, la verdad es que muchos de los foros de discusión para aclarar dudas se encuentran con temas o preguntas sin resolver.

Como resultado se desarrolló AMR NAV que integra una pila de navegación basada en mapas que ROS provee como parte de sus paquetes y librerías.

De este modo, AMR NAV tomó todas esas horas de investigación y práctica y las convirtió en un proceso de sólo 4 pasos para establecer el sistema de navegación en un robot. En contraste con la forma tradicional, permitió solucionar este problema en menos de 5 minutos.

Finalmente, y aunque no fue liberado para su uso y prueba, se espera que este trabajo signifique una buena contribución para todos aquellos han tenido dificultades para utilizar la pila de navegación, al igual que muchos otros. Estamos en la depuración del código y otros detalles para su posterior liberación.

En este punto, AMR NAV es un prototipo que puede y debe ser mejorado mediante la contribución de la comunidad. A continuación, se presentan algunos posibles trabajos futuros que pueden desarrollarse a partir de la investigación y trabajo realizados, o que, por exceder el alcance de esta tesis no han sido considerados en primera instancia, sean:

- Ampliar el número de modelos y mundos disponibles actualmente,
- Agregar algoritmos propios para los planificadores mediante la creación de plugins compatibles con ROS, de este modo, la lista actual no debería ser limitada, pudiendo ser agregados otros tipos de algoritmos comúnmente utilizados en la literatura como RTT,
- Examinar otras formas de localización para obtener un mejor rendimiento en los cálculos realizados por el planificador local,
- Implementar la navegación en robots sin tener el conocimiento del mapa con anticipación, claro que esto incluiría investigación más especializada sobre SLAM,

- Explorar otras áreas de la robótica e incluirlas en este framework como una extensión del propósito original.

Referencias

1. Dellaert, F., Fox, D., Birgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: Proceedings 1999 IEEE International Conference on Robotics and Automation, vol.2, pp. 1322–1328 (1999) doi: 10.1109/ROBOT.1999.772544
2. Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H., Csorba, M.: A solution to the simultaneous localization and map building problem. IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 229–241 (2001) doi: 10.1109/70.938381
3. Karur, K., Sharma, N., Dharmatti. C., Siegel, J.: A survey of path planning algorithms for mobile robots. Vehicles (www.mdpi.com/journal/vehicles). vol. 3, no. 3, pp. 448–468 (2021) doi: 10.3390/vehicles3030027
4. Lentin, J., Cacace, J.: Mastering ROS for robotics programming. Packt Publishing (2021)
5. Leonard, J. J., Durrant-Whyte, H. F.: Mobile robot localization by tracking geometric beacons. IEEE Trans. Rob. Autom., vol. 7, no. 3, pp. 376–382 (1991)
6. Muñoz M. V.: Planificación de trayectorias para robots móviles. Tesis del Departamento de Ingeniería de Sistemas y Automática, Universidad de Malaga – España (1997)
7. Robot operating system (https://www.ros.org/) (2022)
8. Ronquillo, J. B.: Hands-on ROS for robotics programming. Pack Publishing (2020)
9. Siegwart, R., Rez, N. I., Scaramuzza, D., Arkin, R. C.: Introduction to autonomous mobile robots. MIT Press, 2nd Edition (2011)
10. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press (2005)

ISSN 1870-4069

Detección de grietas en concreto con histogramas de gradientes orientados

Perla Lisseth Hernández Ortega[1], Marco Cesar Salomón González[1], Pedro Arguijo[2], Roberto Ángel Meléndez Armenta[2], José Antonio Hiram Vázquez López[1]

[1] Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
División de Ingeniería en Sistemas Computacionales,
México

[2] Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
División de Estudios de Posgrado e Investigación,
México

{182t0432, 182t0175, ramelendeza, jahvazquezl}@itsm.edu.mx,
pedroarguijo16@gmail.com

Resumen. La identificación de grietas es un problema que requiere la participación humana y la identificación manual de las mismas. La detección temprana de grietas en las construcciones civiles, así como evaluación objetiva es de suma importancia para el mantenimiento de la misma pues éstas pueden indicar daños estructurales internos. En este trabajo se presenta un sistema de detección de grietas basado en un vector de características de HOG. La exactitud general del clasificador SVM, con las características extraídas, fue de un 98%. Este resultado indica que el enfoque propuesto puede detectar efectivamente grietas en las imágenes. Los resultados obtenidos son similares a los reportados previamente con métodos de aprendizaje profundo.

Palabras clave: Detección de grietas, HOG (Histograma de Gradientes Orientados), SVM.

Crack Detection in Concrete with Oriented Gradient Histograms

Abstract. Crack identification is a problem that requires human participation and manual crack identification. The early detection of cracks in civil constructions, as well as an objective evaluation, is of utmost importance for their maintenance as they can indicate internal structural damage. In this paper, a crack detection

system based on a HOG feature vector is presented. The overall accuracy of the SVM classifier, with the extracted features, was 98%. This result indicates that the proposed approach can effectively detect cracks in images. The results obtained are similar to those previously reported with deep learning methods.

Keywords: Crack detection, HOG, SVM.

1. Introducción

El concreto u hormigón armado, es un material ampliamente utilizado en la construcción de edificios debido a su versatilidad, resistencia y bajo costo, convirtiéndose en uno de los materiales más empleados a nivel mundial [1].

Es de piedra formada de manera artificial a partir de áridos naturales (grava y arena) pegados por una pasta conglomerada hidráulica, principalmente cementos, adherida fuertemente a armaduras de acero, comúnmente varillas corrugadas de acero [2].

En la ingeniería civil, el concreto es usualmente utilizado para los trabajos de construcción debido a que posibilita la elaboración de múltiples estructuras resistentes a la compresión, a la tracción y a la flexión. De esta manera, si la elaboración de la estructura es adecuada, el material puede llegar a tener una duración permanente sin necesidad de contar con medidas de protección y conservación especiales, siempre y cuando las condiciones ambientales sean normales [2].

Las grietas y fisuras son una de las patologías más características y relevantes que afectan a las estructuras de hormigón armado como pavimentos, puentes, techos y paredes de túneles, por mencionar algunas. Estas son ocasionadas, entre otros factores, debido a tensiones que sobrepasan su capacidad de resistencia, a fluctuaciones de temperatura, al envejecimiento de la superficie e impactos ambientales.

Por su parte, las fisuras son daños superficiales de poca profundidad que no afectan estructuralmente a las construcciones; mientras que las grietas son daños con una profundidad considerable, llegando a cubrir hasta la totalidad del grosor de la estructura. Además, se consideran que son un indicador importante que refleja la degradación y el estado de seguridad de la infraestructura.

La expansión, contracción, sobrecarga, asentamiento o secado prematuro son causas comunes de grietas en la superficie de concreto. Las grietas pueden aparecer aisladas o como un grupo de varias grietas conectadas.

En el pavimento este tipo de deterioro se puede clasificar en función de varios parámetros, como su posición, su dirección y su forma [3, 4]. Su apariencia relativamente similar hace que los algoritmos de análisis de imágenes sean una herramienta ideal para su detección, lo cual conlleva a una prueba no destructiva.

La detección de grietas es el paso más importante durante la inspección, diagnóstico, mantenimiento y predicción de la vida útil de las estructuras civiles de concreto.

Además, reparar a tiempo las grietas evita daños mayores ya que el desarrollo y propagación de las grietas reduce primero la eficacia de una zona de carga, lo que aumenta la presión y, finalmente, daña la estructura [5].

El método convencional de detección de grietas se basa en inspectores capacitados que encuentran grietas en la superficie de una estructura de concreto, basándose en sus conocimientos y años de experiencia. Este método de detección de grietas no sólo requiere mucho tiempo y trabajo, sino que a menudo no es fiable porque los resultados dependen de la experiencia y la habilidad del inspector; lo que puede dar lugar a una supervisión o inspección inexacta [6, 7]. Para facilitar el progreso de la inspección visual, es necesario lograr la detección automática de grietas.

En comparación con el sistema tradicional de detección de grietas basado en la inspección visual, los enfoques basados en la visión artificial y el aprendizaje automático se están adoptando en la inspección de infraestructuras civiles para automatizar la detección e identificación de grietas [8-10]. Una revisión puntual de la detección de grietas basada en la visión artificial y la evaluación de los avances significativos de la infraestructura civil se encuentra en [11].

La detección de grietas de manera automática, permite entre otras cosas el ahorro de tiempo y personal, debido a que no se necesita que un inspector se traslade al terreno para identificar las grietas. Además, esto permite que la inspección se estandarice, es decir, que no existan dobles resultados de inspecciones de distintos inspectores.

Por otro lado, la automatización en la identificación de grietas permite la observación de grietas con acceso restringido o en lugares de difícil acceso, con el consiguiente ahorro económico.

Los métodos de aprendizaje profundo, específicamente las soluciones basadas en redes neuronales convolucionales (CNN), exhiben un rendimiento notable en el reconocimiento de dígitos, clasificación de imágenes, anotación de imágenes y otros campos relacionados. Las CNN son redes multicapa que aprenden, extraen y clasifican características automáticamente. Esto difiere de otros enfoques populares basados en la extracción de las características, como GLCM, Wavelets, LBP.

Inspirados en el rendimiento de los modelos de aprendizaje profundo, algunos estudios recientes han reportado el rendimiento de los modelos basados en CNN para la detección de grietas. Se han propuesto arquitecturas específicas y también se ha utilizado la transferencia de aprendizaje para la detección de las grietas [12-14]. Aunque no cabe duda de que las características profundas extraídas automáticamente a través de los CNN ofrecen una mejor capacidad de representación que las características extraídas de manera tradicional cabe preguntar, ¿se debe abandonar la extracción de características y utilizar únicamente las características profundas? Creemos es una pregunta significativa por diversas razones.

Previo a la adopción generalizada del aprendizaje profundo, la extracción de diversos conjuntos de características era un problema a resolver para la clasificación de imágenes. Aunque el proceso de extracción de características actúa como un puente vital entre la imagen sin procesar y la información discriminatoria de los vectores de características extraídos, la calidad de las características determina en gran medida la precisión de la clasificación.

Las características que funcionan bien en un entorno no son adecuadas en los demás. Consecuentemente, la selección de las características es especialmente importante. Por un lado, si se desea implementar una arquitectura de deep-learning desde cero, se debe disponer de una gran cantidad de datos para el entrenamiento de la misma.

Así mismo, al tener una arquitectura de deep-learning adecuada se puede caer en un ciclo de modificaciones de la misma para mejorar su rendimiento. Si se utiliza una arquitectura pre-entrenada como extractor de características cada una de las capas conectadas describe la imagen de entrada con un diferente nivel de abstracción, esto sin considerar la gran dimensionalidad del conjunto de datos.

En este trabajo, determinamos la presencia o no presencia de grietas en imágenes de concreto, en particular, consideramos el conjunto de datos públicos Concrete Crack Images for Classification [15]. Nuestro enfoque se basa en la extracción de características de Histogramas de Gradientes Orientados (HOG), como descriptor adecuado para la detección de grietas.

Se eligió HOG dado que, al operar en celdas locales, es invariable a transformaciones geométricas y fotométricas, y esto lo hace adecuado para la detección de grietas en pavimento. Además del vector de características, el clasificador tiene una gran influencia en el rendimiento de la detección de grietas.

En nuestro caso seleccionamos una máquina de soporte vectorial (SVM) debido a que está reportado en la literatura que esta combinación, HOG-SVM, tiene un buen desempeño en diversas tareas de clasificación. Cabe mencionar que se logró una alta exactitud de clasificación en la detección de las grietas. El clasificador se verificó con validación cruzada.

El documento está estructurado como se indica: los detalles del conjunto de datos utilizados, la extracción de características y el clasificador se dan en la siguiente sección. La sección 3 describe los resultados y se finaliza en n la sección 4 con las conclusiones.

2. Materiales y métodos

2.1. Conjunto de datos

En estudios previos se han referenciado diversos conjuntos de datos (públicos y privados) para la detección automática de grietas en pavimentos [8-14]. La mayoría emplea conjuntos de datos no públicos, lo que complica establecer un punto de comparación entre las diversas metodologías propuestas.

Además, se debe considerar que en la creación de un conjunto de datos las imágenes deben tomarse de manera homogénea manteniendo condiciones constantes, tales como distancia, ángulo, etc. Por esta razón, en este trabajo utilizamos el conjunto de datos de código abierto de imágenes de grietas en concreto de la METU [15] para identificar la presencia o ausencia de grietas. Este conjunto de datos consta de un total de 40000 imágenes con una resolución de 227×227 pixeles, que se dividen por igual en las clases con grietas y sin grietas.

Las imágenes se tomaron en ensayos estructurales de laboratorio bajo condiciones de iluminación casi uniformes.

La Figura 1 muestra un ejemplo del conjunto de datos utilizado. De dicho conjunto de datos se seleccionaron aleatoriamente 1000 imágenes con grietas y 1000 imágenes sin grietas.

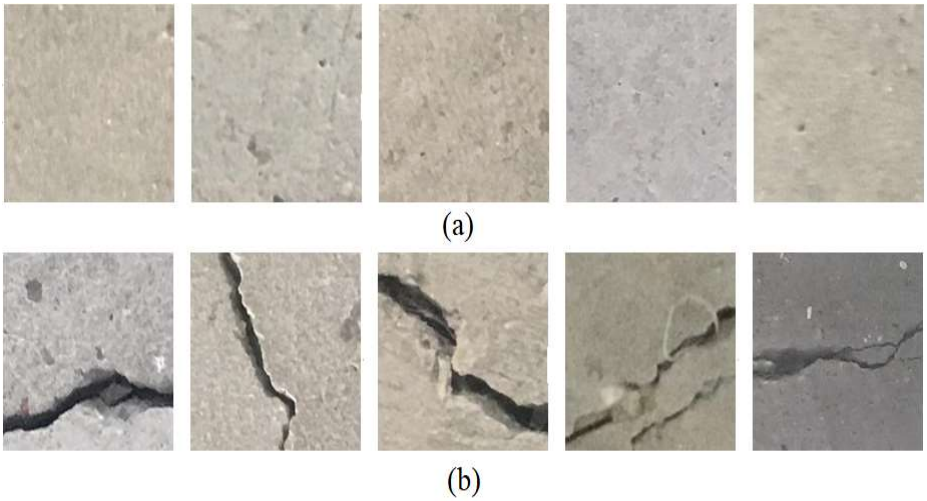


Fig. 1. Muestra del conjunto de imágenes consideradas con diferentes condiciones de iluminación (a) sin grietas y (b) con grietas.

2.2. Histograma de gradientes orientados (HOG)

La extracción de características es la parte más importante de cualquier sistema de reconocimiento. La transformación de los datos de entrada en un conjunto de características se denomina extracción de características.

Para este trabajo el Histograma de Gradientes Orientados (HOG, por sus siglas en inglés) [16] se consideró como el extractor de características para cada una de las imágenes seleccionadas.

Aunque inicialmente se propuso para la detección de peatones, se ha utilizado para diversos problemas de clasificación como detectar células humanas [17], en la detección de vehículos [18], así como en el reconocimiento facial [19].

HOG se basa en la suposición de que el aspecto local y la forma de un objeto pueden caracterizarse por la distribución de gradientes de intensidad local o direcciones de bordes, que por definición son perpendiculares a la dirección del gradiente.

En otras palabras, la distribución de los bordes, que están representados por los gradientes de intensidad, proporciona una caracterización de la forma sin un conocimiento directo de las posiciones de los bordes correspondientes. El proceso para obtener HOG de una imagen se muestra en la Figura 2.

Como se puede observar, se inicia calculando los gradientes de la imagen en las direcciones x e y sin suavizar. A continuación, la imagen se divide en una cuadrícula de bloques, cada uno de los cuales se divide en una cuadrícula de celdas.

Para cada píxel en la celda correspondiente, la orientación y magnitud se calculan en función de los gradientes de intensidad, y los histogramas orientados se votan con ponderación según las magnitudes de los gradientes.

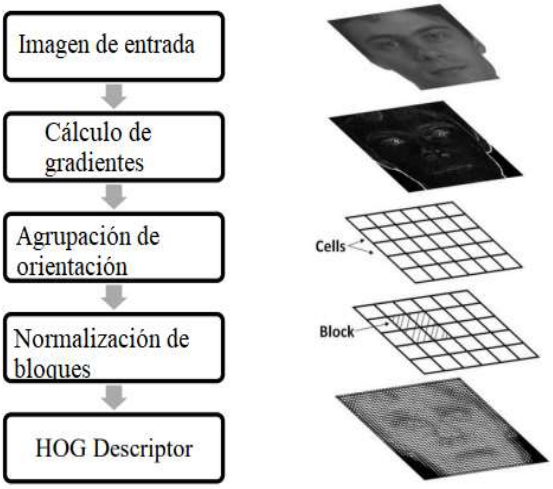


Fig. 2. Proceso de obtención del descriptor de características HOG.

Las frecuencias en los histogramas se normalizan en el intervalo [0, 1] para compensar los cambios en la iluminación. La combinación de los histogramas de todos esos bloques representa los descriptores HOG.

Para este trabajo, utilizamos seis contenedores de histogramas que almacenan las direcciones de los bordes y sus intensidades, para cada celda de 8×8. Con lo cual se obtienen 384 características por imagen.

2.3. Máquina de soporte vectorial

Las máquinas de soporte vectorial (SVM por sus siglas en inglés) pertenece al tipo de aprendizaje supervisado y se utiliza tanto para problemas de clasificación como de regresión. Es un sistema de aprendizaje que separa un conjunto de vectores de entrada en dos clases con un hiperplano de separación óptimo.

Se dice que el conjunto de vectores está separado de manera óptima, por el hiperplano, si está separado sin error y la distancia entre los vectores más cercanos al hiperplano es máxima.

El clasificador SVM se obtiene aplicando una variedad de funciones de núcleo, también conocido como kernel; lineal, polinomial, funciones de base radial, como posibles conjuntos de funciones de optimización.

Su base se encuentra en la solución de un problema de programación cuadrática dual y utilizando la minimización del riesgo estructural como principio inductivo, a diferencia de los algoritmos estadísticos clásicos que maximizan el valor absoluto de un error o de un error al cuadrado.

Tabla 1. Desempeño del modelo de clasificación considerando diversos valores de k.

k	Exactitud	Precisión	Recall
2	0.9855	0.9856	0.9855
4	0.9895	0.9895	0.9896
5	0.989	0.989	0.9891
8	0.9895	0.9895	0.9896
10	0.988	0.988	0.988

Tabla 2. Resultados de las métricas de evaluación con k = 10.

Fold	Exactitud	Precisión (CG)	Precisión (SG)	Recall (CG)	Recall (SG)
1	0.985	0.9815	0.9891	0.9906	0.9785
2	0.99	1	0.9809	0.9794	1
3	0.985	0.9811	0.9897	0.9905	0.9789
4	1	1	1	1	1
5	0.985	0.9794	0.9903	0.9896	0.9808
6	0.98	0.96	1	1	0.9615
7	0.985	0.9780	0.9908	0.9889	0.9818
8	0.99	0.9909	0.98889	0.9909	0.98889
9	0.985	0.9906	0.9787	0.9813	0.9892
10	0.995	1	0.9908	0.9899	1
Promedio	0.988	0.9861	0.9899	0.9900	0.9859

3. Resultados

El problema de la detección automática de grietas puede formularse como un problema de clasificación binaria que se resuelve clasificando las características de HOG con SVM. Con este fin y para asegurar el significado estadístico de los resultados y, además, verificar el desempeño del modelo se aplicó validación cruzada con $k = 2$, [4, 5, 8 y 10].

En la validación cruzada, los datos de entrenamiento se dividen aleatoriamente en k subconjuntos mutuamente excluyentes de aproximadamente el mismo tamaño. La regla de decisión se obtiene usando $k-1$ de los subconjuntos y luego se prueba en el subconjunto que queda fuera. Este procedimiento se repite k veces y de esta manera cada subconjunto se usa para probar una vez. Promediar el error de prueba sobre los k intentos da una estimación del error de generalización esperado.

Las métricas consideras para determinar la eficiencia del clasificador junto con las características de HOG fueron la precisión, el recall, la exactitud y f1-score. La precisión informa sobre las predicciones correctas realizadas a partir de falsos positivos, mientras que el recall informa sobre las predicciones precisas realizadas a partir de falsos negativos. La exactitud es el número de predicciones correctas entre los falsos positivos y los falsos negativos. Todas las métricas de rendimiento para los modelos entrenados se determinaron como se indica:

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP+FN},$$

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP},$$

donde TP son verdaderos positivos, TN son verdaderos negativos, FP son falsos positivos y FN son falsos negativos. Aquí, TP y TN son las predicciones correctas, mientras que FP y FN son las predicciones incorrectas realizadas por los modelos.

En la Tabla 1 se muestra los resultados promedios obtenidos para cada una de las métricas consideradas para los valores de *k* indicados. Como se puede apreciar los resultados son similares sin importar el número de *k*.

En la Tabla 2 se muestran los resultados de la validación cruzada con k = 10. La CG y SG tanto en la precisión como en el recall indican, con grieta y sin grieta, respectivamente. Como se puede observar, en el valor promedio, el método propuesto puede distinguir entre imágenes con grietas y sin grietas con una exactitud del 98%.

Nuestros resultados se comparan con resultados previamente reportados con el mismo dataset, pero haciendo uso de transferencia de aprendizaje a través de CNN pre-entrenadas [20]. Con estos resultados concluimos que la detección de grietas en concreto puede analizarse a través de la extracción tradicional de características en conjunto con un clasificador también tradicional. Obviamente, falta analizar el tipo y orientación de grietas, pero consideramos que si se incrementan el número de orientaciones en el contenedor esto es posible.

4. Conclusiones y trabajo a futuro

En este trabajo se exploró el uso de características tradicionales, como lo es HOG, al problema de detección de grietas en imágenes de concreto; esto con la finalidad de utilizarlas en la inspección in situ de una infraestructura. La principal contribución de nuestro trabajo es el estudio y demostración de que las características de HOG son bastante robustas para determinar la presencia o ausencia de grietas. El tamaño vector considerado fue de 384 características por imagen, ya que se consideraron celdas de 8x8 y seis orientaciones.

Los resultados demuestran que el enfoque propuesto, HOG-SVM, tiene una buena capacidad para discriminar entre la presencia y ausencia de grietas, con una exactitud, precisión y recall del 98%. Estos resultados son similares a los reportados previamente con métodos de aprendizaje profundo.

Aunque este documento se centra en las características de HOG y SVM como clasificador, creemos que, con investigación adicional, otras técnicas tradicionales de extracción de características pueden tener un papel importante en la detección de grietas. Esto conlleva varios desafíos y direcciones de investigación. Por ejemplo, se podría aplicar algoritmos de segmentación como fase adicional o paralela para determinar la longitud de las grietas.

También podrían emplearse otras técnicas de aprendizaje automático y esto daría un comparativo en la detección y probable tipo de grietas.

Referencias

1. Serrano-González, L,. Merino-Maldonado, D. A., Antolín-Rodríguez, P. C. Lemos, A. S. Pereira, P. Faria, A. Juan-Valdés, J. García-González, J. M. Morán-Del cPozo: Biotreatments using microbial mixed cultures with crude glycerol and waste pinewood as carbon sources: influence of application on the durability of recycled concrete. Materials 15/3, no. 1181 (2022) doi: 10.3390/ma1503118
2. Medina-Sánchez, E.: Construcción de estructuras de hormigón armado en edificación. Tercer Edición, Editorial Bellisco (2014)
3. Téllez Gutiérrez, R.: Catalogo de deterioros en pavimentos flexibles de carreteras mexicanas. Publicación Técnica no. 21 (1991) https://imt.mx/archivos /Publicaciones/ PublicacionTecnica /pt21.pdf.
4. Consejo de directores de carreteras de Iberia e Iberoamérica, M5.1. Catálogo de deterioros de pavimentos flexibles, Colección de documentos, vol. no. 11 (2002) https://sjnavarro. files. wordpress.com/2008/08/manualfallas.pdf.
5. Bayar, G., Bilir, T.: A novel study for the estimation of crack propagation in concrete using machine learning algorithms. Construction and building materials, vol. 215, pp. 670–685 (2019) doi: 1016/j.conbuildmat.2019.04.227
6. Zhang, L., Shen, J., Zhu, B.: A research on an improved Unet-based concrete crack detection algorithm. Structural Health Monitoring, vol. 20 no. 4, pp. 1864–1879 (2021) doi:10.1177/1475921720940068
7. Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P.: A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced engineering informatics, vol. 29, no. 2, pp. 196–210 (2015) doi: 10.1016/j.aei.2015.01.008
8. Oliveira, H., Correia, P. L.: Automatic road crack detection and characterization. IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 1, pp. 155–168 (2013) doi: 10.1109/TITS.2012.2208630
9. Tang, J., Gu, Y.: Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 3026–3030 (2013) doi:10.1109/SMC.2013.516
10. Qingbo, Z.: Pavement crack detection algorithm based on image processing analysis. In: 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 15–18 (2016) doi:10.1109/IHMSC.2016.96
11. Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P.: A review on computer vision-based defect detection and condition assessment of concrete and asphalt civil infrastructure. Adv. Eng. Inform., vol. 29, no. 2, pp. 196–210 (2015) doi: 10.1016/ j.aei.2015.01.008
12. Cha, Y. J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378 (2017) doi: 10.1111/mice.12263

13. Leal da Silva, W. R., Schwerz de Lucena, D.: Concrete cracks detection based on deep learning image classification. In: 18th International Conference on Experimental Mechanics (ICEM18), Proceedings, vol. 2, no. 8, pp. 489 (2018) doi: 10.3390 /ICEM18-05387
14. Kim, B., Cho, S.: Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sensors, vol. 18, no. 10, pp. 3452 (2018) doi: 10.3390/ s18103452
15. Özgenel, Ç. F.: Concrete crack images for classification. Mendeley Data (2019) doi: 10.17632/5Y9WDSG2ZT.2
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893, (2005) doi: 10.1109/CVPR.2005.177
17. Tikkanen,T., Ruusuvuori, P., Latonen, L., Huttunen, H.: Training based cell detection from bright-field microscope images. In: 9th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 160–164 (2015) doi: 10.1109/ISPA.2015.7306051
18. Laopracha, N., Sunat, K.: Comparative study of computational time that hog-based features used for vehicle detection. In: Recent Advances in Information and Communication Technology 2017. IC2IT 2017. Advances in Intelligent Systems and Computing, vol 566, (2018) doi: 10.1007/978-3-319-60663-7_26
19. Carcagnì, P., del Coco, M., Leo, M., Distante, C.: Facial expression recognition and histograms of oriented gradients: a comprehensive study. SpringerPlus 4, pp. 645 (2015) doi: 10.1186/s40064-015-1427-3
20. Özgenel, Ç. F., Sorguç, A. G.: Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In: Proceedings of the 35th ISARC, pp. 693–700 (2018)

ISSN 1870-4069

Nuevo método para atribución de autoría en muestras de entrenamiento balanceadas y desbalanceadas del corpus C10

Cesar Alexis Estrada Palacios, José Luis Tapia Fabela

Universidad Autónoma del Estado de México,
México

{kirdrazler,joseluis.fabela}@gmail.com

Resumen. Atribución de Autoría es la tarea de identificar el autor de un texto anónimo. El uso de inteligencia artificial es muy común en esta tarea y uno de los principales problemas es que no se cuenta con suficientes textos de entrenamiento de uno o más candidatos autores, lo que genera un problema de desbalance, afectando en gran medida el rendimiento de los métodos propuestos en el estado del arte. La mayoría de estos métodos utilizan SVM como clasificador y una de las siguientes características como modelo de representación de texto: unigramas de palabra o trigramas de carácter. En este artículo se propone un método con un rendimiento superior a los métodos del estado del arte en muestras balanceadas y desbalanceadas, esto se logra mediante la combinación de distintos tamaños de n-gramas para formar la bolsa de n-gramas y generar un nuevo modelo de representación de texto. Los experimentos realizados muestran la importancia de entrenar al clasificador con una bolsa de n-gramas de diferentes tamaños y no solo con un tamaño fijo. Los mejores resultados se obtienen cuando se combinan unigramas y bigramas de palabra, además se muestra la importancia que tiene la estrategia de clasificación cuando se utiliza SVM.

Palabras clave: Atribución de autoría, representación de texto, estrategia de clasificación, SVM, bolsa de n-gramas.

New Method for Authorship Attribution in Balanced and Unbalanced Training Samples from the C10 Corpus

Abstract. Authorship Attribution is the task of identifying the author of an anonymous text. The use of artificial intelligence is very common in this task, and one of the main problems is that there are often not enough training texts from one or more candidate authors this generates an imbalance problem, greatly affecting the performance of the methods proposed in the state of the art. Most of these methods use SVM as a classifier and one of the following features as a

text representation model: word unigrams or character trigrams. This article proposes a method with superior performance to the methods of the state of the art in balanced and unbalanced samples, this is achieved by combining different sizes of n-grams to form the bag of n-grams and generate a new model. of text representation. The experiments carried out show the importance of training the classifier with a bag of n-grams of different sizes and not only with a fixed size. The best results are obtained when word unigrams and bigrams are combined, and the importance of the classification strategy when using SVM.

Keywords: Authorship attribution, text representation, classification strategy, SVM, bag of n-grams.

1. Introducción

La tarea de atribución de autoría consiste en identificar el autor de un texto del cual se desconoce su autoría a partir de un conjunto de posibles autores [1, 2]. Actualmente hay una gran cantidad de información alojada en la web en forma de texto, como e-mails, mensajes en foros, blogs, código fuente, entre otros [1, 3]; a partir de esta información surgen problemas como ciber bullying, plagio, spam, fraude, etc. En estos casos la atribución de autoría juega un papel importante [5]. Normalmente se utilizan métodos de inteligencia artificial para identificar el autor de dichos textos, estos métodos necesitan de textos de referencia para su entrenamiento, el problema es que a menudo se cuentan con pocos textos sobre un autor candidato, lo que genera un problema de desbalance, o en general se tiene pocos documentos de todos los autores lo cual afecta drásticamente el rendimiento de los métodos de aprendizaje supervisado [6].

Se han generado propuestas que funcionen en distintos escenarios de balance y desbalance, por ejemplo, en [1] se proponen varios métodos, donde se concluye que al utilizar 2500 características en promedio para la representación de texto se logran buenos resultados. En [2] se compara el rendimiento del clasificador SVM y STM, siendo superior SVM, teniendo buenos resultados con muestras balanceadas. Por otra parte [3] consigue buenos resultados en muestras desbalanceadas a través de una nueva representación de texto.

En ninguno de los casos anteriores hay un método que sea superior a los demás, ya que solo tienen un buen rendimiento en escenarios muy específicos, es por esto que el objetivo de este trabajo es desarrollar un método que tenga un rendimiento superior a los métodos del estado del arte tanto con muestras balanceadas como desbalanceadas.

Como primer características del método, se propone utilizar una bolsa de n-gramas conformada por distintos tamaños de n-gramas de palabra, con la finalidad de que el clasificador no solo aprenda de un tipo de característica, esto se basa en la observación de los autores [4] donde se utilizan los 100 n-gramas de palabra más frecuentes de tamaño 1, 2 y 3 (300 en total), obteniendo resultados competentes en comparación con los 2500 utilizados en los trabajos de [1–3]. Con base en los trabajos anteriormente mencionados se propone formar una bolsa de n-gramas a partir de los 2500 más frecuentes probando con distintos tamaños.

Como segunda característica del método y debido a los buenos resultados mostrados por el clasificador SVM en trabajos del estado del arte sobre autoría, se pone a prueba distintas variantes de SVM y a diferencia de los trabajos mencionados anteriormente, comparamos la estrategia de clasificación utilizada por el clasificador; ya que los resultados muestran que es uno de los factores que más influye en el proceso de clasificación.

Para evaluar esta idea, se presentan una serie de experimentos con muestras de entrenamiento balanceadas y desbalanceadas propuestas por [2] basadas en el corpus C10.

Los resultados muestran que utilizar una bolsa de n-gramas con diferentes tamaños de n-grama mejora los resultados obtenidos por el clasificador. El mejor resultado se obtuvo cuando se combinan palabras y bi-gramas de palabra. Además, la estrategia de clasificación uno contra todos obtiene los mejores resultados. Las distintas variantes de SVM juegan un papel mínimo en el resultado de la clasificación; sin embargo, los resultados muestran que SVM Dual es la mejor opción.

2. Corpus y método

2.1. Corpus

Para evaluar el método propuesto, se utiliza el conjunto de datos C10 [2], el cual es un subconjunto de Reuters Corpus Volumen 1 [5] contempla a los 10 autores con mayor cantidad de documentos relacionados al tema de noticias corporativas e industriales; de acuerdo con el autor, con la finalidad de que el tema no sea un factor para distinguir entre los autores. Cada autor cuenta con 50 documentos de *train* y 50 de *test*.

A partir del corpus C10, se experimenta en 6 posibles escenarios de entrenamiento, 3 escenarios balanceados donde se utilizan 50, 10 y 5 textos de entrenamiento por autor y 3 escenarios desbalanceados en donde el número de textos de entrenamiento por autor varia en un rango de 2:10, 5:10 y 10:20; considerando el caso 2:10 se pueden utilizar entre 2 a 10 documentos por autor.

Estos escenarios fueron propuestos anteriormente por [2] para evaluar el comportamiento de su método simulando distintos escenarios realistas donde puede existir un desbalance o se cuenta con muy pocos documentos por autor.

La Figura 1 y Figura 2 son ejemplos de una distribución de muestra balanceada y desbalanceada respectivamente.

2.2. Método

El método propuesto sigue las etapas que normalmente se siguen en la clasificación de textos, ya que atribución de autoría puede ser vista como una tarea de clasificación de textos [2].

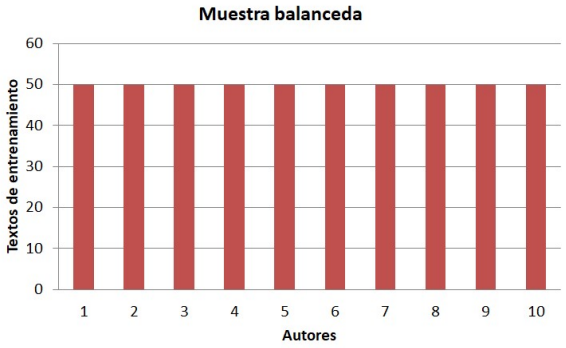


Fig. 1. Distribución de la muestra balanceada con 50 datos de entrenamiento.

Fig. 2. Distribución de la muestra desbalanceada para el caso 2:10.

En general, el proceso de clasificación de textos según [1] consiste en las siguientes etapas.

1. Adquisición de datos: El método propuesto necesita de un train y test, debido a que el aprendizaje utilizado es del tipo supervisado; porque utiliza SVM como algoritmo de clasificación. Con fines comparativos se utilizó el corpus C10, descrito en la sección 2.1.

2. Análisis y etiquetado de los datos: El corpus C10 originalmente está dividido en train y test, el etiquetado de datos dependerá del enfoque elegido, ya sea basado en instancias o basado en perfil [7], la elección del enfoque dependerá: del tipo de desbalance que se tenga, del clasificador a utilizar y del costo de entrenamiento, entre otros factores. El método propuesto utiliza el enfoque basado en instancias; porque es el recomendado cuando el desbalance es producido por la cantidad de textos de entrenamiento disponibles y cuando se utiliza un algoritmo de aprendizaje, en este caso SVM.

Fig. 3. Proceso general de un método de clasificación de textos basado en (Mirończuk & Protasiewicz, 2018).

3. Construcción y pesado de características: Para poder construir las características deseadas, en este caso palabras y bigramas de palabra, es necesario realizar primero un preprocesamiento; el cual consiste en dar un formato adecuado al texto, eliminando la información que no se necesita. Existen diversas formas de llevar a cabo el preprocesamiento, específicamente el método propuesto utiliza las siguientes: eliminación de saltos de línea, números y signos de puntuación; dejando solo los caracteres que son letras. A partir de este punto surge la posibilidad de utilizar stemming o no, en la etapa de experimentación se prueba su efectividad. De cada texto preprocesado, se obtienen los n-gramas de palabra, los cuales conforman la bolsa de n-gramas. Se proponen tres opciones a explorar, unigramas + bigramas, unigramas + trigramas y unigramas + bigramas + trigramas; lo anterior, con el fin de averiguar si al combinar n-gramas mejora los resultados y cuál es la mejor combinación. Como siguiente etapa se obtiene el pesado de características, también conocido como pesado de términos, como técnica se utilizó pesado binario, esto quiere decir que se asigna el valor de 1 si el término aparece en el texto y 0 si no aparece.

4. Selección de características: Consiste en seleccionar las características que utiliza el clasificador. De acuerdo con [1–3] en promedio 2500 características son necesarias para obtener buenos resultados; por lo tanto, para entrenar al clasificador se propone utilizar las 2500 características más frecuentes.

5. Entrenamiento del modelo de clasificación: El modelo utilizado en la mayoría de los trabajos de autoría es SVM; debido a sus buenos resultados [2, 3]. Existen diferentes variantes de SVM, uno de los más conocidos; además de utilizado por los autores

Tabla 1. Comparación de rendimiento entre diferentes tamaños de n-gramas de palabra.

Característica	**5**	**10**	**50**	**02:10**	**05:10**	**10:20**
Unigramas	**67.74**	**72.07**	79.60	**60.87**	**68.94**	**73.58**
Bigramas	62.41	69.34	**80.40**	56.56	65.87	71.38
Trigramas	49.15	56.53	68.40	47.78	52.97	60.02

Tabla 2. Comparación de la combinación de las bolsas n-gramas.

Característica	**5**	**10**	**50**	**02:10**	**05:10**	**10:20**
Unigramas+bigramas	**68.05**	73.58	**83.60**	60.89	**69.79**	**74.97**
Unigramas+trigramas	67.30	72.49	80.40	**61.48**	69.25	73.93
Bigramas+trigramas	60.49	69.27	77.20	56.20	63.57	70.78
Unigramas+bigramas +trigramas	67.73	**73.60**	82.20	61.00	69.48	74.56

Tabla 3. Aplicación de stemming en el modelo de n-gramas combinado.

Característica	**5**	**10**	**50**	**02:10**	**05:10**	**10:20**
Unigramas+bigramas	**68.05**	**73.58**	**83.60**	**60.89**	**69.79**	**74.97**
Unigramas+bigramas+stemming	67.63	72.67	81.60	60.72	69.05	74.16

Tabla 4. Comparación del rendimiento de las variantes de SVM.

Clasificador	**5**	**10**	**50**	**02:10**	**05:10**	**10:20**
SVM Linear	66.89	73.33	83.40	60.06	68.98	74.60
SVM Primal	**68.38**	73.00	82.20	60.79	69.67	74.49
SVM Dual	68.05	**73.58**	**83.60**	**60.89**	**69.79**	**74.97**

mencionados anteriormente es SVM Lineal. Se experimentó con la versión LibSVM Linear; SVM L2 Primal y L2 SVM Dual logrando los mejores resultados con SVM Dual. Cuando se presentan problemas multiclase SVM necesita de una estrategia de clasificación. En nuestro caso de estudio, el corpus utilizado consta de 10 autores, por lo tanto, de 10 clases; por esto es necesario utilizar una estrategia de clasificación, las más comunes son la estrategia uno contra uno y uno contra todos.

6. Con fines comparativos se utilizó la métrica *accuracy*, debido a que es la métrica utilizada en el estado del arte por diversos autores para evaluar el corpus C10 [4, 8, 9].

3. Experimentos

En esta sección se evalúa el rendimiento del método propuesto, se presenta detalle en que consistió cada experimento, además de analizar los resultados obtenidos. En los siguientes experimentos se ponen a prueba diferentes tamaños de n-gramas, combinación de n-gramas, stemming, variantes de SVM y estrategias de clasificación. Cada experimento se evalúa en 6 escenarios diferentes, 3 con muestras de entrenamiento balanceadas y 3 desbalanceadas, estas muestras se detallan en la sección de corpus.

Con fines estadísticos y de réplica, cada resultado mostrado en cada experimento es el promedio de 100 corridas y no solo 10 corridas como se realizó en los experimentos de [3] permitiéndonos mostrar resultados más confiables y precisos.

3.1. Modelo de n-gramas simple

En este experimento se compara el rendimiento del método propuesto cuando se utiliza n-gramas de palabra como modelo de representación de texto. En la tabla 1 se compara el rendimiento entre unigramas, bigramas y trigramas de palabra.

El mejor resultado se obtiene al utilizar unigramas de palabras, que son básicamente palabras. Se puede observar que entre más grande es el n-grama peores resultados se obtienen tanto en muestras balanceadas como desbalanceadas, por lo que no es necesario experimentar con un tamaño de n-grama mayor.

3.2. Modelo bolsa de n-gramas compuesta

En este experimento se crea una bolsa de n-gramas a partir de n-gramas de palabra de diferentes tamaños, en la tabla 2 se muestran las diferentes combinaciones posibles entre unigramas, bigramas y trigramas.

Como se muestra en la tabla 2, los mejores resultados se obtienen a partir de la combinación de unigramas y bigramas de palabra. Además, ocurre un comportamiento semejante al experimento anterior, entre más pequeños sean los n-gramas a combinar mejores resultados se obtienen. Se observa también que la estrategia de clasificación influye en gran medida en el resultado obtenido.

Los resultados muestran que la representación de texto del método propuesto mejora los resultados en todas las pruebas realizadas, especialmente en la muestra 50, donde se obtiene casi 4% de diferencia respecto a las demás formas de representación de texto.

3.3. Stemming

Con el fin de saber si aplicar stemming mejora los resultados, en este experimento se evalúa el efecto de aplicar stemming al modelo de bolsa de n-gramas combinado unigramas + bigramas de palabra ya que fue la combinación que obtuvo mejores resultados en el experimento anterior. Los resultados muestran que utilizar stemming tiene un efecto negativo con todas las muestras de entrenamiento evaluadas y en ningún caso se logró una mejora. También se observa que entre más documentos de

entrenamiento se tengan para el entrenamiento peores resultados se obtendrán si se aplica stemming, es por ello que existe una mayor diferencia en los resultados de la muestra balanceada 50.

3.4. Clasificador

El clasificador SVM ha sido utilizado en cada uno de los trabajos con los cuales comparamos el método propuesto [2, 3, 10]; sin embargo, ninguno de esos trabajos ha evaluado las variantes de este clasificador, por lo tanto, los siguientes dos experimentos se realizaron con la intención de conocer cual variante de SVM y estrategia de clasificación logra obtener los mejores resultados.

Variantes de SVM. En este experimento, se compara el rendimiento de las variantes de SVM (Linear, l2 Primal y l2 Dual) para conocer cual tiene un mejor rendimiento en la clasificación. Los parámetros utilizados fueron c= 1 mismo que fue utilizado por [2] y el valor de épsilon = 0.001 estos dos parámetros aplican para las 3 variantes de SVM ya mencionadas. Como modelo de representación texto, se utiliza el modelo combinado de unigramas + bigramas de palabra sin aplicar stemming.

La variante del clasificador con mejor rendimiento fue SVM Dual, logrando los mejores resultados en 5 de 6 muestras; solo fue superado en la muestra balanceada con 5 documento por una diferencia mínima. SVM Primal obtuvo un rendimiento inferior en la mayoría de las pruebas en comparación de SVM Dual; sin embargo, fue superior a SVM Linear en la mayoría de las pruebas, con excepción de la muestra 50. SVM Linear no logró obtener el mejor resultado en ninguna de las muestras.

Estrategia de clasificación. Usando el resultado del experimento anterior, donde se comprobó que usando SVM Dual se obtiene los mejores resultados en la clasificación, se evalúa ahora la estrategia de clasificación: uno contra uno OvO y uno contra todos OvA, para saber cuál estrategia ayuda más al clasificador.

La estrategia de clasificación uno contra todos es superior en todos los casos a la estrategia uno contra uno, en todas las muestras se tiene una mejora considerable, en especial en las muestras donde existe un mayor desbalance y se cuenta con el menor número de documentos. Por lo anterior podemos concluir que la estrategia de clasificación es el factor que más influye en el clasificador cuando existe desbalance.

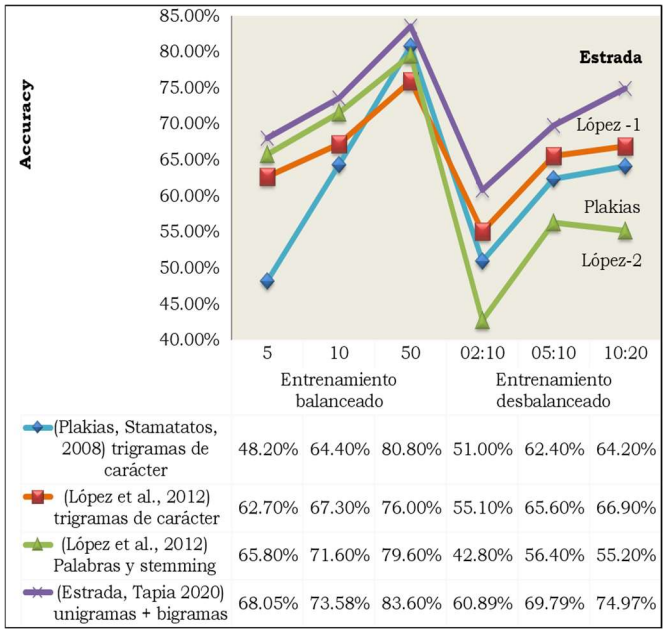
4. Discusión

Con base en los resultados obtenidos en la etapa de experimentación, se observa que la estrategia de clasificación es el factor que más influye en el resultado cuando hay pocos documentos disponibles y existe un desbalance. Combinar n-gramas mejora ligeramente los resultados en comparación con no combinar n-gramas. Entre más documentos se tengan mayor ventaja tiene el combinar n-gramas.

A pesar de que SVM Dual es superior en todas las pruebas a SVM Linear y SVM Primal, la diferencia es mínima, por lo que la variante del clasificador no tiene un gran impacto sobre el resultado obtenido. En cuanto al stemming, este produce un impacto negativo en los resultados obtenidos mediante el método propuesto.

Tabla 5. Comparación de las estrategias de clasificación uno contra uno y uno contra todos.

Característica	**Tamaño**	**OvO**	**OvA**	**Diferencia**
Unigramas + bigramas	5	65.47	*68.05*	2.58
	10	71.16	*73.58*	2.42
	50	79.60	*83.60*	4.00
	2:10	52.06	*60.89*	8.83
	5:10	64.53	*69.79*	5.26
	10:20	71.53	*74.97*	3.44
Unigramas + trigramas	5	64.11	*67.30*	3.19
	10	71.32	*72.49*	1.17
	50	77.00	*80.40*	3.40
	2:10	52.33	*61.48*	9.15
	5:10	65.05	*69.25*	4.20
	10:20	71.84	*73.93*	2.09
Bigramas + trigramas	5	57.18	*60.49*	3.31
	10	66.20	*69.27*	3.07
	50	74.60	*77.20*	2.60
	2:10	49.58	*56.20*	6.62
	5:10	58.56	*63.57*	5.01
	10:20	66.92	*70.78*	3.86
Unigramas + bigramas + trigramas	5	64.59	*67.73*	3.14
	10	70.90	*73.60*	2.70
	50	78.40	*82.20*	3.80
	2:10	52.25	*61.00*	8.75
	5:10	64.34	*69.48*	5.14
	10:20	71.75	*74.56*	2.81



	5	10	50	02:10	05:10	10:20
(Plakias, Stamatatos, 2008) trigramas de carácter	48.20%	64.40%	80.80%	51.00%	62.40%	64.20%
(López et al., 2012) trigramas de carácter	62.70%	67.30%	76.00%	55.10%	65.60%	66.90%
(López et al., 2012) Palabras y stemming	65.80%	71.60%	79.60%	42.80%	56.40%	55.20%
(Estrada, Tapia 2020) unigramas + bigramas	68.05%	73.58%	83.60%	60.89%	69.79%	74.97%

Fig. 4. Comparación del método propuesto por Estrada con el estado del arte.

En la figura 4 se compara el mejor resultado obtenido a partir del método propuesto (Estrada) respecto al estado del arte en las muestras balanceadas y desbalanceadas del corpus C10.

Los resultados muestran que el método propuesto, supera a los métodos presentados en el estado del arte en todas las muestras, balanceadas y desbalanceadas, principalmente existe una mayor diferencia cuando se entrena con muestras desbalanceadas, siendo estas las que mayor se presentan en un escenario realista, además el método propuesto es robusto, debido a que mantiene un mejor rendimiento cuando se tiene menos documentos de entrenamiento o existe un desbalance.

5. Conclusiones

Se puede hacer las siguientes conclusiones:

1. Combinar distintos tamaños de n-gramas de palabras como modelo de representación de texto beneficia al clasificador logrando una mejor accuracy tanto en muestras balanceadas como desbalanceadas.
2. Utilizar la combinación unigramas+bigramas de palabras como modelo de representación de texto produce los mejores resultados. Estos resultados son superiores a los expuestos por [2, 3] en los 6 diferentes escenarios.
3. El factor que más influye en el accuracy del método es la estrategia de clasificación.

4. En términos de accuracy la estrategia de clasificación uno contra todos es muy superior a la estrategia uno contra uno, pues se obtienen resultados superiores con todos los modelos de representación de texto evaluados en la etapa de experimentación, con excepción de trigramas de carácter en la muestra de 50.
5. Al igual que el estado del arte, el método propuesto disminuye su rendimiento cuando se tienen menos documentos o es mayor el desbalance.
6. Usar palabras es más robusto que utilizar n-gramas de carácter.
7. Utilizar stemming empeora los resultados con el modelo de representación de texto propuesto.

Referencias

1. Stamatatos, E.: Author identification using imbalanced and limited training texts. In: 18th International Conference on Database and Expert Systems Applications (DEXA 2007). IEEE, Regensburg, Germany, pp. 237–241 (2007) doi: 10.1109/DEXA.2007.5
2. Plakias, S., Stamatatos, E.: Tensor space models for authorship identification. Darzentas, J., Vouros, G. A., Vosinakis, S., and Arnellos, A. (eds.). Artificial intelligence: theories, models and applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 239–249 (2008) doi: 10.1007/978-3-540-87881-0_22
3. López-Monroy, A. P., Montes-y-Gómez, M., Villaseñor-Pineda, L., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F.: A New Document author representation for authorship attribution. Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., Olvera López, J. A., Boyer, K. L. (eds.) Pattern Recognition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 283–292 (2012) doi: 10.1007/978-3-642-31149-9_29
4. Sari, Y., Stevenson, M., Vlachos, A.: Topic or style? exploring the most useful features for authorship attribution. In: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA pp. 343–353 (2018)
5. Lewis, D. D., Yang, Y., Rose, T. G., Li, F.: RCV1: A new benchmark collection for text categorization research. 37 (2004)
6. Mirończuk, M. M., Protasiewicz, J.: A recent overview of the state-of-the-art elements of text classification. Expert Syst. Appl., vol. 106, pp. 36–54 (2018) doi: 10.1016/j.eswa. 2018.03.058
7. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol., vol. 60, pp. 538–556 (2009) doi: 10.1002/asi.21001
8. Popescu, M., Grozea, C.: Kernel methods and string Kernels for authorship analysis. CLEF (2012)
9. Potthast, M., Braun, S., Buz, T., Duffhauss, F., Friedrich, F., Gülzow, J. M., Köhler, J., Lötzsch, W., Müller, F., Müller, M. E., Paßmann, et al: Who wrote the web? Revisiting influential author identification research applicable to information retrieval. Ferro, N., Crestani, F., Moens, M.F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., Silvello, G.

(eds.) Advances in Information Retrieval, pp. 393–407, Springer International Publishing, Cham (2016) doi: 10.1007/978-3-319-30671-1_29

10. Escalante, H. J., Solorio, T., Montes y Gomez, M.: Local histograms of character n-grams for authorship attribution. 11 (2011)

ISSN 1870-4069

Modelado de estrategias de regulación emocional en una arquitectura computacional de emociones

Juan José Solórzano[1], Juan Martínez-Miranda[2], Rosa María Michel Nava[1]

[1] Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Guzmán, México

[2] Centro de Investigación Científica, México

juanjose.solorzano.c@gmail.com,
rosa.mn@cdguzman.tecnm.mx,
jmiranda@cicese.mx

Resumen. El modelado de procesos afectivos en sistemas artificiales contribuye a generar comportamientos más creíbles en sistemas interactivos como los agentes conversacionales personificados. El presente artículo describe un modelo computacional de regulación de emociones basado en la teoría de J.J. Gross la cual propone cinco estrategias de regulación emocional. Además, el modelo propuesto considera las diferencias individuales en la implementación de cada estrategia de regulación a partir de los cinco tipos de personalidad tomados de los cinco grandes rasgos (Big-Five). La implementación de este modelo y su integración con una arquitectura computacional de emociones existente se describe en detalle y se presentan los resultados iniciales a partir de un conjunto de simulaciones representando un escenario interactivo con agentes conversacionales personificados.

Palabras clave: Regulación emocional, modelado de emociones, agentes virtuales conversacionales.

Modeling Emotion Regulation Strategies in a Computational Architecture of Emotions

Abstract. The modeling of affective processes in artificial systems contributes to generate more believable behaviors in interactive systems such as embodied conversational agents. This article describes a computational model of emotion regulation based on the theory proposed by J.J. Gross which considers five emotional regulation strategies. In addition, the proposed model considers the individual differences in the implementation of each emotion regulation strategy based on the personality types taken from the Big-Five personality model. The implementation of this model and its integration with an existing computational architecture of emotions are described in detail. The initial results obtained from a set of simulations representing an interactive scenario with embodied conversational agents are also presented.

Keywords: Emotion regulation, emotion modeling, embodied conversational agents.

1. Introducción

En el campo de la inteligencia artificial un concepto ampliamente utilizado es el de agentes artificiales inteligentes. Russell y Norvig [18], definen a un agente artificial como cualquier ente capaz de percibir su entorno y actuar sobre él con la ayuda de sensores y actuadores. Los agentes artificiales tienen características específicas que conducen a su aplicación en diferentes dominios como la interacción humano-computadora para desarrollar sistemas interactivos con mejores capacidades.

Una variante interesante de los agentes artificiales es aquella que se presentan ante el usuario con una apariencia humana. Este tipo de agentes se conocen como agentes conversacionales personificados (en inglés: embodied conversational agents).

Este tipo de agentes implementan capacidades de interacción basadas no únicamente en lenguaje natural también en expresiones faciales y movimientos corporales, lo que resulta en interacciones más realistas debido al uso de la comunicación verbal y no verbal.

Gracias a estas características los agentes conversacionales personificados se han utilizado en aplicaciones de diversos campos, como en la educación [13], la salud [14], y el turismo [11]. El componente que genera el comportamiento cognitivo y emocional en este tipo de agentes es la arquitectura computacional de emociones subyacente, cuyo fin es generar reacciones emocionales coherentes que permitan lograr comportamientos más realistas y creíbles para los usuarios.

En los últimos años, se han desarrollado diversas arquitecturas computacionales de emociones basadas en diferentes teorías de generación emocional [12]. Sin embargo, uno de los procesos emocionales que muy pocos trabajos han tomado en cuenta en el desarrollo de este tipo de arquitecturas, es el proceso de regulación emocional considerando además las diferencias individuales que influyen en este proceso.

Generar comportamientos emocionales empáticos y coherentes de acuerdo al contexto de aplicación es importante en agentes conversacionales personificados utilizados como medio de interacción con usuarios con características especiales. Un ejemplo de esto son los agentes conversacionales desarrollados con objetivos psicoterapéuticos, en los cuales no es conveniente mostrar emociones negativas intensas durante la interacción con el usuario.

Con el objetivo de facilitar la creación de agentes conversacionales personificados con comportamientos emocionales más coherentes de acuerdo al contexto de aplicación, este trabajo presenta el desarrollo de una arquitectura computacional de regulación de emociones basada en el modelo de regulación emocional propuesto por J.J. Gross [7] y tomando en cuenta las diferencias individuales basadas en el modelo de los cinco grandes factores de personalidad (Big-Five) [15].

El objetivo final del desarrollo de esta arquitectura es poder utilizarla como el mecanismo interno para la generación del comportamiento emocional en agentes conversacionales personificados utilizados en aplicaciones interactivas de ayuda a la prevención y tratamiento de problemas de salud mental como depresión o ansiedad.

1.1. Fundamentos teóricos de la regulación de emociones

La regulación de emociones es un proceso afectivo que hace referencia a las acciones que un individuo puede llevar a cabo para influir sobre qué tipo de emociones experimentar, cuándo y cómo experimentarlas, y en qué grado expresarlas [7]. Una emoción, por otro lado, es un estado mental complejo que involucra tres aspectos distintos: la experiencia subjetiva, la respuesta mental a esa experiencia y la respuesta conductual o fisiológica [1]. Actualmente, no existe un consenso sobre si los procesos de generación y regulación emocional pueden y deben distinguirse entre sí o, por el contrario, pueden usarse indistintamente.

Gross & Feldman Barret realizaron un estudio en el que, a través de una serie de preguntas, agruparon cuatro enfoques principales de la psicología emocional para destacar y determinar si se puede diferenciar entre estos dos procesos [8].

El resultado de este estudio permite visualizar una clara distinción entre el proceso de generación y regulación de emociones bajo las perspectivas de las teorías de emociones básicas [6] y de la valoración cognitiva [16].

Sin embargo, para los enfoques construccionistas [1] y socio-construccionistas [20] esta separación no es tan evidente. Basándose en la perspectiva de la valoración cognitiva, J. J. Gross estableció su Modelo Modal de la emoción y describió los conceptos básicos de la regulación emocional [7].

Este Modelo Modal postula un conjunto de cinco estrategias que una persona puede llevar a cabo para regular sus emociones en su vida diaria:

1. **Selección de la situación**: Tomar acciones que vuelven probable o no terminar en una situación esperada, dando lugar a emociones deseadas o no deseadas,
2. **Modificación de la situación**: Modificar directamente la situación que se está viviendo con la finalidad de alterar su impacto emocional,
3. **Enfoque de la atención**: Modificar el grado de atención a un determinado evento con la finalidad de influir en las emociones experimentadas,
4. **Cambio cognitivo**: Cambiar la forma en que se valora un evento con el propósito de alterar su significado emocional, cambiando la perspectiva que se tiene sobre la situación,
5. **Modulación de la respuesta**: Esta estrategia se implementa una vez que se produce la emoción y es la acción que se toma para influir en la respuesta fisiológica, experiencial o conductual producida por la emoción que ya se ha generado.

Las personas tienden a regular sus emociones de diferentes maneras. Dependiendo de su tipo de personalidad, los individuos optarán por implementar, en mayor o menor grado, una de las cinco estrategias de regulación emocional.

En este sentido, Gross y Oliver llevaron a cabo un estudio para determinar la correlación entre los tipos de personalidad y las diferentes estrategias de regulación emocional [10]. Estos estudios son los que se han tomado de referencia en este trabajo para la implementación del modelo de regulación emocional tomando en cuenta las diferencias individuales.

2. Trabajo relacionado

El grupo académico de Jan Treur y colegas ha desarrollado el lenguaje formal LEADSTO para modelar matemáticamente las relaciones cuantitativas y cualitativas de sistemas complejos a través de la combinación de métodos de modelado lógico y matemático [3]. Este lenguaje se ha utilizado en diferentes trabajos para representar computacionalmente el proceso de regulación emocional [2, 21].

Por ejemplo, CoMERG [4] es un modelo computacional basado en LEADSTO, en el cual se modelan cuatro estrategias de regulación emocional mediante un conjunto de variables para representar aspectos cuantitativos (como los niveles de respuesta emocional) y cualitativos (como las posibles decisiones para regular una emoción). Este conjunto de variables incluye: el nivel de intensidad de la emoción actual, el nivel óptimo esperado de una emoción, y la tendencia personal a modificar el valor de la intensidad emocional.

En [23] se presenta un modelo computacional inspirado en teorías neurológicas que utiliza un enfoque de modelado orientado a redes (Network-Oriented Modelling, ver [22]) para representar la variación que existe en la elección de diferentes estrategias de regulación emocional en función de la edad y el género.

Este enfoque se deriva del lenguaje LEADSTO y se utilizan funciones matemáticas para representar conceptualmente nodos y enlaces, donde cada nodo corresponde a una variable y cada enlace a su relación. Por ejemplo, un nodo corresponde a un estado emocional de un agente producido por un estímulo desencadenado por otro nodo.

Un peso asociado a este par de nodos es la variable que representa la edad y el género en un agente. Dependiendo de los valores preestablecidos en cada nodo, la elección entre las estrategias modeladas (cambio cognitivo y modulación de la respuesta) dependerá de los valores asociados a los pesos de cada enlace.

El trabajo descrito en [9] presenta un modelo computacional que utiliza cadenas de Markov para representar probabilísticamente el modelo de regulación emocional de Gross. En este trabajo se implementa el modelo Big-Five [15] para relacionar los rasgos de personalidad y el proceso de regulación emocional. La relación entre los diferentes estados emocionales y los rasgos de personalidad se establece mediante una matriz de transición de estados.

Cada rasgo de personalidad ocupa una posición dentro de un vector unidimensional utilizado para calcular el factor de efectividad que determina la implementación de una estrategia. Para representar el efecto que tiene una estrategia en función a los eventos que suceden en el entorno y al tipo de personalidad, se utiliza también una matriz de transición de estados. De esta forma se representan las estrategias de cambio cognitivo y modulación de la respuesta.

A diferencia de estos trabajos relacionados, la principal contribución de nuestro modelo es la representación de las cinco estrategias de regulación emocional propuestas por Gross [7], vinculando de manera explícita los cinco rasgos de personalidad del modelo Big-Five [15] con cada una de las estrategias de regulación emocional. Además, el modelo se ha implementado como un módulo adicional en una arquitectura de generación de emociones, facilitando la implementación de este proceso afectivo en agentes conversacionales personificados para diferentes escenarios de interacción.

3. Metodología

El modelo de regulación emocional propuesto se integra específicamente a partir de tres componentes: generación de emociones, representación de los rasgos de personalidad, y la representación de las cinco estrategias de regulación emocional. A continuación se describe cada uno de estos componentes.

Generación de emociones. Actualmente existen diferentes arquitecturas computacionales de generación de emociones pero que no implementan de manera explícita el proceso de regulación emocional. Una de estas arquitecturas es FAtiMA [5].

La arquitectura FAtiMA proporciona un conjunto de herramientas que funcionan como librerías independientes para la creación de agentes virtuales con comportamientos emocionales apropiados.

El mecanismo de inferencia para la generación de emociones en FAtiMA se basa en el modelo OCC [16] de valoración cognitiva, lo cual hace necesario establecer un conjunto de variables de valoración asociados a los eventos que se producen en el entorno de un agente.

Dependiendo de los deseos, creencias y preferencias del agente, estas variables de valoración se utilizan para generar una emoción específica en el agente artificial a partir de los eventos detectados en su entorno.

Rasgos de personalidad. El concepto rasgos de personalidad denota patrones consistentes de comportamiento, pensamientos y emociones. El modelo Big-Five [15] es, por mucho, el modelo más aceptado, y que integra los resultados de cientos de estudios llevados a cabo durante décadas. Este modelo divide las características de personalidad de un individuo en cinco grandes rasgos: Apertura a la experiencia, Escrupulosidad, Extraversión, Amabilidad, y Neuroticismo.

Relación entre los tipos de personalidad y las estrategias de regulación emocional. Con el objetivo de tomar en cuenta las diferencias individuales en la selección de las estrategias de regulación, cada uno de los cinco tipos de personalidad se relaciona con el uso específico de cada estrategia de regulación emocional. Para definir esta asociación, se tomaron como base los hallazgos presentados en [10, 17].

En estos estudios se correlaciona el uso habitual de las estrategias de regulación emocional en función de los rasgos de personalidad del modelo Big-Five. A partir de los resultados de estos estudios, se creó una matriz de correlación para asociar las cinco estrategias de regulación emocional con los cinco rasgos de personalidad (ver tabla 1).

Tabla 1. Correlación, tipos de personalidad y estrategias de regulación emocional.

Personalidad	**Selección de la situación**	**Modificación de la situación**	**Enfoque de la atención**	**Cambio cognitivo**	**Modulación de la respuesta**
Escrupulosidad	+	+	+	+	-
Extraversión	-	+	+	+	-
Neuroticismo	+	-	-	-	-
Apertura a la experiencia	-	+	+	+	-
Amabilidad	-	-	+	+	-

En la tabla 1, el signo "+" corresponde una correlación positiva entre el uso de la estrategia de regulación y el tipo de personalidad. Por el contrario, un signo "-" indica una correlación negativa. Se decidió utilizar lógica difusa para representar los niveles de personalidad: bajo, medio o alto y se definieron tres conjuntos difusos con dos funciones de pertenencia polinómicas Z y una función de pertenencia Gaussiana (ver figura 1).

De manera similar se definieron tres conjuntos difusos para representar la relación entre los tipos de personalidad y las estrategias de regulación emocional: baja relación, media relación, y alta relación.

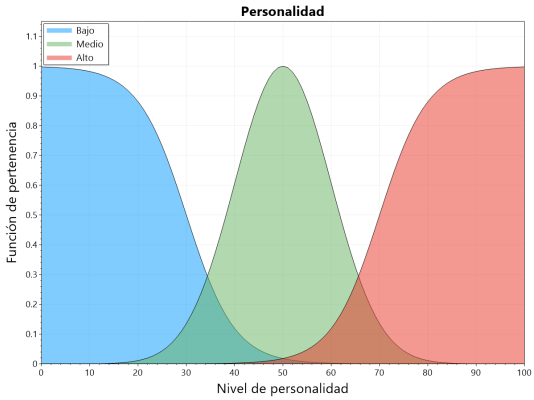


Fig. 1. Funciones de membresía para definir los tipos de personalidad.

Utilizando estos valores difusos, se implementaron las reglas de inferencia de tipo *"SI-ENTONCES"* (IF-THEN), y se comparan mediante del operador "Y" (AND) para definir qué tipo de estrategia es la que se aplicará en función del grado de pertenencia a cada uno de los cinco tipos de personalidad. En general, las reglas para la implementación de cada estrategia de regulación emocional se expresan como:

SI *Escrupulosidad* **ES** *alta* **Y** *Neuroticismo* **ES** *alto* **ENTONCES**: *Selección de la Situación* **ES** *Fuertemente aplicada*
SI *Escrupulosidad* **ES** *media* **Y** *Neuroticismo* **ES** *medio* **ENTONCES**: *Selección de la Situación* **ES** *Ligeramente aplicada*
SI *Escrupulosidad* **ES** *bajo* **Y** *Neuroticismo* **ES** *bajo* **ENTONCES**: *Selección de la Situación* **ES** *Débilmente aplicada*
...

3.1. Implementación de las estrategias de regulación emocional

En la Figura 2-A se presentan los elementos que constituyen el modelo de regulación emocional propuesto (resaltados en amarillo) y su interacción con el resto de

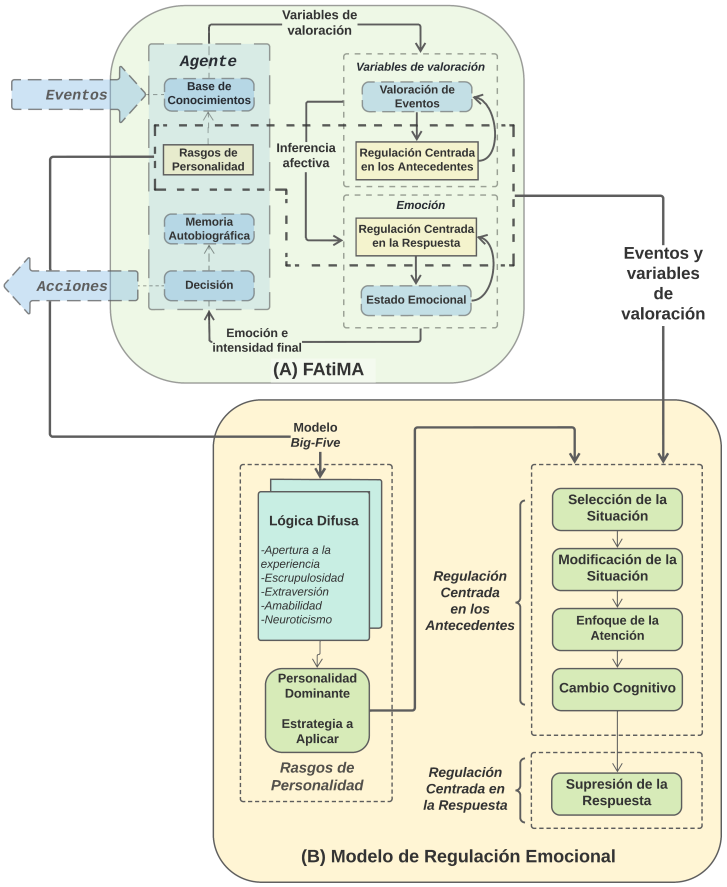


Fig. 2. Esquema de la arquitectura de regulación emocional. A) Integración del modelo propuesto en FAtiMA. B) Componentes del modelo de regulación emocional.

componentes de la arquitectura FAtiMA. En la Figura 2-B, se detallan las funcionalidades específicas de cada componente del modelo propuesto.

Por ejemplo, el componente rasgos de personalidad determina el tipo de personalidad dominante en un agente y las estrategias de regulación emocional que se pueden implementar. Las estrategias centradas en los antecedentes y la estrategia centrada en la respuesta, forman un mismo módulo en donde se lleva a cabo la implementación de cada estrategia de regulación emocional.

Cada característica relevante de las estrategias de regulación emocional se analizó para representarla de manera adecuada en el modelo computacional propuesto. Como resultado de este análisis se obtuvo el diagrama de flujo de la Figura 3 en el cual se visualiza la manera en que se implementa el proceso de regulación a partir de la detección de un evento. En términos generales, cada estrategia puede ser aplicada por

un agente al cumplir con los siguientes requisitos:

– Tipo de personalidad adecuado: La personalidad del agente debe de tener una correlación positiva con la estrategia a implementar (ver tabla 1),
– Información específica para cada estrategia: De acuerdo a la teoría de Gross, la implementación de cada estrategia requiere información específica. Por ejemplo, para la estrategia de selección de situación, se debe indicar si el agente tiene la capacidad de evitar el evento en curso,
– Valor límite de intensidad emocional: Una vez modificados los datos necesarios para que el evento evaluado produzca una intensidad emocional adecuada, se verifica que el valor obtenido en la intensidad de la emoción al aplicar la estrategia esté en el rango adecuado pre-definido por el usuario.

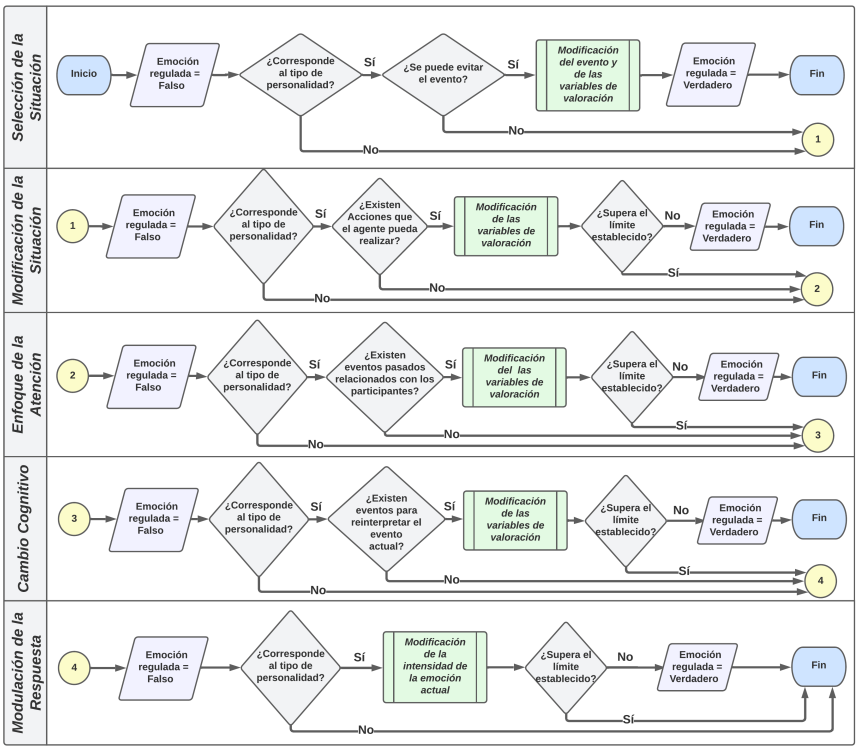


Fig. 3. Diagrama de flujo para las estrategias de regulación emocional.

La implementación de cada estrategia de regulación emocional se describe con más detalle a continuación.

Selección de la situación. De acuerdo con Gross, esta estrategia corresponde a la capacidad de un individuo para evitar que una determinada situación o evento ocurra en

su entorno, sobre todo aquellos eventos que puedan generar una emoción no deseada. En el modelo propuesto, los eventos que ocurren en el entorno del agente tienen la propiedad de poder ser evitados o no de acuerdo a la configuración pre-determinada en el escenario de interacción diseñado por el usuario.

Si se determina que un cierto evento es evitable, dependiendo de la valoración que haga el agente de dicho evento y su tipo de personalidad dominante, esta estrategia podrá aplicarse exitosamente, evitando la generación de una emoción y/o intensidad no deseada en el agente. En caso contrario, se continúa con la implementación de las siguientes estrategias.

Modificación de la situación. Un requisito para implementar esta estrategia es que el agente tenga la posibilidad de realizar al menos una acción para modificar el evento en curso. El conjunto de acciones para modificar el evento actual son definidas por el usuario y están vinculadas internamente a un evento específico.

Si el agente es capaz de realizar una acción que modifique el evento actual y en consecuencia cambiar la valoración que se hace sobre este evento, la emoción y/o intensidad resultante podrían estar dentro de los límites establecidos por el usuario para cada emoción que se desea regular.

El cálculo para determinar el nuevo valor de valoración del evento modificado se realiza mediante la función (1) que relaciona el tipo de personalidad con el valor original de la variable de valoración, es decir, cuanto más dominante sea el tipo de personalidad asociado con esta estrategia de regulación, mayor será la reducción sobre el valor original de la variable de valoración. De esta forma, el éxito en la aplicación de esta estrategia se expresa en función del tipo de personalidad presente en el agente. La función se deriva de la tangente hiperbólica:

$$f_1(x) = \left|\frac{\text{AppVariable}}{2}\right| \cdot \tanh\left(\frac{2x - \text{MaxVal}}{\frac{\text{MaxVal}}{2}}\right) + \left|\frac{\text{AppVariable}}{2}\right|. \qquad (1)$$

En la función 1, el parámetro AppVariable corresponde al valor original asignado a la variable de valoración del evento. El parámetro MaxVal es el valor máximo que puede tomar el dominio de la función. Y, la variable x corresponde al valor promedio de los tipos de personalidad. Si el nuevo valor de valoración es aún mayor al límite establecido para la intensidad de la emoción resultante, se continúa con la implementación de la siguiente estrategia.

Enfoque de la atención. Gross señala que esta estrategia se refiere a dirigir la atención de una manera que pueda alterar la respuesta emocional derivada de un evento. En nuestro modelo, esta estrategia se aplica siempre que exista en la memoria del agente un evento previo valorado como positivo que involucre a los participantes en el evento actual representando que el agente enfoca parte de su atención en eventos positivos anteriores.

Si hay un evento previo relacionado con un participante de la escena, el nuevo valor de la variable de valoración se calculará utilizando la función (1). Si el nuevo valor obtenido es menor que el límite establecido por el usuario, la regulación de la emoción es exitosa, de lo contrario, se continúa con la siguiente estrategia.

Cambio cognitivo. Esta estrategia se refiere a cambiar el significado de un evento para que el individuo pueda influir en las emociones que se generan a partir del evento detectado. En el modelo propuesto, esto se representa a través de generar diferentes perspectivas con una valoración positiva del evento actual. Esto se realiza creando nuevos eventos y asociándolos con el evento actual.

Para que las emociones sean reguladas con éxito por esta estrategia, las variables de valoración asignadas a los eventos utilizados para la re-interpretación del evento actual deben de ser positivas, el tipo de personalidad del agente debe corresponder al indicado por la tabla 1, y el nuevo valor en la variable de valoración calculado mediante la función (1) deberá ser menor al valor límite establecido. De lo contrario, se continuará con la siguiente estrategia.

Modulación de la respuesta. Esta estrategia se aplica una vez que la emoción ha sido generada, por lo que su implementación modifica directamente la intensidad de la emoción ya generada en lugar de la variable de valoración asociada al evento en curso. El nuevo valor de intensidad emocional dependerá del tipo de personalidad presente en el agente, por lo que se utiliza la función 1 para calcular esta nueva intensidad.

Si la nueva intensidad no supera el límite establecido, la emoción se regulará con éxito, en caso contrario, al ser la última estrategia, se indica que la emoción del agente no ha podido ser regulada. Una vez que se obtiene la emoción generada y regulada (en caso de éxito de las estrategias de regulación), el control se regresa a los módulos de la arquitectura FAtiMA para la generación del comportamiento correspondiente en el agente para hacer frente al evento actual.

4. Resultados

En esta sección se presentan y compararon los primeros resultados obtenidos de realizar diferentes pruebas de simulación. El modelo de regulación emocional se integró en la arquitectura FAtiMA [5], desarrollando diferentes eventos en un mismo escenario.

En cada simulación se configuraron diferentes valores en el tipo de personalidad de un mismo agente. Se consideraron únicamente las emociones negativas como objetivo a ser reguladas. El escenario consta de un total de 11 eventos que representan los acontecimientos que suceden en el entorno de un agente en el transcurso de un día:

"Sam es un oficinista que trabaja para una empresa desde hace algunos años. A lo largo del día, Sam experimenta diferentes emociones a causa de los eventos que suceden a su alrededor. Un día normal para Sam inicia al llegar a su trabajo (Evento 1), y finaliza al llegar a su hogar (Evento 11):

- **Evento 1**: Sam es notificado para hablar con su jefe ("Talk-To-Boss"),
- **Evento 2**: Sam saluda a María ("Hello"),
- **Evento 3**: Sam conversa con María ("Conversation"),
- **Evento 4**: María abraza a Sam ("Hug"),
- **Evento 5**: Sam discute con compañeros de la oficina ("Discussion"),
- **Evento 6**: Sam es felicitado por su amigo ("Congrat"),

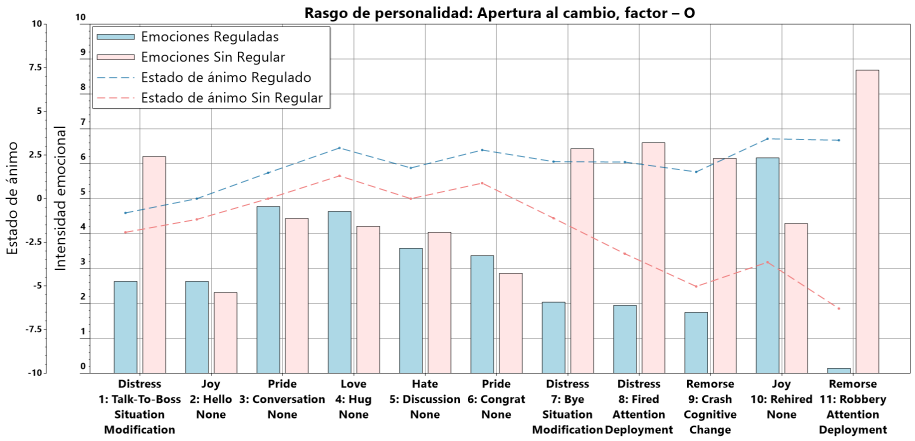


Fig. 4. Gráfica comparativa para el rasgo de personalidad: Apertura al cambio.

- **Evento 7**: María le da la noticia sobre su renuncia (“Bye”),
- **Evento 8**: Sam es despedido de su trabajo (“Fired”),
- **Evento 9**: Sam sufre un percance automovilístico (“Crash”),
- **Evento 10**: Sam recupera su trabajo (“Rehired”),
- **Evento 11**: Robo en casa de Sam (“Robbery”).

Se realizaron un total de 5 simulaciones para representar en el agente cada uno de los 5 tipos de personalidad dominante. En cada simulación el valor para cada rasgo de personalidad se definió de manera aleatoria, excepto el tipo de personalidad dominante en el agente. La configuración en los valoresde los tipos de personalidad para cada simulación fue la siguiente:

- **Simulación 1**: *Apertura al cambio = 85 Escrupulosidad = 30 Extraversión = 15, Amabilidad = 40, Neurótico = 10*,
- **Simulación 2**: *Apertura al cambio = 45 Escrupulosidad = 85 Extraversión = 15, Amabilidad = 40, Neurótico = 10*,
- **Simulación 3**: *Apertura al cambio = 45 Escrupulosidad = 30 Extraversión = 85, Amabilidad = 40, Neurótico = 10*,
- **Simulación 4**: *Apertura al cambio = 45 Escrupulosidad = 30 Extraversión = 15, Amabilidad = 85, Neurótico = 10*,
- **Simulación 5**: *Apertura al cambio = 45 Escrupulosidad = 30 Extraversión = 15, Amabilidad = 40, Neurótico = 85*.

En la gráfica de la Figura 4 se puede apreciar que el estado de ánimo (variable agregada a partir de las emociones positivas y negativas generadas) del agente “Sam” es más positivo cuando se utiliza la arquitectura de regulación emocional (línea de color

azul) que cuando no se hace uso de ella (línea de color rojo). Este "mejor" estado de ánimo se genera al implementar una sola estrategia de regulación emocional en el escenario, provocando que las emociones negativas posteriores se presenten con una menor intensidad.

Por ejemplo, en el evento 5 ("Discussion") el agente no pudo implementar ninguna estrategia de regulación emocional debido a su tipo de personalidad dominante, y que en el escenario este evento solo se configuró para ser evitado. Sin embargo, en el primer evento ("Talk-To-Boss") se implementó la estrategia de selección de la situación (situation selection), de modo que, cuando ocurre el evento "Discussion" el valor de intensidad emocional debido a este evento es menor.

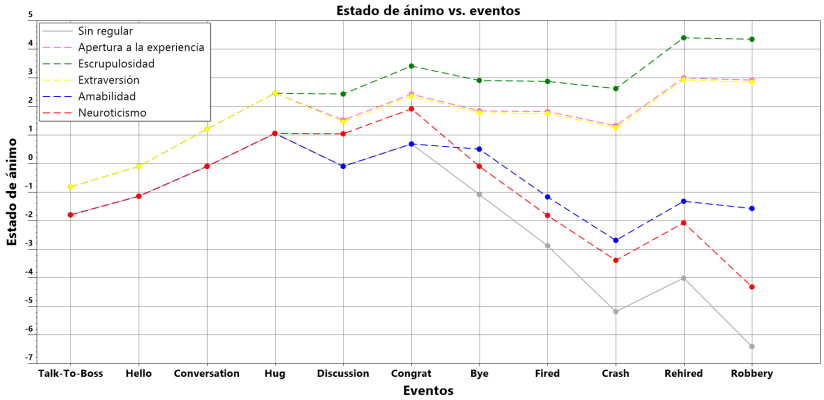


Fig. 5. Gráfica comparativa para los 5 rasgos de personalidad.

Esto es consistente con lo expuesto por Samson y Gross (consultar [19]), quienes explican que el estado de ánimo puede ser visto como un mecanismo de ayuda para la regulación de una emoción, especialmente en la regulación de emociones negativas.

En la Figura 5 se puede observar que el valor del estado de ánimo del agente Sam, y el valor de la intensidad de cada emoción, es diferente para cada tipo de personalidad. Cuando el rasgo dominante en el agente es Escrupulosidad, el estado de ánimo promedio es generalmente mejor que en cualquier otro rasgo de personalidad.

Esto concuerda con los resultados expuestos en [10], ya que este rasgo de personalidad es el que puede implementar un mayor número de estrategias de regulación emocional para cada evento que se presente en el escenario. Por el contrario, cuando el rasgo de personalidad dominante en el agente es de Neuroticismo, solo es posible aplicar una única estrategia de regulación emocional, lo que reduce la posibilidad de regular una emoción negativa.

Lo anterior se representa en la gráfica de la Figura 5, en la que se se comparan los valores de estado de ánimo de los cinco rasgos de personalidad, así como los valores obtenidos del estado de ánimo cuando no se utiliza la arquitectura de regulación emocional (línea gris en el gráfico de la Figura 5).

5. Conclusiones y trabajo futuro

En este artículo se presenta un modelo de regulación emocional y su integración en una arquitectura computacional de emociones. Este tipo de arquitecturas son el componente subyacente para el desarrollo de agentes conversacionales personificados. La generación de reacciones emocionales adecuadas al contexto de aplicación en este tipo de agentes conversacionales contribuyen a mejorar la credibilidad y aceptabilidad por parte de los usuarios.

El modelo de regulación presentado en este artículo está basado en las cinco estrategias de regulación propuestas por J.J. Gross tomando en cuenta los tipos de personalidad como diferencias individuales en la aplicación de cada estrategia.

Los resultados iniciales obtenidos a partir de un conjunto de simulaciones son consistentes con los modelos teóricos utilizados como fundamentos del modelo desarrollado. Como trabajo futuro, se pretende implementar un caso de estudio en el cual se desarrollará un par de agentes virtuales en la plataforma Unity, los cuales harán uso de la arquitectura con el modelo de regulación emocional desarrollado.

Se diseñarán diferentes escenarios interactivos con estos agentes virtuales representando diversos tipos de personalidad. Utilizando este caso de estudio e involucrando a un conjunto de usuarios, se evaluará si los comportamientos emocionales generados por los agentes virtuales con diferentes tipos de personalidad son más creíbles de acuerdo al escenario definido en comparación con los mismos agentes conversacionales, pero sin hacer uso del modelo de regulación emocional.

Agradecimientos. Agradecemos al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo otorgado mediante el programa “CONACYT PRONACE - Salud” para el proyecto 3210 “Desarrollo y evaluación de una plataforma tecnológica de ayuda a la detección, seguimiento e intervención temprana de problemas de salud mental y adicciones en la comunidad escolar, primer y segundo nivel de atención”.

Referencias

1. Barrett, L. F., Mesquita, B., Ochsner, K. N., Gross, J. J.: The experience of emotion. Annual Review of Psychology, vol. 58, pp. 373–403 (2007) doi: 10.1146/annurev.psych.58.110405.085709
2. Bosse, T., Gerritsen, C., de Man, J. d., Treur, J.: Learning emotion regulation strategies: A cognitive agent model. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). vol. 2, pp. 245–252. IEEE (2013) doi: 10.1109/WI-IAT.2013.116
3. Bosse, T., Jonker, C. M., Meij, L. v. d., Treur, J.: LEADSTO: A language and environment for analysis of dynamics by simulation. In: Eymann, T., Klügl, F., Lamersdorf, W., Klusch, M., Huhns, M. N. (eds) German Conference on Multiagent System Technologies (MATES). pp. 165–178. Springer (2005) doi: 10.1007/11550648_15
4. Bosse, T., Pontier, M., Treur, J.: A computational model based on Gross’ emotion regulation theory. Cognitive Systems Research, vol. 11, no. 3, pp. 211–230 (2010) doi: 10.1016/j.cogsys.2009.10.001

5. Dias, J., Mascarenhas, S., Paiva, A.: Fatima modular: Towards an agent architecture with a generic appraisal framework. In: Bosse, T., Broekens, J., Dias, J., van der Zwaan, J. (eds) Emotion modeling, pp. 44–56. Lecture Notes in Computer Science, Springer (2014)
6. Ekman, P.: An argument for basic emotions. Cognition and emotion, vol. 6, no. 3–4, pp. 169–200 (1992) doi: 10.1080/02699939208411068
7. Gross, J. J., Thompson, R. A.: Emotion regulation: Conceptual foundations. Handbook of Emotion Regulation, pp. 3–27 (2007)
8. Gross, J. J., Feldman, B. L.: Emotion generation and emotion regulation: One or two depends on your point of view. Emotion review, vol. 3, no. 1, pp. 8–16 (2011) doi: 10.1177/1754073910380974
9. Hu, X., Zhang, H., Yuan, Y., Chen, Y., Zhong, M.: A model for reappraisal with personality in emotion regulation. International Journal of Computer Integrated Manufacturing, vol. 33, no. 10–11, pp. 1197–1207 (2020) doi: 10.1080/0951192X.2020.1718769
10. John, O. P., Gross, J. J.: Individual differences in emotion regulation. Handbook of emotion regulation, pp. 351–372 (2007)
11. Kopp, S., Gesellensetter, L., Krämer, N. C., Wachsmuth, I.: A conversational agent as museum guide–design and evaluation of a real-world application. In: Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T. (eds) International workshop on intelligent virtual agents. pp. 329–343. Lecture Notes in Computer Science (2005) doi: 10.1007/11550617_28
12. Marsella, S., Gratch, J., Petta, P.: Computational models of emotion. Blueprint for Affective Computing-A sourcebook and manual, vol. 11, no. 1, pp. 21–46 (2010)
13. Marsella, S. C., Johnson, W. L., LaBore, C.: Interactive pedagogical drama. 4th International conference on autonomous agents. In: Proceedings of the fourth international conference on Autonomous agents. pp. 301–308 (2000) doi: 10.1145/336595.337507
14. Martínez-Miranda, J.: Embodied conversational agents for the detection and prevention of suicidal behaviour: current applications and open challenges. Journal of medical systems, vol. 41, no. 9, pp. 1–14 (2017) doi: 10.1007/s10916-017-0786
15. McCrae, R. R., John, O. P.: An introduction to the five-factor model and its applications. Journal of personality, vol. 60, no. 2, pp. 175–215 (1992) doi: 10.1111/j.1467-6494.1992.tb00970.x
16. Ortony, A., Clore, G. L., Collins, A.: The cognitive structure of emotions. Cambridge University press (1988)
17. Purnamaningsih, E. H.: Personality and emotion regulation strategies. International Journal of Psychological Research, vol. 10, no. 1, pp. 53–60 (2017) doi: 10.21500/20112084.2040
18. Russell, S., Norvig, P.: Inteligencia Artificial: Un enfoque moderno. Pearson, 2nd edn. (2004)
19. Samson, A. C., Gross, J. J.: Humour as emotion regulation: The differential consequences of negative versus positive humour. Cognition & emotion, vol. 26, no. 2, pp. 375–384 (2012) doi: 10.1080/02699931.2011.585069
20. Solomon, R. C.: Not passion's slave: emotions and choice. Oxford University Press (2007)
21. Treur, J.: A cognitive agent model displaying and regulating different social response patterns. In: Twenty-Second International Joint Conference on Artificial Intelligence. pp. 1735–1742 (2011)
22. Treur, J.: Network-oriented modeling: Addressing complexity of Cognitive, Affective and Social Interactions. Springer (2011)
23. Ullah, N., Gao, Z., Liu, R., Treur, J.: A second-order adaptive temporal-causal network model for age and gender differences in evolving choice of emotion regulation strategies. Journal of Information and Telecommunication, vol. 4, no. 2, pp. 213–228 (2020) doi: 10.1080/24751839.2020.1724738

ISSN 1870-4069

Estimación de esfuerzo en desarrollo de software ágil utilizando redes neuronales artificiales

Eduardo Rodríguez Sánchez, Eduardo Vázquez Santacruz,
Humberto Cervantes Maceda

Universidad Autónoma Metropolitana,
Posgrado en Ciencias y Tecnologías de la Información,
México

erodsmx@gmail.com, evazquez.santacruz@izt.uam.mx,
hcm@xanum.uam.mx

Resumen. La estimación de esfuerzo es importante para planificar correctamente el uso de recursos en un proyecto de TI. En la fase de planeación de un proyecto, durante la elaboración de los artefactos de visión y alcance, el equipo involucrado realiza una estimación inicial aproximada de tiempo y costo. Para mejorar la precisión de la estimación de esfuerzo en desarrollo de software existen varias técnicas de estimación: function points, object points, use case points, story points, etc. Los puntos de historia de usuario o story points, son la base de marcos de trabajo ágil que actualmente están tomando más fuerza en el desarrollo de software. Una de las principales lagunas de conocimiento se encuentra en la aplicación de redes neuronales y aprendizaje profundo en estimación de esfuerzo en desarrollo ágil. Este artículo contribuye a fortalecer el uso de redes neuronales como método de estimación de esfuerzo a nivel de proyecto en marcos de trabajo que usan un enfoque de puntos de historia de usuario como Scrum. Este trabajo de investigación presenta un estudio sobre el análisis de la precisión de la predicción del proceso de estimación ejecutado con técnicas de aprendizaje automático, tomando como referencia el modelo de estimación de esfuerzo para el desarrollo de software ágil propuesto en [8]. Los algoritmos de aprendizaje supervisado propuestos para realizar las estimaciones son redes neuronales de tipo perceptrón multicapa y redes neuronales recurrentes. El desempeño de los modelos se compara a través del error cuadrático medio, error relativo medio y el coeficiente de determinación R^2. Los algoritmos se ejecutan aplicando validación cruzada 10-Fold. Los resultados comparan la ejecución de los algoritmos con y sin la propuesta de utilizar categorías de clasificación por tamaño probando que el uso de categorías mejora la precisión de la predicción.

Palabras clave: Estimación, tiempo, costo, desarrollo ágil, aprendizaje automático, redes neuronales.

Effort Estimation in Agile Software Development Using Artificial Neural Networks

Abstract. Effort estimation is important to correctly plan the use of resources in an IT project. In the planning phase of a project, during the elaboration of the

vision and scope artifacts, the team involved makes an initial rough estimate of time and cost. To improve the accuracy of effort estimation in software development, there are several estimation techniques: function points, object points, use case points, story points, etc. User story points or story points are the basis of agile frameworks that are currently gaining strength in software development. One of the main knowledge gaps is in the application of neural networks and deep learning in effort estimation in agile development. This article contributes to strengthening the use of neural networks as a project-level effort estimation method in frameworks that use a user story point approach such as Scrum. This research paper presents a study on the analysis of the prediction accuracy of the estimation process executed with machine learning techniques, taking as reference the effort estimation model for agile software development proposed in [8]. The supervised learning algorithms proposed to perform the estimations are multilayer perceptron-type neural networks and recurrent neural networks. The performance of the models is compared through the mean square error, mean relative error and the determination coefficient R^2. The algorithms are executed applying 10-Fold cross-validation. The results compare the execution of the algorithms with and without the proposal to use size classification categories, proving that the use of categories improves the prediction accuracy.

Keywords: Estimation, time, cost, agile development, machine learning, neural networks.

1. Introducción

El proceso de estimación en el desarrollo de software es una actividad complicada para el equipo involucrado [3]. La planeación de un proyecto contempla tres elementos: recursos, alcance y fecha de entrega. La estimación requiere de una aproximación a los costos, tiempos, y factores que pueden afectar el desarrollo del proyecto [8], también conocidos como riesgos. Un modelo de estimación ayuda a tomar acciones de control y mejores decisiones que impactan directamente en la organización.

El principal problema es que cuando no se aplica un modelo de estimación, ni se realizan proyecciones adecuadas, las fechas estimadas no consideran el trabajo relacionado y las predicciones son aleatorias poniendo en riesgo al proyecto al no tener actividades y fechas claras.

El impacto de una mala planeación se refleja en entregas tardías, aumento de costos, deuda técnica, y a largo plazo posible pérdida de clientes. En el 14vo. informe anual del estado ágil [12] se reporta que una de las principales razones para adoptar un enfoque ágil es la aceleración en la entrega de software y la mejora en la habilidad de adaptarse al cambio de prioridades.

Como lo menciona el informe, el éxito de un proyecto se mide a través del valor de negocio entregado, la satisfacción del cliente, la velocidad del desarrollo, el presupuesto vs el costo actual, la planeación vs la ejecución real de las historias, entre otras características. En metodologías ágiles como Scrum, una historia de usuario es un enunciado corto que refleja los requerimientos funcionales del sistema, similar a un caso de uso.

La estructura de una historia de usuario permite identificar al actor, la acción que quiere realizar y la justificación de por qué el sistema debe realizar dicha acción. El conjunto de historias que componen el sistema se implementan en ciclos de trabajo conocidos como *Sprints*.

El presente proyecto de investigación está orientado a proponer una solución que apoye el proceso de estimación de proyectos desarrollados con un enfoque ágil. Aplicando técnicas de aprendizaje supervisado, este trabajo proporciona un modelo de estimación de esfuerzo híbrido basado en redes neuronales artificiales e historias de usuario que en conjunto con categorías de clasificación se mejora la precisión de la predicción de las estimaciones de tiempo de finalización y costo total de un proyecto.

En la sección 2 se presenta el trabajo relacionado, resaltando aquellas investigaciones que se encuentran dentro del mismo campo de estudio: desarrollo de software ágil utilizando puntos de historia. Posteriormente en la sección 3 se desglosa la propuesta del trabajo de investigación, seguido del marco teórico.

En la sección 5 se desglosa la evaluación empírica reportando la estructura de los experimentos, el diseño de los modelos de aprendizaje automático utilizados y los criterios de evaluación aplicados. Posteriormente se describen los resultados agrupando las gráficas y tablas correspondientes a la evaluación de los modelos y la comparativa con el estado del arte. Por último se presentan las conclusiones y el trabajo a futuro.

2. Trabajo relacionado

El desarrollo de software ágil surgió en la primera década del siglo XXI, en 2002 Grenning propuso el uso de Planning Poker como técnica de estimación y más tarde en 2005 M. Cohn sugirió que este método era útil en estimación de proyectos ágiles [10]. Coelho [9] destaca la importancia de los puntos de historia de usuario y la velocidad como medición del progreso por iteración del proyecto.

Fue hasta el año 2012, que Wen J. y otros investigadores sugirieron el aprendizaje automático como otra categoría de las técnicas de estimación [10]. M. Gultekin & O. Kalipsiz [19] prueban técnicas de aprendizaje automático y dos notaciones de puntos de historia, aritmética y fibonacci, probando que la segunda da mejor precisión al realizar estimaciones.

Marta Fernández et al. [15] realiza una revisión sistemática de la literatura destacando investigaciones sobre métodos ágiles y clasificando los trabajos dentro de las categorías de juicio de experto (Planning Poker, Wideband Delphi) y aquellas basadas en datos (técnicas de aprendizaje automático).

Dentro de los modelos de estimación de esfuerzo se encuentran el trabajo de Zia [8] orientado a producir estimaciones de tiempo y costo de un proyecto utilizando un modelo de regresión lineal. Rashmi Popli & Naresh Chauhan [13] calculan el costo, esfuerzo y duración de un proyecto pequeño y mediano usando un enfoque de historias de usuario.

Atef Tayh Raslan et al. [4] propone mejorar la precisión de la estimación aplicando un modelo de estimación basado en puntos de historia y lógica difusa. Sakshi Garg et al. [16] plantea un modelo basado en Principal Component Analysis y programación con restricciones asociado a los costos del desarrollo de software.

Tabla 1. Micro estado del arte, selección de tres artículos que comparte el mismo modelo de estimación y conjunto de datos.

No	Autor	Título	Técnica de Machine Learning reportadas	Precisión(%) Tiempo	Precisión(%) Costo	Entradas	Salidas	Criterios de evaluación	Dataset
1	Aditi Panda et al.	Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points	General Regression Neural Network	85.92	No	Total de puntos de historia y velocidad inicial del proyecto	Predicción de tiempo	Error cuadrático medio	21 proyectos desarrollados por seis compañías
			Probabilistic Neural Network	87.66				Coef ciente de determinación (R^2)	
			GMDH Polynomial Neural Network	89.67				Magnitud media de error relativo	
			Cascade-Correlation Neural Network	94.76				Porcentaje de precisión	
2	Ch. Prasada Rao et al.	An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques	Adaptive Neuro-Fuzzy Interface System	76.19	57.14		Predicción de tiempo y costo	Magnitud media de error relativo	
			Generalized Regression Neural Networks	76.19	76.19			Porcentaje de precisión	
			Radial Basis Function Networks	76.19	76.19				
3	S. M. Satapathy, S. K. Rath	Empirical assessment of machine learning models for agile software development effort estimation using story points	Decision Tree	90.48	No		Predicción de tiempo	Error absoluto medio	
			Random Forest	95.24				Magnitud media de error relativo	
			Stochastic Gradient Boosting (SGB)	95.24				Porcentaje de precisión	

E. Scott, & D. Pfahl [5] proponen la estimación de puntos de historia a través de un modelo que asigna puntos de historia a los reportes de incidencia tomando en cuenta el perfil del desarrollador. Jitender Choudhari y Ugrasen Suman [11] plantean un modelo basado en Extreme Programming (XP) que usa puntos de historia para calcular el volumen de esfuerzo en fase de mantenimiento.

O. Malgonde & K. Chari [17] plantea un modelo para la predicción de esfuerzo de la historia de usuario mediante un modelo predictivo, un modelo basado en ensamble y un modelo de optimización para el esfuerzo. Otras propuestas con un enfoque a nivel de historia de usuario son los trabajos de Morakot Choetkiertikul et al.[7] y M. Durán et al. [18] en donde se propone un modelo de predicción basado en redes neuronales de tipo LSTM y Recurrent Highway, y un método para estimar la complejidad de la historia a través de su descomposición utilizando redes bayesianas respectivamente.

El presente trabajo de investigación tiene particular interés en el modelo propuesto por Zia et al. [8], con enfoque a nivel de proyecto. El modelo de Zia forma parte del conjunto de modelos de estimación de esfuerzo en desarrollo ágil y genera un porcentaje de precisión del 57.14 % para tiempo de finalización y 61.40 % para costo total.

Tres investigaciones retomaron este modelo para realizar estimaciones utilizando aprendizaje automático, Aditi Panda et al. [1] aplicando cuatro tipo de redes neuronales, Ch. Prasada Rao et al. [3] con tres tipos de redes neuronales, y S. M. Satapathy, S. K. Rath [6] con árboles de decisión, bosques aleatorios y Stochastic Gradient Boosting. En la tabla del Cuadro 1 se agrupan estas tres investigaciones que toman como datos de entrada los 21 proyectos de [8].

En los tres casos, los autores evalúan los modelos de aprendizaje automático con el error relativo y generan un porcentaje de precisión de la predicción. La comparativa de la tabla 1 muestra que los árboles de decisión ofrecen los mejores resultados. El trabajo de Satapathy [6] reconoce que hay poca disponibilidad de investigación para proporcionar un procedimiento sistemático con el fin de estimar el esfuerzo de los proyectos desarrollados con metodología ágil, por lo que el presente proyecto pretende

contribuir a esta línea de conocimiento.

3. Propuesta

El presente trabajo de investigación tiene la finalidad de estudiar algoritmos de aprendizaje automático, en particular redes neuronales artificiales aplicadas a la estimación de tiempo de finalización y costo total de un proyecto de software.

El estudio desglosa un análisis de la precisión de la predicción del proceso de estimación ejecutado con redes neuronales y aplicando un enfoque de puntos de historia de usuario. La hipótesis del proyecto establece que el uso de categorías de tamaño ofrece estabilidad en la precisión de la predicción de tiempo y costo de los proyectos, lo que reduce la desviación estándar de las estimaciones.

La propuesta contiene un modelo de estimación de esfuerzo basado en regresión lineal que considera el esfuerzo medido en puntos de historia de usuario, la velocidad del equipo,las categorías de clasificación de tamaño esfuerzo, tiempo y costo de un proyecto, así como un modelo de aprendizaje automático compuesto por una red neuronal de tipo perceptrón multicapa y una red neuronal recurrente encargadas de realizar las estimaciones.

A diferencia de los trabajos del estado del arte, ambas técnicas aplicadas componen un ensamble el cual mejora las estimaciones emitidas al promediar los resultados de las técnicas individuales.

Para probar la hipótesis establecida el proyecto genera dos escenarios, el primero recibe como entrada el esfuerzo medido como el total de puntos de historia de usuario requeridos (*Effort*), y la velocidad inicial del equipo (V_i).

Un segundo escenario ejecuta el modelo teniendo como entrada el esfuerzo, la velocidad y las categorías de clasificación de tamaño. Los resultados se comparan para mostrar que las categorías generan mejores predicciones. Los tamaños para las categorías de clasificación son:

$$0 = \text{Grande} = L, \quad 1 = \text{Mediano} = M, \quad 2 = \text{Pequeño} = S.$$

El conjunto de datos consiste en un total de 21 proyectos desarrollados por seis compañías de software en Pakistán, por lo que el costo de los mismos está dado en rupias pakistaníes. El método más simple para clasificar los datos consiste en dividirlos en tres grupos de siete, ordenados por esfuerzo, tiempo y costo.

Cada proyecto tiene un conjunto de características que lo definen como el total de esfuerzo para completar el proyecto, la velocidad del equipo, la desaceleración debido a factores de fricción y fuerzas dinámicas (riesgos), el tamaño del Sprint, los días laborales en el mes y el salario del equipo. Realizando un análisis de componentes principales, las características más importantes son el esfuerzo y la velocidad.

Los modelos de aprendizaje automático se entrenan utilizando 10-Fold cross-validation para evaluar la capacidad de generalización de los algoritmos. Debido a que el conjunto de datos es pequeño se usa una técnica llamada aumento de datos, la cual consiste en duplicar muestras agregando pequeñas cantidades de ruido a las características principales (esfuerzo y velocidad).

Esto da como resultado dos valores con ruido por cada proyecto, teniendo un total de 42 valores con ruido utilizados en entrenamiento y 21 valores sin ruido para prueba (datos reales). Con esta técnica se logra reducir el sobreentrenamiento u *overfitting*.

Adicionalmente los modelos de redes neuronales cuentan con dos capas de ruido gaussiano las cuales toman como datos de entrada la salida de las neuronas de la capa anterior y agregan ruido gaussiano, esto permite modificar los valores en cada época y permite un mejor ajuste del algoritmo de retropropagación al calcular los pesos de la red en cada ciclo.

Aplicando k-Fold los datos se dividen en k subconjuntos de aproximadamente el mismo tamaño y cada partición se usa para entrenar el modelo. De este subconjunto el 30 % se reserva para validación, la cual ocurre después de cada época de entrenamiento para ayudar a determinar si ocurre *underfitting* u *overfitting*. Una vez que finaliza el entrenamiento de los modelos, se emiten las predicciones tomando como valores de entrada los datos reales de los 21 proyectos.

4. Marco teórico

Los elementos clave del proyecto son el modelo de estimación algorítmico y las técnicas de aprendizaje automático descritas a continuación.

4.1. Modelo de estimación de esfuerzo en desarrollo ágil

El modelo de estimación basado en regresión lineal [8] emite estimaciones de tiempo y costo para 21 proyectos desarrollados por 6 organizaciones, los conceptos resumidos más relevantes del modelo son los siguientes.

Tamaño de la historia de usuario: Con una escala propuesta de uno a cinco, siendo uno el tamaño más pequeño y cinco una historia épica:

$$[1, 5] = \{S \in \mathbb{N} : 1 \leq S \leq 5\}.$$

Complejidad: Asocia la dificultad técnica:

$$[1, 5] = \{C \in \mathbb{N} : 1 \leq C \leq 5\}.$$

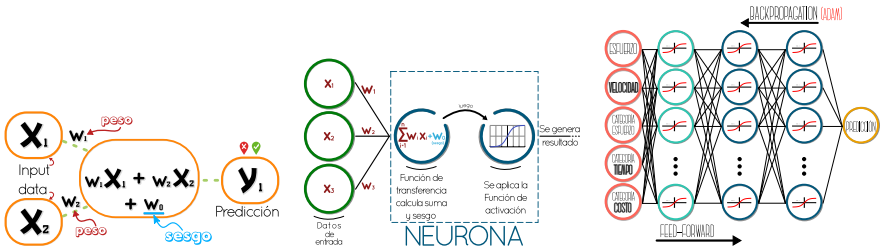
Esfuerzo: Combina el tamaño de la historia (S) y su complejidad (C):

$$E_S = C \times S.$$

El esfuerzo de un proyecto es la suma de los esfuerzos de las historias:

$$E = \sum_{i=1}^{n} (E_S)_i \, .$$

Velocidad: Considera el esfuerzo total (puntos de historia) y el tiempo (duración del sprint):



(a) Composición de un perceptrón simple.

(b) Proceso ejecutado en una neurona

(c) Red neuronal de tipo feedforward

Fig. 1. Una red neuronal de tipo feed forward (c), se compone de perceptrones interconectados entre sí donde los elementos de una capa conectan con las neuronas de la capa siguiente.

$$V_i = \frac{\text{Unidades de esfuerzo completado}}{\text{Sprint}}.$$

Tiempo de finalización: Duración necesaria para completar el proyecto:

$$T = \frac{\text{Esfuerzo}}{\text{Velocidad}}.$$

T se mide en días. Dividiendo T entre el número de días laborales en el mes se obtiene el número de meses para completar el proyecto. Costo total: Contempla una relación de los recursos económicos invertidos como salarios, equipo tecnológico, licencias,marketing, rentas, mobiliario, etc:

$$\text{Costo} = (\alpha T_S)\, T.$$

El valor de α corresponde al valor neto de la relación de los costos asociados al proyecto. T_S es el salario mensual del equipo y T es el tiempo calculado en meses. En la siguiente sección se presentan los modelos que construyen un mecanismo capaz de predecir tiempo de finalización y costo total a partir de los datos de entrada.

4.2. Redes neuronales artificiales

Las redes neuronales artificiales (ANN) son un intento de modelar la capacidad del sistema nervioso para procesar información [2]. Una red neuronal se compone de perceptrones (Figura 1c) y un perceptrón es un modelo simplificado de una neurona capaz de realizar clasificación binaria.

La composición de un perceptrón se muestra en la Figura 1a y se puede pensar como una combinación lineal, el proceso de activación de una neurona se muestra en 1b.

Retomando la definición matemática de una red neuronal de Hornik et al. [14] se tiene que: Para todo $r \in \mathbb{N} \equiv \{1, 2, ...\}$, A^r es el conjunto de todas las funciones afines de $\mathbb{R}^r a\, \mathbb{R}$, esto es, el conjunto de todas las funciones de la forma $A(x) = w \cdot x + b$ donde *w* y *x* son vectores en R^r y *b* es un escalar.

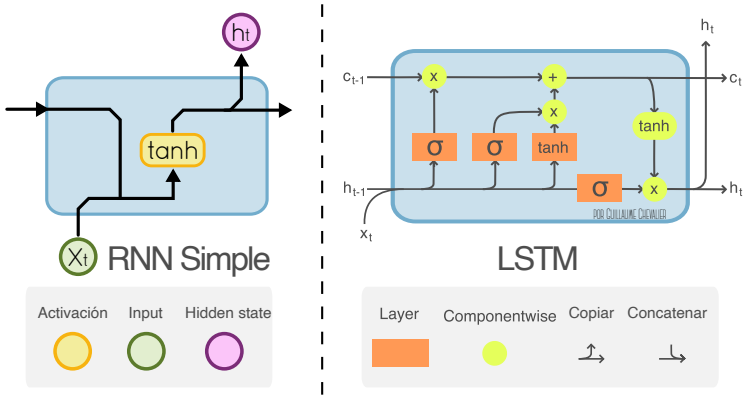


Fig. 2. Una red neuronal recurrente incorpora el concepto de memoria implementando el estado oculto y una capa de activación a la entrada.

$A(x)$ es entonces el producto punto de dos vectores y la adición de un escalar, w corresponde a los pesos de la red y x corresponde a las entradas de la red. El sesgo de una red corresponde al escalar b. Partiendo de esta definición varios autores probaron que una red neuronal de tipo *feedforward* con una capa oculta puede aproximar una función continua multivariante [14].

Red neuronal recurrente El siguiente tipo de red neuronal que mejora la arquitectura del perceptrón multicapa de la Figura 1c son las redes recurrentes o RNN *(Recurrent Neural Network)* que incorporan el concepto de memoria. Matemáticamente una red recurrente simple se formula como:

$$\begin{aligned} h(t) &= f_H\left(W_{IH}\, x(t) + W_{HH}\, h\left(t-1\right)\right) \\ y(t) &= f_O\left(W_{HO}\, h\left(t\right)\right), \end{aligned} \tag{1}$$

donde $x(t)$ y $y(t)$ son los vectores de entrada y salida. W_{IH}, W_{HH}, W_{HO}, son las matrices de los pesos y f_h, f_O son las funciones de activación ocultas y la de salida. En la Figura 2 se muestra una celda recurrente simple y una celda LSTM (*Long Short Term Memory*).

4.3. Métricas de evaluación

El entrenamiento de los modelos diseñados se someten a distintas funciones de error para evaluar su desempeño. Se busca aquella configuración de modelo que disminuya más el error de las predicciones.

Error Cuadrático Medio: La función error cuadrático medio se define como:

$$MSE = (y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}\left(y_i - \hat{y}_i\right)^2, \tag{2}$$

donde n es el número de muestras, y son los datos reales, $\hat{y}$ las predicciones.

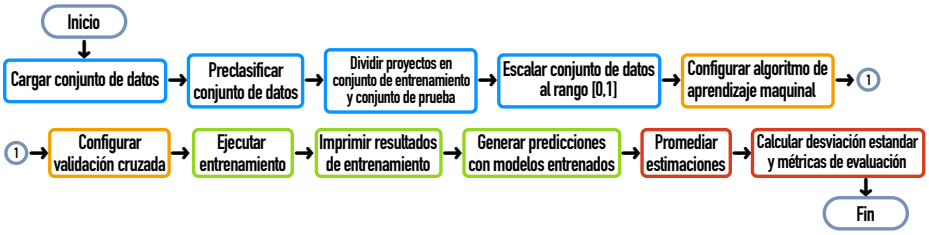


Fig. 3. Diagrama de flujo de los experimentos.

Precisión: La precisión del modelo se calcula utilizando el valor del error relativo *MRE*. El porcentaje de precisión es una primera aproximación de fácil interpretación para conocer qué tan buena es la técnica evaluada:

$$MRE = \frac{|AE_i - PE_i|}{AE_i}, \quad \text{AE = Actual, PE = Predicción ,}$$
$$\text{Precisión } (\%) = (1 - MRE) \times 100. \tag{3}$$

Coeficiente de determinación (R^2): Proporciona una indicación de que tan bueno es el entrenamiento, y por lo tanto, una medida de qué tan probable es que el modelo prediga las muestras no conocidas a través de la proporción de varianza. Si el valor de R^2 es cercano a 1, consideraremos que el algoritmo de aprendizaje automático es bueno o que ofrece resultados confiables:

$$R^2\left(y, \hat{y}\right) = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2}, \quad \text{donde } \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

Explained variance score: Mide la proporción de la variación de un conjunto de datos determinado:

$$\text{explained\_variance}(y, \hat{y}) = 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}.$$

5. Experimentación y Resultados

La ejecución de los experimentos aplica la estrategia de la Figura 3. En una primera etapa se preparan los datos, preclasificandolos y escalándolos al rango $[0, 1]$, posteriormente se configura el algoritmo de aprendizaje automático y la validación cruzada 10-Fold. Al ejecutar el entrenamiento se aplican e imprimen las métricas de evaluación.

Cada partición de la validación cruzada genera un modelo entrenado que emite un conjunto de 21 predicciones. El valor final de tiempo y costo de cada proyecto consiste en el promedio de las estimaciones de los modelos entrenados. Al promediar todos los resultados se calcula la desviación estándar y se ejecutan las métricas de evaluación finales. La configuración de los dos modelos aplicados se muestran en las tablas del Cuadro 2a y 2b.

Tabla 2. Configuraciones para perceptrón multicapa y red neuronal recurrente simple.

Parámetro					
Configuración	Optimizador	Adam			
	Activación	Tanh			
	Capas Ocultas	2			
	Neuronas p/Capa	8			
	Tasa de aprendizaje	0.001			
	Función de pérdida (Loss)	Error Cuadrático Médio (MSE)			
	k-Fold Cross Validation	RepeatedKFold con 10-Fold y n_repeats=2			
	Normalización de datos	MinMaxScaler con valores entre (0,1)			
	Entradas (columnas)	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost
	Salida	Tiempo		Costo	
Regularización	Dropout	No			
	Ruido Gaussiano	stddev=0.00001			
	No. Capas de ruido	2			
Máximo de épocas en entrenamiento		2000			

(a) Perceptrón Multicapa (MLP).

Parámetro					
Configuración	Optimizador	Adam			
	Activación	Relu			
	Activación recurrente	Tanh			
	Capas Ocultas	2			
	Tipo de celda recurrente	Simple RNN			
	Neuronas p/Capa RNN	40			
	Neuronas p/Capa Dense	10			
	Tasa de aprendizaje	0.001			
	Loss	Error Cuadrático Medio (MSE)			
	Normalización de datos	MinMaxScaler en el rango (0,1)			
	Entradas (columnas)	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost
	Salida	Tiempo		Costo	
Regularización	Dropout recurrente	0.1			
	Ruido Gaussiano	stddev=0.00001			
	No. Capas de ruido	1			
Máximo de épocas en entrenamiento		900			

(b) Red Neuronal Recurrente Simple.

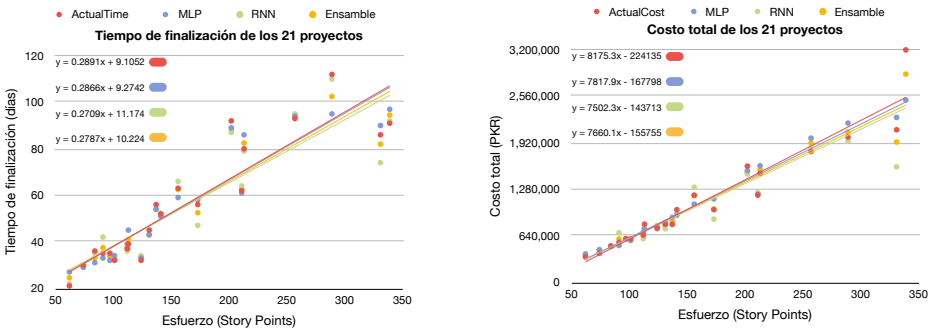


Fig. 4. Comparativa de estimaciones generadas con MLP, RNN y el ensamble.

Ambos casos consisten en redes neuronales de arquitectura pequeña ya que solo contienen capa de entrada, capa de salida y dos capas ocultas. Para mejorar la capacidad de generalización de las redes neuronales se aplica ruido gaussiano a la salida de las neuronas de las capas ocultas, lo que resulta en una estructura de seis capas aunque las capas de ruido solo modifican los valores de salida de las neuronas de la capa anterior y no contienen neuronas o pesos asociados.

Los modelos de redes neuronales se construyen utilizando el API de Keras, programadas en python y utilizando Scikit-learn para ejecutar la validación cruzada. La implementación se realiza de manera remota a través de la versión gratuita de Google Colab que brinda dos procesadores Intel(R) Xeon(R) CPU @ 2.20GHz, 13 GB de memoria RAM, 106 GB de espacio en disco y una GPU Tesla K80 con (x2) 2496 cores a una velocidad de 560 MHz y (x2) 12GB de memoria GDDR5 @ 2500 MHz. La ejecución del modelo basado en MLP con validación cruzada genera 20 estimadores debido a que se ejecuta una repetición de 10-Fold, por cada partición generada se entrena un modelo. La estimación de tiempo y costo corresponde al promedio de las estimaciones realizadas por todos los modelos entrenados. En el caso de la red neuronal recurrente la validación cruzada genera una ventana de n proyectos aumentando en tamaño en cada iteración.

Se ejecutan dos modelos de redes neuronales, uno para predecir tiempo de finalización y otro para las estimaciones de costo. La variación en número de neuronas indica

Tabla 3. Resultados del presente trabajo de investigación.

Parámetro		Modelo MultiLayer Perceptron (MLP)				Modelo Recurrent Neural Network (RNN)			
Datos	Entradas (columnas)	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost	Effort, Vi	Eff, Vi SizeEffort, SizeTime, SizeCost
	Salida	Tiempo		Costo		Tiempo		Costo	
Prueba	Precisión (%)	**91.61**	**93.14**	**90.47**	**92.52**	**91.47**	**95.27**	**89.49**	**93.44**
	R^2	**0.9399**	**0.9614**	**0.9201**	**0.9341**	**0.9441**	**0.9751**	**0.9547**	**0.9658**
	MRE	0.0839	0.0686	0.0953	0.0748	0.0853	0.0473	0.1051	0.0656
	MSE	39.00	25.05	37,851,092,256	31,223,726,814	36.48	16.24	21,482,779,729	16,207,233,554
	RMSE	6.25	5.00	194,554	176,702	6.04	4.03	146,570	127,308
	Explained Variance	0.9402	0.9615	0.9201	0.9341	0.9477	0.9764	0.9584	0.9677
Máximo de épocas en entrenamiento		2000				900			
Promedio de épocas en entrenamiento		784	1196	473	700	-			

(a) Comparativa del uso de categorías de clasificación.

	Algoritmo	MLP	RNN	Ensamble	MLP	RNN	Ensamble
Datos	Entradas	Effort, Vi, SizeEffort, SizeTime, SizeCost			Effort, Vi, SizeEffort, SizeTime, SizeCost		
	Salidas	Tiempo			Costo		
Resultados	Predicción(%)	93.14	95.27	**95.58**	92.52	93.44	**95.58**
	Coef. determinación	0.9614	0.9751	**0.9855**	0.9341	0.9658	**0.9814**
	Error relativo MRE	0.0686	0.0473	**0.0442**	0.0748	0.0656	**0.0442**
	RMSE	5.00	4.03	**3.07**	176,702	127,308	**93,762**
	Explained Variance	0.9615	0.9764	**0.9856**	0.9341	0.9677	**0.9820**

(b) Métricas de evaluación.

Author	Estimación	Técnica	Precisión (%)	R^2	MMRE
Zia et al.	Tiempo	No Aplica	57.14	No incluida	0.0719
	Costo		61.90	No incluida	0.0576
Aditi Panda et al.	Tiempo	Cascade Correlation Neural Network	94.76	0.9303	0.1486
SM Satapathy et al.	Tiempo	Stochastic Gradient Boosting	95.24	No incluida	0.1632
Ch Prasada Rao et al.	Tiempo	Generalized Regression Neural Network	76.19	No incluida	0.0483
	Costo		76.19	No incluida	0.0276
Proyecto de investigación	Tiempo	Ensamble de Redes Neuronles	95.58	0.9855	0.0442
	Costo		95.58	0.9814	0.0442

(c) Resultados de la literatura.

que a mayor número de neuronas se reduce el número de épocas para entrenar los modelos, lo que reduce a su vez el tiempo de ejecución. El número mínimo de neuronas es igual a la dimensión de los datos de entrada y el número máximo de neuronas es de diez, al seguir incrementando el número de neuronas la red no mejora la precisión de la predicción. Los experimentos arrojan que ambas redes dan buenos resultados cuando se utilizan ocho neuronas en cada capa.

Como se observa en la Tabla 3a del Cuadro 3 las estimaciones realizadas por los algoritmos mejoran al clasificar los proyectos pues en los casos de tiempo y costo se logró reducir el error y aumentar los coeficientes de determinación. En la Figura 4 se grafican en rojo el tiempo de finalización real y el costo total de los 21 proyectos, en azul las estimaciones hechas con MLP, en verde las predicciones de la red neuronal recurrente y en amarillo el ensamble formado por ambas técnicas.

Como se puede observar en las rectas de regresión los mejores resultados se obtienen con la red neuronal de tipo perceptrón multicapa ya que fue el único algoritmo capaz de reducir el sobreentrenamiento. Sin embargo, el objetivo del modelo híbrido de estimación de esfuerzo consiste en aplicar distintas técnicas para obtener diversos puntos de comparación, por este motivo no se seleccionan solo los resultados del perceptrón multicapa.

Al tomar el promedio de las estimaciones realizadas por los dos algoritmos, se generan valores de tiempo de finalización y costo total que se comparan con los valores reales de los proyectos. En la tabla del Cuadro 4 se comparan los valores obtenidos al aplicar el modelo original propuesto en [8] con los valores centrales obtenidos por el ensamble.

Las primeras columnas corresponden a los valores de tiempo en días, ordenando

Tabla 4. Tabla comparativa de las diferencias de los valores predichos con los valores reales de cada proyecto y las estimaciones del modelo original.

No.	ActualTime	Zia et al (días)	ΔT Zia (días)	Ensamble	ΔT (días)	ActualCost	Zia et al. (PKR)	ΔC Zia (PKR)	Ensamble	ΔC (PKR)	¿Mejora Tiempo?	¿Mejora Costo?
1	63	58	5	63 ± 4	1	1,200,000	1,023,207	176,793	1,192,138 ± 116,558	7,863	Sí	Sí
2	92	81	11	88 ± 1	4	1,600,000	1,680,664	80,664	1,514,478 ± 24,438	85,522	Sí	No
3	56	52	4	53 ± 6	4	1,000,000	992,270	7,730	1,008,666 ± 138,458	8,666	Sí	No
4	86	87	1	82 ± 8	4	2,100,000	2,002,767	97,233	1,930,933 ± 339,441	169,067	No	No
5	32	29	3	34 ± 1	2	750,000	676,081	73,919	760,036 ± 21,891	10,036	Sí	Sí
6	91	95	4	95 ± 3	4	3,200,000	2,895,133	304,867	2,866,229 ± 356,537	333,772	Sí	No
7	35	29	6	34 ± 2	2	600,000	540,114	59,886	600,986 ± 3,033	986	Sí	Sí
8	93	84	9	95 ± 1	2	1,800,000	1,614,079	185,921	1,918,859 ± 66,126	118,859	Sí	Sí
9	36	35	1	33 ± 2	3	500,000	507,265	7,265	496,287 ± 1,980	3,713	No	Sí
10	62	66	4	63 ± 2	1	1,200,000	1,267,180	67,180	1,223,608 ± 12,456	23,608	Sí	Sí
11	45	41	4	43 ± 0	2	800,000	786,732	13,268	780,428 ± 42,805	19,573	Sí	No
12	37	39	2	38 ± 2	1	650,000	597,143	52,857	643,424 ± 43,528	6,576	Sí	Sí
13	32	35	3	33 ± 1	1	600,000	538,495	61,505	577,724 ± 4,662	22,277	Sí	Sí
14	30	26	4	30 ± 1	1	400,000	394,546	5,454	446,285 ± 2,913	46,285	Sí	No
15	21	22	1	25 ± 3	4	350,000	330,561	19,439	372,355 ± 18,420	22,355	No	No
16	112	103	9	103 ± 8	10	2,000,000	1,971,485	28,515	2,069,168 ± 122,723	69,168	No	No
17	39	40	1	41 ± 4	2	800,000	770,857	29,143	737,820 ± 9,324	62,181	No	No
18	52	50	2	52 ± 1	1	1,000,000	961,866	38,134	936,329 ± 11,850	63,671	Sí	No
19	80	76	4	83 ± 4	3	1,500,000	1,453,032	46,968	1,551,282 ± 52,870	51,282	Sí	No
20	56	51	5	54 ± 0	2	800,000	854,348	54,348	842,344 ± 51,656	42,344	Sí	Sí
21	35	34	1	38 ± 5	3	550,000	567,484	17,484	595,126 ± 88,635	45,126	No	No

el valor actual, el valor obtenido por el modelo original, la diferencia de días entre el modelo y el valor actual, el valor generado por el ensamble con su desviación estándar y la diferencia de días entre el valor real y el estimado por el modelo propuesto.

El modelo híbrido con categorías de clasificación mejora los tiempos en 15 de los 21 proyectos, las desviaciones estándar son pequeñas ya que solo uno de los proyectos tiene un desfase de dos semanas laborales o 10 días. Respecto a las estimaciones de costo la tabla ordena los valores de manera similar, comparando aquellos obtenidos por el modelo original, su diferencia, y los valores obtenidos con el modelo híbrido.

En este caso la mejora en las estimaciones ocurre solo en 9 de los 21 proyectos comparado con el modelo original. Sin embargo es importante resaltar que de los 12 proyectos donde no mejoró la predicción, la diferencia de costo no supera las 41,000 rupias. Solo en el proyecto número cuatro la diferencia de costo rebasó esta cantidad y las desviaciones estándar en tres proyectos superan las 100,000 rupias de diferencia entre el valor real y el valor central de la estimación.

Finalmente la comparativa con el estado del arte se muestra en la Tabla 3c mostrando que la aplicación de redes neuronales artificiales da resultados competitivos comparado con las técnicas aplicadas en la literatura.

6. Conclusión y trabajo a futuro

Los resultados obtenidos por el ensamble redujeron la desviación estándar de las predicciones brindando resultados confiables para el tiempo de finalización y el costo total de los proyectos. El mejor desempeño lo tuvo el perceptrón multicapa ya que la

Tabla 5. Ventajas y desventajas de los distintos algoritmos de aprendizaje automático empleados en la literatura y en la presente propuesta de investigación.

Técnica	Ventajas	Desventajas
Cascade Correlation Neural Network	Autoorganizadas	Con facilidad caen en sobreentrenamiento
	La arquitectura crece a medida que se ejecuta el entrenamiento	el algoritmo de aprendizaje define el tamaño correcto de la red
	Rapido entrenamiento	Contiene una capa oculta
Decision Tree	Adaptabilidad al contexto del problema	Con facilidad caen en sobreentrenamiento
	Se pueden generar distintos tipos de ensambles a bajo costo	Entre más debil el estimador, mayor el numero de modelos a entrenar en un ensamble
	Facil Interpretación y corto tiempo de entrenamiento, operación en tiempo real	
Generalized Regression Neural Network	Adaptabilidad al contexto del problema	No recomendables en problemas no lineales
	Se basa en la distancia euclidiana, la capa de patrón realiza operaciones elementales de suma y producto y la capa de decisión aplica cocientes.	Arquitectura fija de cuatro capas
MultiLayer Perceptron	Modelado flexible	Altos tiempos de entrenamiento
	Autoorganizadas	Costo computacional
	Adaptabilidad a problemas no lineales	Complejas para implementar desde cero
	Tolerancia a fallas y operación en tiempo real	Dificil interpretación de la ejecución
Recurrent Neural Network	Adaptabilidad a problemas no lineales y series temporales	Formateo especial de los datos de entrada
	Modelado flexible	Bajo desempeño en conjuntos de datos pequeños
	Autoorganizadas, tolerancia a fallas y operación en tiempo real	Arquitectura compleja en relación al MLP

validación cruzada permitió evaluar la capacidad de generalización de los algoritmos, siendo favorable para MLP y dando malos resultados para la red neuronal recurrente debido a que una ventana de tiempo pequeña no tiene los suficientes elementos para realizar las estimaciones de entradas desconocidos.

La preclasificación de proyectos mejoró la precisión de la predicción y también aumentó el coeficiente de determinación. Además, el ensamble de técnicas de aprendizaje automático también contribuye a la mejora de los resultados dando estabilidad a las predicciones y reduciendo la dispersión de los datos.

El ensamble tuvo resultados favorables teniendo una precisión de la predicción de

95.58 % con coeficientes de determinación mayores a 0.98. Como lo muestran las tablas comparativas de las predicciones emitidas, la desviación estándar de las estimaciones de tiempo no superan las dos semanas laborales y la mayoría de las estimaciones de costo tienen una desviación por debajo de las 100 mil rupias.

La presente propuesta en comparación con el modelo de estimación [8], brindó resultados confiables en las estimaciones de tiempo mejorando en 15 de los 21 proyectos y reduciendo la desviación estándar a no más de una semana laboral y tres días. Sin embargo, las estimaciones de costo no presentaron una mejora significativa ya que los mejores costos se obtienen al aplicar el modelo original [8].

Al comparar los resultados del perceptrón multicapa con las técnicas presentadas en la literatura se observa una mejora en la precisión con respecto a las redes neuronales de regresión generalizada. También superó a la red de tipo Cascade Correlation al obtener un coeficiente de determinación mayor en las estimaciones de tiempo. Una ventaja de la presente propuesta de investigación con respecto a los trabajos [1,6,3] es la facilidad de modelado y adaptabilidad de las redes neuronales, así como la capacidad de emitir predicciones de tiempo de finalización y costo total utilizando el mismo modelo. En el trabajo futuro, se buscará aplicar otras técnicas de aprendizaje automático como el algoritmo KNN, árboles de decisiones y autoencoders para generar estimaciones de tiempo y costo y robustecer el ensamble agregando todas estas técnicas al modelo híbrido ya existente.

Referencias

1. Panda, A., Satapathy, S. M., Rath, S. K.: Empirical validation of neural network models for agile software effort estimation based on story points. Procedia Computer Science, vol. 57, pp. 772–781 (2015) doi: 10.1016/j.procs.2015.07.474
2. Rojas, R.: Neural networks: a systematic introduction. Springer-Verlag (1996) doi: 10.1007/978-3-642-61068-4
3. Rao, C. P., Kumar, P. S., Sree, S. R., Devi, J.: An agile effort estimation based on story points using machine learning techniques. Bhateja, V., Tavares, J., Rani, B., Prasad, V., Raju, K., (eds). In: 2nd International Conference on Computational Intelligence and Informa-tics Advances in Intelligent Systems and Computing, vol. 712, pp. 209–219 (2018) doi: 10.1007/978-981-10-8228-3_20
4. Raslan, A. T., Darwish, N. R., Hefny, H. A.: Effort estimation in agile software projects using fuzzy logic and story points. In: 50th Annual Conference on statistics, computer scien-ces, and operation research, pp. 27–30 (2015)
5. Scott, E., Pfahl, D.: Using developers' features to estimate story points. In: International Conference on Software and System Process, pp. 106–110 (2018) doi: 10.1145/3202710.3203160
6. Satapathy, S. M., Rath, S. K.: Empirical assessment of machine learning models for agile software development effort estimation using story points. Innovations in Systems and Software Engineering, vol. 13, no. 2–3, pp. 191–200. (2017) doi: 10.1007/s11334-017-0288-z
7. Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A., Menzies, T.: A deep learning model for estimating story points. IEEE Transactions on Software Engineering, vol. 45, no. 7, pp. 637–656 (2019) doi: 10.1109/TSE.2018.2792473
8. Ziauddin, Tipu, S. K., Zia, S.: An effort estimation model for agile software development. Advances in Computer Science and its Applications, vol. 2, no. 1, pp. 314—324 (2012)

9. Coelho, E., Basu, A.: Effort estimation in agile software development using story points. International Journal of Applied Information Systems (IJAIS), vol. 3, no. 7, pp. 7—10 (2012) doi: 10.5120/ijais12-450574
10. Vyas, M., Bohra, A., Lamba, C. S., Vyas, A.: A review on software cost and effort estimation techniques for agile development process. International Journal of Recent Research Aspects, vol. 5, no.1, pp. 1–5 (2018)
11. Choudhari, J., Suman, U.: Story points based effort estimation model for software maintenance. Procedia Technology, vol. 4, pp. 761–765 (2012) doi: 10.1016/j.protcy.2012.05.124
12. Digital.ai.: The 14th Annual state of agile report is here. digital.ai software Inc. (2020)
13. Popli, R. N., Chauhan: Cost and effort estimation in agile software development. In: International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 57–61 (2014) doi: 10.1109/ICROIT.2014.6798284
14. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks, vol. 2, no. 5, pp. 359–366 (1989) doi: 10.1016/0893-6080(89)90020-8
15. Fernandez-Diego, M., Mendez, E. R., Gonzalez-Ladron-De-Guevara, F., Abrahao, S., Insfran, E.: An update on effort estimation in agile software development: A systematic literature Review. IEEE Access, vol. 8, pp. 166768—166800 (2020) doi: 10.1109/ACCESS.2020.3021664
16. Garg, S., Gupta, D.: PCA based cost estimation model for agile software development projects. In: International Conference on Industrial Engineering and Operations Management (IEOM), pp. 1–7 (2015) doi: 10.1109/IEOM.2015.7228109
17. Malgonde, O., Chari, K.: An ensemble-based model for predicting agile software development effort. Empirical Software Engineering, vol. 24, no. 2, pp. 1017–1055 (2019) doi: 10.1007/s10664-018-9647-0
18. Durán, M., Juárez-Ramírez, R., Jiménez, S., Tona, C.: User story estimation based on the complexity decomposition using Bayesian networks. Programming and Computer Software, vol. 46, no. 8, pp. 569—583 (2020) doi: 10.1134/S0361768820080095
19. Gultekin, M., Kalipsiz, O.: Story point-based effort estimation model with machine learning techniques. International Journal of Software Engineering and Knowledge Engineering, vol. 30, no. 1, pp. 43–66 (2020) doi: 10.1142/S0218194020500035

ISSN 1870-4069

Medida de similitud para experiencias de juego basada en el ritmo

José A. Torres-León[1], Marco A. Moreno-Armendáriz[1], Carlos A. Duchanoy[2], Hiram Calvo[1]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Laboratorio de Ciencias Cognitivas Computacionales, México

[2] Gus Chat, México

{jtorresl2019,mam_armendariz,hcalvo}@cic.ipn.mx, carlos.duchanoy@gus.chat

Resumen. En este escrito se presenta una medida de similitud para evaluar la experiencia de juego que brinda una sección de nivel. La medida es una propuesta para evaluar matemáticamente el concepto de "ritmo" en los video juegos. Se presenta una investigación sobre medidas existentes para evaluar las experiencias de juego en el área de la generación procedimental de contenido y por qué la medida de similitud presentada es relevante. Además, se presenta un estudio realizado sobre la utilidad de la medida para al menos tres diferentes experiencias de juego.

Palabras clave: Inteligencia artificial, medidas de similitud, experiencia de juego, generación procedimental de contenido.

Rhythm Based Similarity Measure for Game Experiences

Abstract. In this paper a similarity measure for evaluating game experiences in a level chunk is presented. The measure is a proposal to mathematically evaluate the "rhythm" concept on videogames. Research on existing measures to evaluate the game experiences in the procedural content generation area is presented and why the proposed similarity measure is relevant. In addition, a utility study of the measure to at least three different game experiences are presented.

Keywords: Artificial intelligence, similarity measures, game experience, procedural content generation.

1. Introducción

La generación procedimental de contenido (GPC) es una disciplina que se concentra en la generación algorítmica de contenido, principalmente, para videojuegos. La GPC puede usarse para generar todo tipo de contenido, se ha usado para crear niveles, música, sonidos, flujos de misiones, personajes e incluso reglas de juego, ya sea en tiempo de ejecución o como herramienta de diseño [1]. El alto impacto que ha tenido esta disciplina ha generado un gran interés en la comunidad académica y ha desembocado en un considerable número de libros y estudios en las últimas dos décadas [2, 3, 4, 5, 6, 7, 8].

En general, estos métodos requieren de una evaluación cuantitativa de la calidad o las características del contenido generado, de manera que se pueda afirmar si los resultados que de ellos se obtienen son adecuados. Por ello, existen una gran variedad de formas de evaluar el contenido de los videojuegos.

La propuesta que se presenta, consiste en una nueva medida de similitud para experiencias de juego, la cual toma como base la idea del *ritmo*, que ha sido usado por los diseñadores de niveles de video juegos en la industria. Esta medida de similitud pretende ser una herramienta de apoyo para sistemas de GPC y de inteligencia artificial (IA) involucrados en las tareas de desarrollo de videojuegos.

La propuesta, surge al detectar la necesidad de los sistemas de generación automática de niveles para determinar si el contenido generado incluye una cierta experiencia de juego.

Por ello, en la siguiente sección se presentan algunos trabajos del estado del arte sobre la evaluación del contenido de los videojuegos, revisando qué aspectos se consideran en general al evaluar el contenido y cuáles son las propuestas sobre la evaluación de experiencias de juego.

2. Métodos de evaluación de contenido de videojuegos

Los primeros acercamientos a la evaluación de contenido de videojuegos se basan en la capacidad de un experto para asegurar la calidad del producto. Por ello, los primeros esfuerzos para formalizar la evaluación de contenido se centran en traducir y/o resumir el conocimiento de un experto en conjuntos de consejos y buenas prácticas para que los no expertos sean capaces de evaluar el contenido de un videojuego de manera similar a como lo haría un experto.

Algunos ejemplos de las propuestas más tradicionales se encuentran en [10, 11], que proporcionan una serie de heurísticas o consejos, que servirán para determinar la calidad del nivel y la experiencia de juego que brindan. Estas propuestas ofrecen una primera noción acerca de lo que debería considerar cuando se evalúan contenido para un videojuego. Las siguientes son dos de las recomendaciones más relevantes de [10]:

1. "El *gameplay* (la *jugabilidad*) debe ser balanceado, con múltiples caminos para ganar". Para medir este aspecto se puede contar el número de caminos diferentes en los niveles para llegar a la plataforma meta,
2. "La fatiga del jugador se minimiza al variar las actividades y el paso (ritmo) durante el juego". Para medir la variación en las actividades se puede implementar una medida de ritmo.

Por su parte, en [11] se afirma que cada género de videojuego tiene un ritmo característico de la dificultad, este ritmo o paso se ve reflejado al momento de jugar. Por ejemplo, en un juego de plataforma es común que entre un reto y otro el nivel ofrezca un espacio para descansar al jugador o que se ofrezcan dos rutas distintas, una llena de enemigos y recompensas y otra que es un camino vacío y que no presenta ni enemigos ni recompensas. Estos cambios a lo largo del nivel marcan una forma específica de interactuar con él, misma que implica subidas y bajadas en la dificultad que implica superar cada sección del nivel para el jugador, estos cambios son los que constituyen el ritmo característico. En conjunto, estas referencias dejan en claro que características como la variedad de caminos, la dificultad o el tipo de reto presente en ellos, así como el ritmo de estas características en los caminos, son útiles para definir el tipo de nivel al que se enfrenta un jugador.

2.1. Modelado de la experiencia de juego

Una de las motivaciones principales para el desarrollo de las métricas para niveles de videojuegos es que describan las diferentes experiencias de juego.

Como se indica en [12], la experiencia de juego se refiere a la relación que existe entre las reacciones del jugador al momento de interactuar con el videojuego y el contenido con el que interactúa en dicho momento, por ello se considera como una función del contenido del juego y el jugador. Por su parte, la jugabilidad del videojuego se refiere a cómo se ha diseñado el contenido para que el jugador interactúe con él.

Entonces, el contenido se diseña de modo que hay una cierta jugabilidad asociada a éste. Sin embargo, la experiencia de juego varía dependiendo de cada jugador dada una misma *jugabilidad*, por lo tanto, aunque no se puede asegurar que todos los jugadores tendrán una misma experiencia bajo una misma jugabilidad, sí existe una relación entre la *jugabilidad* y la experiencia de juego, al hacer que todos los jugadores interactúen de una misma forma con el videojuego, se espera que sus reacciones sean similares.

Para evaluar la experiencia de juego existen algunas propuestas que consideran tomar mediciones directamente del jugador durante sesiones de juego y después de ellas, tales como [13, 14, 15, 16, 17, 18, 19, 20], sin embargo, estas evaluaciones son costosas, pues algunas requieren de equipo especializado para tomar las medidas y en general, todas requieren de jugadores que deben ser remunerados por su participación en el proyecto y que inevitablemente dan una evaluación sesgada debida a sus propias habilidades y preferencias de juego.

Sin embargo, también existen trabajos como [21, 22] cuyos autores hacen un análisis del contenido de juego presente en una porción de nivel y lo relacionan con la mecánica o reto que presenta dicha porción de nivel para el jugador.

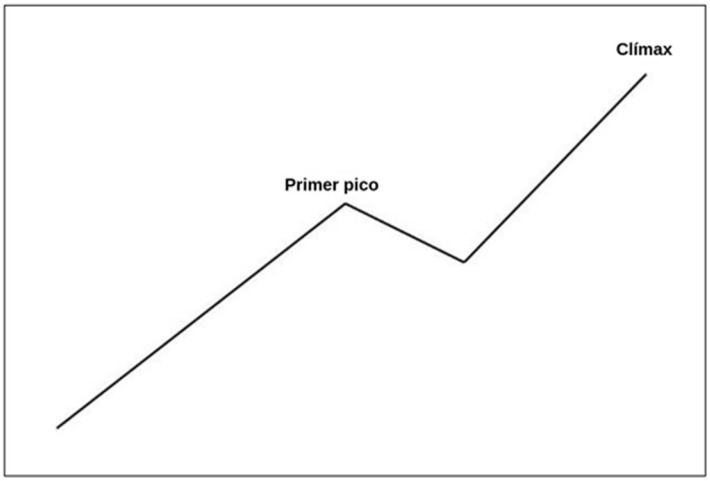


Fig. 1. Ilustración de una característica de nivel de videojuegos con dos fluctuaciones.

Aunque estos trabajos dejan de lado aspectos importantes como la dificultad o el ritmo que evalúan los expertos para determinar si una experiencia de juego está presente en cierto contenido de un videojuego.

3. Medida de similitud de experiencias de juego propuesta

Con base en los conceptos presentados en la sección 2, se ideó una nueva evaluación para experiencias de juego, que toma en consideración el ritmo de las características de un nivel, en otras palabras, esta medida toma en consideración el cambio de una medida a lo largo de un nivel del videojuego, permitiendo analizar el nivel en términos de cómo éste fue diseñado y qué jugabilidad ofrece a lo largo de la ruta. Por ello, se hace la suposición de que en cada experiencia de juego debe existir un ritmo específico en las características de las rebanadas del nivel, lo cual implica que a partir del ritmo de esas características es posible afirmar que una rebanada de nivel ofrece una experiencia de juego.

El ritmo característico de una característica de nivel se ilustra en la Fig. 1, en la cual se observan dos picos o fluctuaciones en la medida de la característica.

La medida de similitud propuesta evalúa el comportamiento de las características medidas del nivel, a partir del uso de métricas que midan características de interés (como la dificultad, la suma de puntos que puede obtener el jugador, la distancia entre dos objetos del nivel, etc.), evaluando estas mediciones a lo largo de un camino que lleva de inicio a fin del nivel. Lo que se busca con esta evaluación es asegurar que el ritmo y el rango de valores (valor máximo y mínimo), de las características a evaluar, son cercanos a un parámetro conocido de una experiencia de juego.

Es decir, nuestra evaluación del cumplimiento de la experiencia de juego se basa en la comparación del nivel medido/evaluado con un nivel ideal/prototipo, donde la

comparación se hace a través del ritmo y el rango de valores de las características de interés.

Por lo tanto, nuestra medida de similitud se divide en dos partes, la "exactitud de ritmo" y la "exactitud de rango de valores", se definen como "exactitud" debido a que entre más cercanos sean los valores medidos con los valores objetivo, mayor será la certeza de que el nivel evaluado brindará la experiencia de juego deseada. Para fines prácticos, decidimos establecer los valores de similitud en el rango de [0,1], donde '0' es lo más distinto posible y '1' es lo más similar posible.

3.1. Exactitud de ritmo

Para esta parte de la medida de similitud, se utilizó una función con forma de campana, cuyo valor se calcula con base en la Ecuación (1). Para esta función, 'x_i' es el ritmo de la característica i-ésima, 'σ' es la varianza deseada del ritmo, que sirve para ampliar o reducir el ancho de la campana, haciendo la evaluación del ritmo más relajada (más amplia) o más estricta (más reducida), 'μ' es el ritmo objetivo, en la ecuación es la media, indicando que la campana tendrá su centro y, por lo tanto, valor máximo, en este punto:

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

Esta función es normalizada para ajustar sus valores al rango [0,1]. Por lo tanto, la primera función de exactitud para la medida de similitud se muestra en la Ecuación (2):

$$a_r = \frac{f(x_i)}{f(\mu)}. \tag{2}$$

3.2. Exactitud de rango de valores

Para esta parte de la medida de similitud, la meta es evaluar qué tan cercano es el rango de valores medidos contra el rango de valores objetivo. Para este propósito, se proponen a su vez dos diferentes comparaciones entre los rangos. Para definir qué tan iguales son dos rangos, se propone comparar la amplitud del rango y su centro, entre más cercanos sean dos rangos en cuanto a sus centros y sus amplitudes, más semejantes son.

Estas ideas se ilustran en la Fig. 2, donde se muestra una curva de una característica de un nivel de videojuego, cuyos valores son la medida de dicha característica a lo largo de los puntos de un camino de inicio a fin del nivel. En esta Figura se representa el rango objetivo (en azul) en el que esa característica debería estar de acuerdo a la experiencia de juego objetivo y el rango real medido (en verde) de esa característica.

Estos rangos se definen a través de los valores máximo y mínimo, marcados con letras azules.

Sean γ el valor máximo medido, δ el valor mínimo medido, λ el valor máximo objetivo, τ el valor mínimo objetivo, entonces, la primera de estas comparaciones se

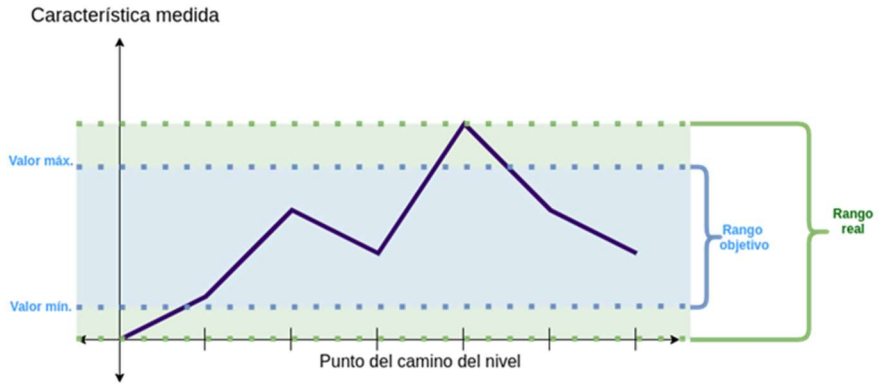


Fig. 2: Ilustración de la exactitud de rango.

muestra en la Ecuación (3), la similitud de amplitud (S_a), que está dada por 1 menos la distancia normalizada entre las amplitudes:

$$S_a = 1 - \frac{|(\gamma - \delta) - (\lambda - \tau)|}{\max(\gamma, \lambda)}. \tag{3}$$

La segunda comparación se muestra en la Ecuación (4), la similitud de centros (S_c), que está dada por 1 menos la distancia normalizada entre los centros:

$$S_c = 1 - \frac{\left(\frac{1}{2}(\gamma - \delta) + \delta\right) - \left(\frac{1}{2}(\lambda - \tau) + \tau\right)}{\max(\gamma, \lambda)}. \tag{4}$$

Estas ecuaciones son válidas si y sólo si $\max(\gamma, \lambda) \neq 0$, de lo contrario, deben utilizarse las Ecuaciones (5) y (6):

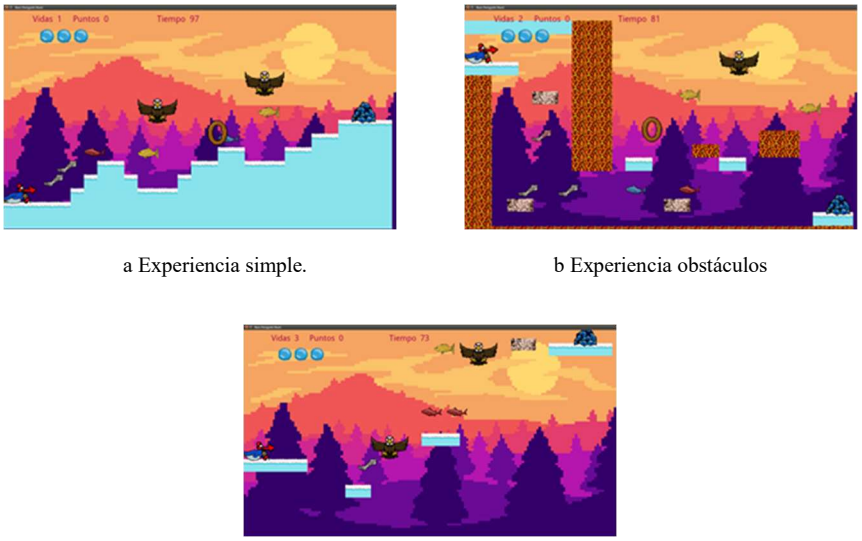
$$S_a = 1 - |(\gamma - \delta) - (\lambda - \tau)|, \tag{5}$$

$$S_c = 1 - \left[\left(\frac{1}{2}(\gamma - \delta) + \delta\right) - \left(\frac{1}{2}(\lambda - \tau) + \tau\right)\right]. \tag{6}$$

Finalmente, la exactitud de rango de valores se muestra en la Ecuación (7):

$$a_v = \frac{1}{2}S_a - \frac{1}{2}S_c. \tag{7}$$

Entonces, la medida de similitud *S* para una característica es una suma ponderada de las exactitudes a_r y a_v, tal y como se muestra en la Ecuación (8), donde $\alpha + \beta = 1$. Los pesos α y β definen qué exactitud es más relevante, la del ritmo o la del rango de valores:



a Experiencia simple.

b Experiencia obstáculos

c Experiencia Saltos

Fig. 3. Experiencias de juego modeladas en Pingu Run.

$$S = \alpha \cdot a_r + \beta \cdot a_v \,. \tag{8}$$

Dada una experiencia de juego, debe calcularse la similitud de acuerdo a la Ecuación (8) para cada característica de interés y finalmente hacer una suma ponderada de cada una de ellas. La similitud hacia una experiencia de juego se define en la Ecuación (9), donde w_i es el peso de la característica i-ésima y S_i es la función de similitud de esa misma característica, donde $\sum_i w_i = 1$.:

$$R_{GX} = \sum_i w_i \cdot R_i \,. \tag{9}$$

4. Experimentos y resultados

4.1. Niveles prototipo

Para poner a prueba nuestra medida de similitud en un caso de estudio concreto se modelaron tres diferentes experiencias de juego, que se presentan en la Figura 3, para el videojuego "Pingu Run"' disponible en [23]. Para tomar las medidas de las características de los niveles usamos el método de evaluación de niveles de videojuegos propuesto en [24], donde se genera un grafo del nivel. En este grafo, cada nodo es un elemento del nivel (piso, enemigos, bonificaciones, etc.) y cada arista entre nodos es una posible transición entre esos bloques.

Este grafo se anota con ocho métricas diferentes: riesgo, obstrucción, distancia, precisión, recompensa de nivel, recompensa de bonus, motivación de nivel y motivación de bonus.

Para obtener una medida del ritmo y de los valores máximo y mínimo, se calculó la ruta de "recompensa de nivel máxima" (la ruta con suma máxima en la medida de recompensa de nivel) con ayuda del algoritmo de Dijkstra para cálculo de rutas óptimas en grafos. Bajo este método de evaluación, el ritmo de una característica se mide contando el número de fluctuaciones que tiene su curva a lo largo de la ruta evaluada. Los valores máximo y mínimo son la medida máxima y la medida mínima de esa misma característica, respectivamente.

Para poder usar la medida de similitud propuesta, es necesario definir los valores objetivo para las ecuaciones presentadas ($\mu, \sigma, \lambda, \tau, \alpha, \beta, w$), puesto que a partir de ellos es que se cálcula la similitud contra una experiencia de juego. Como resultado de una definición correcta de estos valores, al evaluar la presencia de una experiencia de juego en una rebanada de nivel, entonces para las rebanadas que sí contengan esa experiencia de juego se debe obtener un valor cercano a 1 y un valor cercano a cero en caso de que esa experiencia de juego no esté presente en la rebanada evaluada.

Para definir estos valores objetivo, se crearon tres niveles para cada una de las experiencias de juego (simple, obstáculos y saltos). Se aplicó la metodología de evaluación seleccionada, obteniendo un grafo por cada nivel evaluado. De esos grafos, se analizó el ritmo y rango de valores a lo largo de la ruta de interés (recompensa de nivel máxima), para las características seleccionadas (de las ocho posibles se seleccionaron seis, la recompensa de nivel, motivación de nivel, riesgo, motivación de bonus, recompensa de bonus y distancia, dado que en [24] demostraron ser las que mayor relación tenían con la etiqueta de experiencia de juego).

Después de proponer diferentes conjuntos de valores objetivo, se encontraron aquellos que se muestran en la Tabla 1. En general, los valores de ritmo (μ), variación del ritmo (σ), valor máximo (λ) y valor mínimo (τ) fueron definidos con base en el valor más frecuente o en el promedio de las medidas observadas en los niveles evaluados.

Por otro lado, los pesos de la exactitud de ritmo (α), exactitud de rango de valores (β) y el peso (w) fueron propuestos con base en la importancia Gini de cada medida (la importancia Gini es una medida de relevancia que se usa dentro del método de selección de características ExtraTreesClassifier, el cual se usa como parte del método de evaluación de niveles seleccionado).

Si una característica tiene mayor importancia Gini que otra, entonces su peso (w) debe ser mayor en la medida de similitud. Para determinar esta relación, sumamos las importancias Gini de cada característica seleccionada y calculamos la aportación proporcional de cada una de ellas para obtener cada peso (w_i).

Si el ritmo de una característica tiene una mayor importancia Gini que su valor máximo o su valor mínimo, entonces la ponderación de la exactitud de ritmo (α) debe ser mayor que la de la exactitud de rango de valores (β) y vice versa.

Para ello, se experimentó con varios valores de α y β, considerando las importancias Gini y buscando una mejor evaluación en la medida de similitud en los niveles ejemplo.

Cabe destacar que, como parte de esta búsqueda de los valores objetivo, se identificó que la *motivación de bonus*, en el caso de la experiencia de juego *simple* causaba una mala identificación de las experiencias de juego, reduciendo la similitud calculada para los niveles que sí correspondían a dicha experiencia; por lo tanto, para esa experiencia de juego no se consideró a la *motivación de bonus* como una característica a evaluar dentro de la medida de similitud.

4.1. Evaluación usando la medida de similitud propuesta

Con base en los valores objetivo definidos en la Tabla 1 los niveles creados obtuvieron las evaluaciones mostradas en la Tabla 2, la cual, muestra que cada nivel tienen un valor de similitud cercano a 1 cuando se evalúa con la medida de similitud ajustada a su correspondiente experiencia de juego y una evaluación cercana a 0 para las medidas de similitud ajustadas a otras experiencias de juego, por lo tanto, se cumple que la similitud es alta cuando se ajustan los valores objetivo de manera adecuada para cada experiencia de juego.

Esto le da la utilidad a la medida de similitud de ser usada en sistemas de inteligencia artificial dando la capacidad de determinar si una experiencia de juego especificada está dentro de una porción de nivel o no. Algunos ejemplos de dichos sistemas son: asistentes de diseño (le indicarían un estimado de cumplimiento de experiencia de juego al diseñador), generadores de secciones de nivel de una experiencia de juego dada (ayudarían al sistema de generación para indicar cuando una experiencia de juego está presente en el contenido generado), entre otros.

5. Conclusiones

En este artículo presentamos una medida de similitud para experiencias de juego capaz de adecuarse al menos a tres diferentes experiencias de juego. Nuestra medida de similitud es independiente de las características a evaluar y las correspondientes métricas a usar para medirlas, sin embargo, requiere que se obtenga un ritmo (o un dato similar) así como los valores máximo y mínimo de las características, además de algún criterio que ayude a determinar el conjunto de valores objetivo (la variación del ritmo, los tres aspectos ya mencionados y sus correspondientes ponderaciones).

También es una medida de similitud independiente del jugador, pues no requiere ningún dato proveniente de él para estimar la experiencia de juego presente en una rebanada de nivel de videojuego.

Finalmente, sería interesante implementar esta medida de similitud para otros video juegos, así como para otras experiencias de juego. Al mostrar la utilidad de esta medida de similitud para determinar que otras experiencias de juego están presentes, en el caso de “Pingu Run” o en otro video juego, podría corroborarse la relevancia de la propuesta para el área de la generación procedimental de contenido.

Tabla 1. Valores objetivo para cada métrica en cada experiencia de juego.

Experiencia de juego	Métrica	μ	σ	λ	τ	α	β	w
Simple	recompensa de nivel	2	3	916	0	0.95	0.05	0.29
	motivación de nivel	0	3	1406	0	0.75	0.25	0.26
	riesgo	2	0.001	491	0	0.65	0.35	0.22
	recompensa de bonus	2	2	265	0	0.85	0.15	0.15
	distancia	4	4	5	1	0	1	0.08
Obstáculos	recompensa de nivel	4	3	800	0	0.95	0.05	0.25
	motivación de nivel	2	3	1384	-70	0.75	0.25	0.22
	riesgo	6	4	243	0	0.65	0.35	0.19
	motivación de bonus	2	0.001	160	-184	0.35	0.65	0.13
	recompensa de bonus	4	4	90	0	0.85	0.15	0.11
	distancia	4	4	11	1	0	1	0.1
Saltos	recompensa de nivel	1	0.001	1000	0	0.95	0.05	0.25
	motivación de nivel	1	0.001	1805	-56	0.75	0.25	0.22
	riesgo	5	2	155	0	0.65	0.35	0.19
	motivación de bonus	1	2	530	-68	0.35	0.65	0.13
	recompensa de bonus	5	3	350	0	0.85	0.15	0.11
	distancia	4	4	8	1	0	1	0.1

6. Contribuciones por autor

Conceptualización y metodología, M.A.M.-A., J.A.T.-L.C.A.D. y H.C.; investigación y recursos, M.A.M.-A., H.C., C.A.D y J.A.T.-L.; software, visualización y curación de datos, J.A.T.-L.; validación, H.C. y M.A.M.-A.; análisis formal, M.A.M.-

Tabla 2. Similitud medida en los niveles ejemplo usando los valores objetivo de la Tabla 1.

Nivel	Medida de similitud		
	Obstáculos	**Simple**	**Saltos**
obstáculos 1	**0.748515**	0.216647	0.169015
obstáculos 2	**0.741509**	0.33168	0.192525
obstáculos 3	**0.695833**	0.253077	0.285230
simple 1	0.083789	**0.843283**	0.08732
simple 2	0.197562	**0.765775**	0.178005
simple 3	0.204252	**0.829947**	0.183384
saltos 1	0.189115	0.255675	**0.975981**
saltos 2	0.158059	0.268668	**0.702896**
saltos 3	0.181520	0.261360	**0.970456**

A.; escritura - preparación del manuscrito original, J.A.T.-L., C.A.D.; escritura – revisión y edición M.A.M.-A. y H.C.; supervisión, administración del proyecto y adquisición de financiamiento, M.A.M.-A. Todos los autores han leído y están de acuerdo con la versión a publicar del manuscrito.

7. Financiamiento

Este trabajo fue posible gracias al apoyo del gobierno mexicano (CONACyT)y el Instituto Politécnico Nacional a través de las becas de investigación SIP-2083, SIP-20220553, SIP-20221064 y SIP 20220533 de la SIP-IPN; IPN-COFAA e IPN-EDI.

8. Declaración de la junta de revisión institucional:

No aplicable.

9. Declaración de consentimiento informado:

No aplicable.

10. Declaración de disponibilidad de datos

Los autores están comprometidos a proveer acceso a toda la información necesaria para que los lectores puedan reproducir completamente los resultados obtenidos en este trabajo. Los conjuntos de datos usados tienen disponibilidad pública.

11. Conflictos de intereses

Los autores declaran que no hay conflicto de intereses.

Referencias

1. Togelius, J., Kastbjerg, E., Schedl, D., Yannakakis, G.N.: What is procedural content generation. Mario Borderline. In: PCGames '11: Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, no. 3, pp. 1–6 (2011) doi: 10.1145/2000919.2000922
2. De Kegel, B., Haahr, M.: Procedural puzzle generation: A survey. IEEE Trans. Games, vol. 12, no. 1, pp. 21–40 (2020) doi: 10.1109/TG.2019.2917792
3. Risi, S., Togelius, J.: Increasing generality in machine learning through procedural content generation. Nat Mach Intell, vol. 2, pp. 428–436 (2020) doi: 10.1038/s42256-020-0208-z
4. Shaker, N., Togelius, J., Nelson, M. J.: Procedural content generation in games. Springer. Berlin/Heidelberg, Germany (2016)
5. Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., ISaksen, A., Nealen, A., Togelius, J.: Procedural content generation via machine learning (PCGML). EEE Transactions on Games, vol. 10, no. 3, pp. 257–270 (2018) doi: 10.1109/TG.2018.2846639
6. Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G.N., Togelius, J.: Deep learning for procedural content generation. Neural Comput & Applic vol. 33, pp. 19–37 (2021) doi: 10.1007/s00521-020-05383-8
7. Togelius, J., Yannakakis, G. N., Stanley, K. O., Browne, C.: Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games, vol. 3, no. 3, pp. 172–186 (2011) doi: 10.1109/TCIAIG.2011.2148116
8. Yannakakis, G. N., Togelius, J.: Artificial Intelligence and Games. Springer. Berlin/Heidelberg, Germany (2018). Available online: gameaibook.org (accessed on 8 April 2022)
9. Alvarez, A., Dahlskog, S., Font, J., Togelius, J.: Empowering quality diversity in dungeon design with interactive constrained map-elites. In: IEEE Conference on Games (CoG), pp. 1–8 (2019) doi: 10.1109/CIG.2019.8848022
10. Desurvire, H., Caplan, M., Toth, J. A.: Using heuristics to evaluate the playability of games. CHI EA '04: CHI '04 Extended Abstracts on Human Factors in Computing Systems, pp. 1509–1512 (2004) doi:10.1145/985921.986102
11. Feil. J., Scattergood, M.: Beginning game level design. Game development series. Thomson Course Technology, (2005)
12. Nacke, L., Drachen, A., Kuikkaniemi, K., Niesenhaus, J., Korhonen, H. J., Hoogen, W. M., De Kort, Y. A.: Playability and player experience research. In: Proceedings of Digra´09: Breaking new ground: Innovation in games, play, practice and theory (2009)
13. Gonzalez-Sánchez, J. L., Gutierrez-Vela, F. L., Montero-Simarro, F., Padilla-Zea, N.: Playability: analysing user experience in video games. Behaviour & Information Technology, vol. 31, no. 10, pp. 1033–1054 (2012) doi: 10.1080/0144929X.2012.710648
14. Shaker, Noor, et al.: Fusing visual and behavioral cues for modeling user experience in games. IEEE transactions on cybernetics, vol. 43, no 6, pp. 1519–1531 (2013) doi: 10.1109/TCYB.2013.2271738
15. IJsselsteijn, W. A., De Kort, Y. A., Poels, K.: The game experience questionnaire. Technische Universiteit Eindhoven. (2013)

16. Calvillo-Gamez, E. H.: On the core elements of the experience of playing video games. Doctoral dissertation, UCL (University College London) (2009)
17. Pedersen, C., Togelius, J., Yannakakis, G. N.: Modeling player experience in Super Mario Bros. In: IEEE Symposium on Computational Intelligence and Games, pp. 132–139 (2009) doi: 10.1109/CIG.2009.5286482
18. Marczak, R., van Vught, J., Schott, G., Nacke, L. E.: Feedback-based gameplay metrics: measuring player experience via automatic visual analysis. In: Proceedings of the 8th Australasian Conference on Interactive Entertainment: no. 6, pp. 1–10 (2012) doi: 10.1145/2336727.2336733
19. Mirza-Babaei, P., Nacke, L. E., Gregory, J., Collins, N., Fitzpatrick, G.: How does it play better? exploring user testing and biometric storyboards in games user research. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1499 (2013) doi:10.1145/2470654.2466200
20. Cowley, B., Charles, D.: Behavlets: A method for practical player modelling using psychology-based player traits and domain specific features. User Modeling and User-Adapted Interaction, vol. 26, no. 2, pp. 257–306 (2016)
21. Dahlskog, S., Togelius, J.: Patterns as objectives for level generation. Design Patterns in Games (DPG), Chania, Crete, Greece (2013)
22. Khalifa, A., Green, M. C., Barros, G., Togelius, J.: Intentional computational level design. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 796–803 (2019) doi: 10.1145/3321707.3321849
23. Torres-Léon, J. A.: Pingu Run GitHub Repository. Disponible en: https://github.com/JAlbertoTorres/Pingu-run
24. Torres-Léon, J. A.: Generación automática de niveles de videojuegos. Tesis de maestría. Centro de Investigación en Computación, Instituto Politécnico Nacional pp. 94–114 (2021)

ISSN 1870-4069

Super-resolución de imágenes multiespectrales e híperespectrales usando DWT con SWT y representación dispersa

Yeredith G. Mora-Martínez, Beatriz P. García-Salgado,
Clara Cruz-Ramos, Osvaldo López-García,
Volodymyr Ponomaryov, Rogelio Reyes-Reyes,

Instituto Politécnico Nacional,
México

{ymoram1500,bgarcias1404}@alumno.ipn.mx
{ccruzra,olopezg,vponomar,rreyesre}@ipn.mx

Resumen. Las imágenes multiespectrales (MS) e híperespectrales (HS) muestran representaciones en diferentes longitudes de onda de una misma escena que se utilizan una amplia gama de aplicaciones, como la agricultura, medicina, astronomía y vigilancia, entre otros. Sin embargo, este tipo de imágenes suelen sufrir degradaciones provocadas por la baja resolución espacial del sensor que las registra. En este artículo se propone un método de superresolución de imágenes MS e HS para escenas urbanas a partir de una imagen de baja resolución (LR). Este método crea diccionarios de una base de datos de escenas naturales mediante k-Singular Value Decomposition (k-SVD) usando las sub-bandas de detalle de la Transformada Discreta Wavelet (DWT). Después, los diccionarios son empleados para reconstruir la imagen de entrada. Posteriormente, se realiza una interpolación de las sub-bandas de detalle de la DWT y la Transformada Estacionaria Wavelet (SWT) de la imagen de entrada junto con una reconstrucción de sus bordes. Finalmente, se utiliza la imagen reconstruida a partir de los diccionarios como la sub-banda de aproximación para el cálculo de la Transformada Discreta Wavelet Inversa (IDWT). El método propuesto se comparó con otros métodos similares del estado del arte en términos de Relación Media Señal-Ruido Pico Promedio (MPSNR), Índice de Similitud Estructural Promedio (MSSIM), Índice de Preservación de Bordes (EPI) y Mapeo de Ángulo Espectral (SAM) para el caso de las imágenes HS, demostrando una competitiva preservación de bordes al aumentar la resolución de las imágenes de prueba.

Palabras clave: Imágenes multiespectrales, imágenes híperespectrales, teledetección, superresolución, representación dispersa, DWT, SWT.

Super-Resolution of Multispectral Images and Hyperspectral Using DWT with SWT and Scattered Representation

Abstract. Multispectral (MS) and hyperspectral (HS) images show different wavelength representations of a single scene employed in various applications,

such as agriculture, medicine, astronomy and surveillance, among others. Nevertheless, these kinds of images are prone to degenerate due to the sensor's low resolution that acquires them. This article proposes a superresolution method for urban MS and HS images using a low-resolution image (LR). First, the method creates dictionaries from a natural scenes database by k-Singular Value Decomposition (k-SVD) using the detail sub-bands of the Discrete Wavelet Transform (DWT). Then, the dictionaries are employed to reconstruct the input image. Subsequently, an interpolation of DWT and Stationary Wavelet Transform (SWT) sub-bands of the input image is performed in combination with edges' reconstruction. Finally, the image reconstructed by dictionaries is utilized as the approximation sub-band in the Inverse Wavelet Transform (IDWT) computation. The proposed method was compared with other similar state-of-the-art methods according to Mean Peak Signal-To-Noise Ratio (MPSNR), Mean Structural Similarity Index (MSSIM), Edge Preservation Index (EPI) and Spectral Angle Mapper (SAM) for the case of HS images, demonstrating competitive edge preservation while increasing the test images' resolution.

Keywords: Remote sensing, multispectral images, hyperspectral images, superresolution, sparse representation, DWT, SWT.

1. Introducción

Los sensores de imágenes HS recopilan alrededor de 200 imágenes con un rango espectral que abarca las regiones de longitud de onda visible e infrarroja (400-2500 nm). Por otra parte, las imágenes MS son obtenidas con menor número de bandas abordando la longitud de onda visible y cercano infrarrojo. A pesar de las imágenes HS logran una alta resolución espectral, su resolución espacial es limitada.

Esto es debido a que los sistemas de teledetección necesitan varias exposiciones para la adquisición simultánea de varias bandas para que se garantice una relación señal/ruido (SNR) suficiente, lo que resulta en el sacrificio de la resolución espacial. Mientras que los sensores de imágenes MS capturan imágenes RGB con una resolución varias veces mayor que una imagen híper-espectral y con mayor SNR con el mismo tiempo de exposición.

Sin embargo, el registro de numerosas bandas espectrales de una misma escena con una alta resolución espacial es importante para las aplicaciones de teledetección [2] como la segmentación, seguimiento y reconocimiento de objetos [21,17,22]. En este artículo se presenta un método de superresolución mediante el uso de diccionarios entrenados con k-SVD [1], para reconstruir una imagen MS e HS a partir de imágenes RGB de alta resolución (HR). Las principales contribuciones de este trabajo son las siguientes:

Primero, se propone un algoritmo de multi-dicconarios donde su salida se utiliza como una nueva sub-banda de aproximación (LL) que puede ser utilizada tanto para imágenes HS como MS. Adicionalmente, se elabora una fusión de sub-bandas de detalle entre la DWT y SWT para mantener la información de alta frecuencia (HF) y mejorar la calidad de los bordes de la imagen. De este modo, se combinan dos enfoques utilizados en el estado del arte: aprendizaje de diccionarios y uso de transformaciones de dominio.

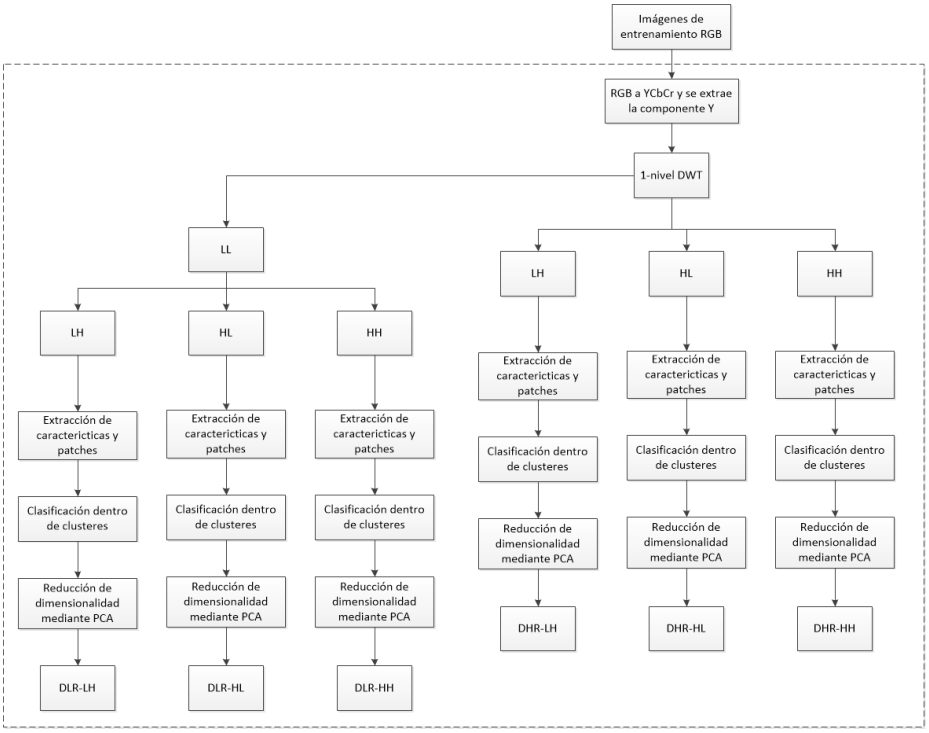


Fig. 1. Aprendizaje de sub-diccionarios en el dominio wavelet.

Para presentar el método propuesto, este documento está organizado de la siguiente forma: los trabajos relacionados con los algoritmos utilizados se presentan en la sección 2. Después, en la sección 3, se describen los pasos del método propuesto. En la sección 4 se establecen las condiciones de los experimentos y discusiones de los resultados. Finalmente, se presentan las conclusiones.

2. Trabajos relacionados

La superresolución puede abordarse de las siguientes formas: métodos basados en interpolación, métodos basados en reconstrucción y métodos basados en aprendizaje. Los métodos basados en interpolación [9,14,18,20], como la interpolación por el vecino más cercano (INN por sus siglas en ingles), interpolación bilineal e interpolación bicúbica, no ocupan muchos recursos computacionales, sin embargo, tienden a perder detalles en la imagen y dejar un efecto mosaico en ella.

Los métodos basados en reconstrucción [8,11,16], por lo general, asumen algunos modelos degradados y resuelven problemas inversos en la obtención de imágenes de HR. Este tipo de método funciona bien cuando el factor de ampliación es pequeño, pero su rendimiento requiere que la versión suavizada y muestreada de la imagen HR sea cercana a la imagen LR. Por último, los métodos basados en el aprendizaje [2,6,10,24] asumen que los detalles de alta frecuencia (HF) perdidos en la imagen LR pueden recuperarse aprendiendo las relaciones entre los parches de imágenes LR-HR.

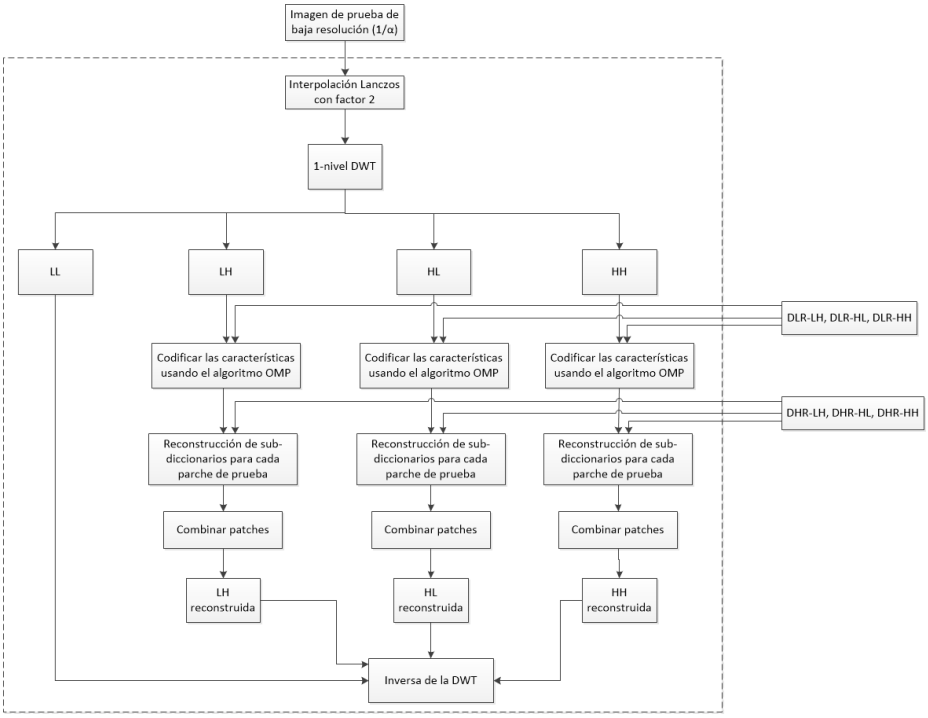


Fig. 2. Reconstrucción de la imagen.

Algunos ejemplos de este tipo de algoritmos se presentan a continuación. Alvarez-Ramos et al. [2] formulan el sistema Super-Resolution Via Wavelet Feature Extraction and Sparse Representation (SR-WAFE-SR) basado en la extracción de características wavelet y representación dispersa, donde utilizan un análisis de componentes principales (PCA) en cada sub-banda de detalle, reduciendo la dimensión de la información para obtener mejores resultados en el proceso de aprendizaje de diccionarios.

Wu et al. [24] proponen el algoritmo Wavelet Domain Multidictionary Learning (WDML-SR) que realiza un aprendizaje multi-diccionario de las sub-bandas wavelet de una imagen de baja resolución mediante un PCA. De igual forma aplican una retroproyección iterativa (IBP) y la autosimilitud no local (NLM) para evitar perder los detalles en la superresolución. Por último, Dong et al. [10] plantean el método Non-Negative Structures Sparse Representation (NSSR) donde realiza la superresolución de imágenes LR a partir de un diccionario de imágenes y una imagen estimada RGB tomada de la misma escena.

Existen otros métodos que implementan transformadas en el dominio de la frecuencia para mejorar la reconstrucción de la imagen. Chavez-Roman et al. [7] proponen el método Super Resolution Wavelet Domain Interpolation Edge Extraction Sparse Representation (SR-WDIEE-SR) donde se interpolan las sub-bandas HF de la imagen en el dominio wavelet, junto con una representación dispersa para lograr una imagen con bordes más nítidos.

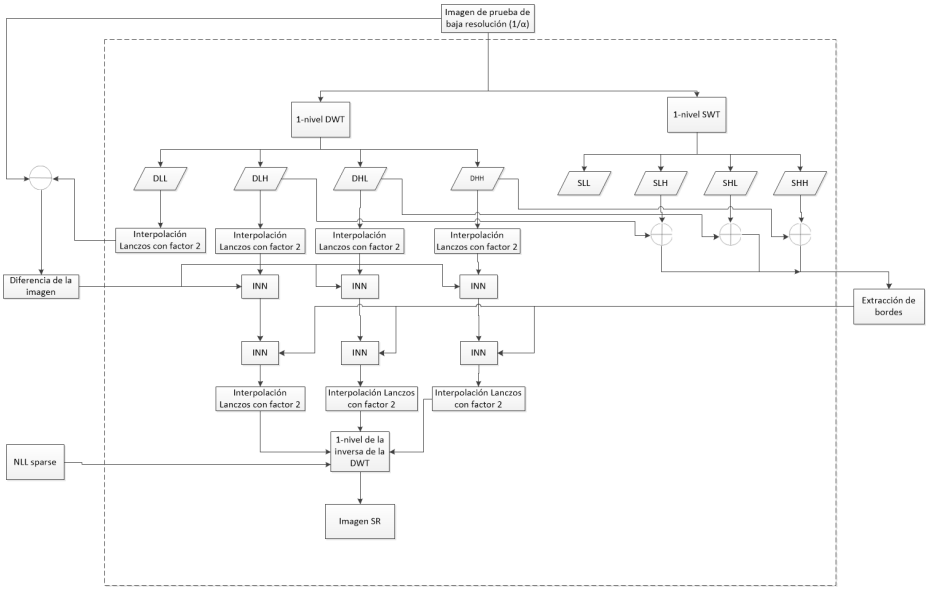


Fig. 3. Mejoramiento de bordes y obtención de imagen SR.

Por su parte, Feng et al. [11] plantean una detección de características de bordes basada en la descomposición y reconstrucción mediante la DWT tomando sólo las sub-bandas de detalle junto con una estimación de distribución de energía.

Estos enfoques mejoran la calidad visual de la imagen, sin embargo, no brindan resultados comparables con los métodos basados en aprendizaje [2,6,10,24]. Considerando esto, el método propuesto se basa en el aprendizaje de diccionarios junto con transformaciones en el dominio wavelet.

Su objetivo consta de recuperar los detalles de la imagen reconstruyendo la información sobre las sub-bandas con un diccionario y un algoritmo de preservación de bordes por medio de las transformaciones.

3. Método propuesto

En esta sección, se describe el proceso del algoritmo propuesto; y sus respectivos diagramas. El algoritmo consta de una fase de entrenamiento, una fase de reconstrucción y una etapa de mejora de bordes.

3.1. Aprendizaje de sub-diccionarios en el dominio wavelet

El aprendizaje de diccionarios, como se muestra en la Fig. 1 consta de encontrar diccionarios de las sub-bandas wavelet por medio de sub-bandas $y^{(i)}, i \in \{LH, HL, HH\}$ de imágenes HR, donde LL es la sub-banda de aproximación y LH, HL, junto con HH las de detalle. Primero, se convierten todas las imágenes de entrenamiento al espacio de color $YCbCr$ y se toma el canal de luminancia Y al ser la componente más sensible al sistema visual humano.

Tabla 1. Comparación de resultados cuantitativos de Pavia Center, Pavia University y Washington DC Mall.

Imagen		INN (1/4)	Bilinear (1/4)	Bicubic (1/4)	SR-WDIEE-SR [8] **(1/4)**	WDML-SR [24] **(1/4)**	NSSR [10] **(1/4)**	Propuesto **(1/4)**
Pavia Center	MPSNR	27.05	27.72	28.26	27.4	28.45	**32.57**	30.03
	MSSIM	0.92	0.91	0.93	0.91	0.94	**0.98**	0.97
	SAM	10.3	9.52	8.96	9.34	8.73	**7.12**	7.21
	EPI	0.06	0.19	0.22	0.19	0.22	**0.5**	0.41
	RMSE	11.61	10.73	10.07	10.87	9.63	**5.99**	8.13
Pavia University	MPSNR	26.25	27.04	27.41	27.91	28.25	**30.33**	30.26
	MSSIM	0.9	0.89	0.91	0.9	0.93	0.97	**0.98**
	SAM	8.18	7.45	7.13	6.11	6.47	5.36	**5.09**
	EPI	0.09	0.24	0.26	0.26	0. 26	0.51	**0.56**
	RMSE	12.56	11.48	11.01	10.25	9.63	**7.76**	7.85
Washington DC Mall	MPSNR	43.08	43.64	44.04	42.98	44.27	42.6	**45.62**
	MSSIM	0.96	0.96	0.96	0.95	0.97	0.94	**0.98**
	SAM	3.53	3.53	3.37	3.37	3.39	3.97	**2.72**
	EPI	0.08	0.22	0.24	0.23	0.24	0.23	**0.37**
	RMSE	4.1	3.84	3.67	1.8	1.55	1.89	**1.33**

Luego, se aplica el primer nivel de la DWT en 2D a las imágenes para separarlas en sus sub-bandas obteniendo el conjunto $y^{(h)}, h \in \{LH, HL, HH\}$ como en [7,11,24] donde los autores probaron diferentes niveles de descomposición de la DWT siendo el primer nivel de descomposición el que mejor funcionó.

Seguidamente, a la representación LL se le aplica de nuevo la DWT para formar el conjunto $y^{(l)}, l \in \{LH, HL, HH\}$. El entrenamiento de los diccionarios consta de extraer parches superpuestos locales, con tamaño de $\sqrt{n} \times \sqrt{n}$ donde $n = 25$, de las sub-bandas de ambos conjuntos por separado. Estos parches se concatenan en un vector para cada banda definido como $p^{(j)}, j \in \{LH, HL, HH\}$.

Siguiendo la idea de Dong et al. [10], se clasifican los parches de entrenamiento en k-clústeres para cada sub-banda de los conjuntos $y^{(h)}$ y $y^{(l)}$. Posteriormente, se aplica el algoritmo k-SVD con el fin de construir un diccionario sobre-completo utilizando la siguiente función:

$$\min_{D^{(j)}, \{\gamma^{(j)}\}} \left\| p^{(j)} - D^{(j)}\gamma^{(j)} \right\|^2 \text{ donde } \left\| \gamma^{(j)} \right\|_0 \leq L. \tag{1}$$

Siendo L el nivel máximo de dispersión y $\gamma^{(j)}$ un vector de coeficientes de representación dispersa de modo que:

$$p^{(j)} = D^{(j)}\gamma^{(j)T} \left(\gamma^{(j)}\gamma^{(j)T} \right)^{-1}. \tag{2}$$

Finalmente, se aplica un análisis de componentes principales (PCA) [19], como método de reducción de dimensión a los parches extraídos para cada sub-banda.

3.2. Reconstrucción de la imagen

En la fase de reconstrucción se amplían las imágenes MS y HS, las cuales se consideran que originalmente están sub-muestreadas con un factor de $(1/\alpha)$, siendo α igual a 4.

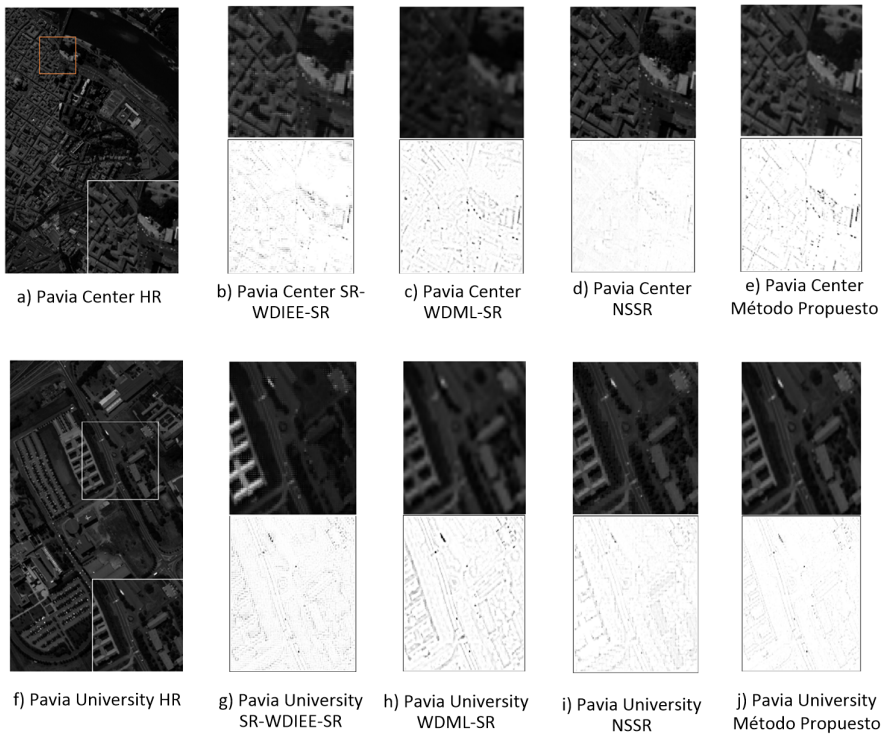


Fig. 4. Resultados visuales de Pavia Center y Pavia University.

Luego se aplica una interpolación Lanczos con factor 2 y se descompone la imagen en sub-bandas mediante la DWT. Después se toman las sub-bandas de detalle (LH, HL y HH) y se utilizan los diccionarios generados en la etapa de aprendizaje de sub-diccionarios, como se muestra en la Fig. 2, con el fin de reconstruir cada sub-banda. Consecutivamente, el algoritmo Orthogonal Matching Pursuit (OMP) [12] se aplica a los parches para encontrar los vectores de representación dispersa utilizando los diccionarios de las sub-bandas de detalle. Los vectores de representación dispersa se calculan como:

$$\gamma^{(j)} = \arg\min_{\gamma^{(j)}} \left\| p^{(j)} - D^{(j)}\gamma^{(j)} \right\|^2 \text{ donde } \left\| \gamma^{(j)} \right\|_0 \leq L. \tag{3}$$

Estos vectores son multiplicados por el diccionario $D^{(h)}$, que es resultante del conjunto $y^{(h)}$, reconstruyendo los parches de las sub-bandas mediante:

$$p^{(h)} = D^{(h)}\gamma^{(h)}. \tag{4}$$

Después, los parches de y_h se fusionan a través de un promedio del área superpuesta para crear las imágenes de sub-banda denominadas NLH, NHL, NHH. La imagen de salida de mediana resolución (NLL sparse) se reconstruye realizando IDWT con las sub-bandas reconstruidas NLH, NHL, NHH y la sub-banda de aproximación original LL.

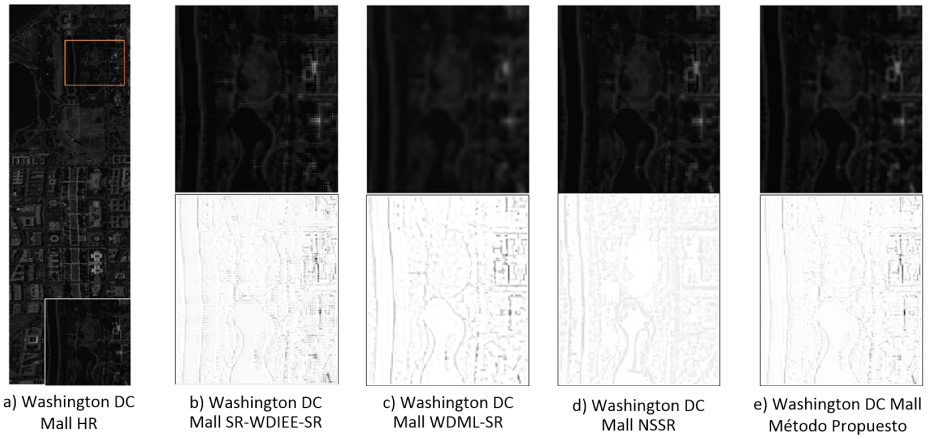


Fig. 5. Resultados visuales de Washington DC Mall.

3.3. Mejoramiento de bordes y obtención de imagen SR

La fase de mejoramiento de bordes toma una imagen LR y la descompone con el primer nivel de la DWT en 2D obteniendo las representaciones DHH, DHL, DLH y DLL. Se realiza el mismo procedimiento a la imagen LR pero con la SWT generando SHH, SHL, SLH y SLL.

Se aplica una interpolación Lanczos con factor 2 a las sub-bandas generadas por la DWT. Luego, se obtiene la diferencia entre DLL y la imagen LR para ajustar las bajas frecuencias. Posteriormente, las sub-bandas de detalles DHL, DLH y DHH son sumadas y normalizadas con las sub-bandas SHL, SLH y SHH.

Seguidamente, se extraen los bordes con estas sub-bandas de alta frecuencia de acuerdo con Feng et al. [11] mediante la siguiente ecuación:

$$S = \sqrt{(HH)^2 + (HL)^2 + (LH)^2}. \tag{5}$$

Por su parte, cada una de las sub-bandas DHL, DLH y DHH se interpola acorde con Chavez-Roman et al. [8] usando la imagen de diferencia e INN y las sub-bandas obtenidas se vuelven a interpolar con INN utilizando la imagen S. Después, cada una de estas sub-bandas es interpolada de nuevo por medio de lanczos con factor 2. Finalmente, las sub-bandas resultantes se emplean en junto con la imagen NLL sparse en el cómputo de la IDWT. La salida obtenida de este proceso resulta en una imagen de superresolución (SR) como se muestra en la Fig. 3.

4. Resultados

4.1. Métricas de evaluación

Para evaluar la calidad espacial de la reconstrucción de las imágenes SR obtenidas de las imágenes HS y MS se tomaron en consideración la métrica MPSNR que es la

Tabla 2. Comparación de resultados cuantitativos de la base de datos CAVE.

Imagen		**INN (1/4)**	**Bilinear (1/4)**	**Bicubic (1/4)**	**SR-WDIEE-SR** [8] **(1/4)**	**WDML-SR** [24] **(1/4)**	**NSSR** [10] **(1/4)**	**Propuesto (1/4)**
Balloons	MPSNR	35.74	38.39	39.37	38.99	39.33	41.88	**42.93**
	MSSIM	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**
	SAM	4.16	3.3	3.09	3.23	3.23	2.63	**1.81**
	EPI	0.12	0.38	0.4	0.33	0.4	0.58	**0.64**
	RMSE	4.21	3.11	2.78	2.86	2.75	2.05	**1.81**
Beads	MPSNR	24.9	26.05	26.9	27.94	26.84	**38.4**	31.15
	MSSIM	0.95	0.94	0.96	0.97	0.96	**0.99**	0.99
	SAM	17.83	15.52	14.31	13.96	14.41	**3.52**	8.84
	EPI	0.05	0.2	0.24	0.21	0.24	**0.87**	0.6
	RMSE	14.82	12.95	11.76	10.21	11.6	**3.06**	7.06
Cloth	MPSNR	25.93	26.24	26.69	29.63	26.64	**38.67**	32.2
	MSSIM	0.91	0.88	0.91	0.94	0.91	**0.99**	0.98
	SAM	16.49	16.77	15.04	14.66	15.12	**2.94**	7.95
	EPI	0.09	0.19	0.22	0.25	0.22	**0.9**	0.64
	RMSE	13.16	12.72	12.1	8.41	11.87	**2.97**	6.25
Face	MPSNR	35.02	36.77	37.69	35.57	37.63	**44.79**	42.47
	MSSIM	0.98	0.99	0.99	0.99	0.99	**0.99**	0.99
	SAM	10.05	8.89	8.2	8.83	8.29	**3.32**	4.1
	EPI	0.1	0.32	0.34	0.31	0.34	**0.78**	0.58
	RMSE	4.62	3.78	3.41	4.24	3.349	**1.46**	1.91
Fake and real peppers	MPSNR	33.62	35.71	37.08	35.34	36.97	**43.81**	38.93
	MSSIM	0.99	0.99	0.99	0.99	0.99	**0.99**	0.99
	SAM	8.81	7.5	6.87	7.46	6.94	**3.46**	5.07
	EPI	0.06	0.3	0.36	0.35	0.36	**0.71**	0.65
	RMSE	5.4	4.29	3.7	4.36	3.61	**1.64**	2.88
Hairs	MPSNR	32.34	33.34	34.18	38.72	34.09	**40.65**	38.33
	MSSIM	0.98	0.97	0.98	0.99	0.98	**0.99**	0.99
	SAM	11.71	11.06	10.4	7.12	10.47	**4.75**	6.96
	EPI	0.17	0.24	0.29	0.54	0.29	**0.79**	0.53
	RMSE	6.26	5.6	5.1	2.95	5.03	**2.36**	3.09

Relación Media Señal-Ruido Pico (PSNR) promediada entre las bandas de diferente longitud de onda de las imágenes, MSSIM que indica el Índice de Similitud Estructural Promedio (SSIM) medio entre las bandas, ambos señalados en [15], el Índice de Preservación de Bordes (EPI) promediado mostrado en [13] y la Raíz del Error Cuadrático Medio (RMSE) promediado.

En el caso de las imágenes HS, también se utilizó el Mapeo del Ángulo Espectral (SAM) [15] para evaluar la calidad de la reconstrucción espectral. Cabe mencionar que MPSNR, MSSIM y EPI altos significan una mejor calidad visual mientras que valores de SAM bajos implican menor distorsión espectral y una mayor calidad de reconstrucción espectral.

4.2. Descripción de datos de prueba utilizados

Para el entrenamiento de los sub-diccionarios se utilizó un total de 91 imágenes RGB [5]. Para evaluar el método propuesto, se realizaron experimentos con el siguiente conjunto de datos de imágenes HS y MS.

1. Pavia Center : contiene 102 bandas espectrales con un tamaño de 1096×715 píxeles,
2. Pavia University [3]: contiene 103 bandas espectrales con un tamaño de 610×340 píxeles,

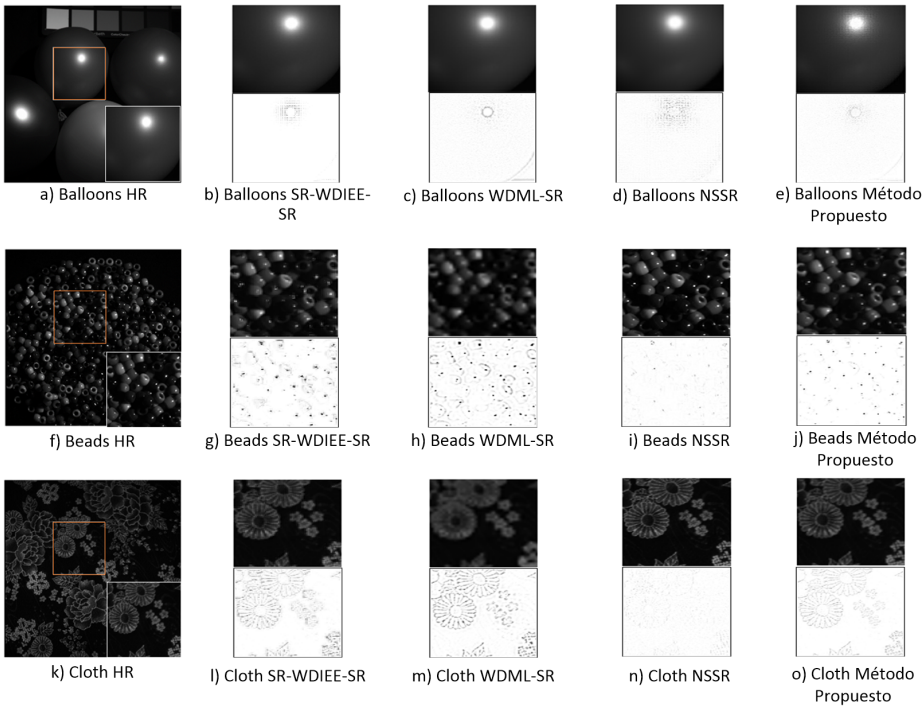


Fig. 6. Resultados visuales de las imágenes Balloons, Beads y Cloth (CAVE).

3. Washington DC Mall [4]: contiene 191 bandas espectrales con un tamaño de 1280×307.
4. CAVE [25]: contiene 31 bandas espectrales con un tamaño de 512×512 píxeles donde se toman 6 datos de prueba,
5. Zurich [23]: contiene las bandas R,G,B y NIR para la experimentación se toman 4 imágenes las cuales son zh2 con tamaño de 833×881, zh3 con tamaño de 926×943, zh6 con tamaño de 812×984 y zh19 con tamaño de 729×1223.

Los parámetros utilizados en el diccionario del método propuesto son los siguientes: el número de átomos para cada diccionario $D^{(j)}$ obtenido de los conjuntos $y^{(h)}$ y $y^{(l)}$ es 25×1024, el número máximo de iteraciones son $t = 100$, con $\eta = 1e^{-3}$ siendo η un parámetro de regularización como en [2], para el entrenamiento del diccionario y reconstrucción de la imagen se ocupó la plataforma de programación MATLAB versión R2021a.

4.3. Resultados cuantitativos y cualitativos

En esta sección se presentan los resultados de los experimentos realizados. El método propuesto representa la combinación de los enfoques de aprendizaje de diccionarios por k-SVD, OMP y la transformación en el dominio a través de DWT y SWT. Por lo que, adicionalmente a la comparación con los métodos clásicos de

Tabla 3. Comparación de resultados cuantitativos de la base de datos Zurich.

Imagen		INN (1/4)	Bilinear (1/4)	Bicubic (1/4)	SR-WDIEE-SR [8] (1/4)	WDML-SR [24] (1/4)	Propuesto (1/4)
Zh2	MPSNR	19.35	20.3	20.24	23.81	24.18	**24.83**
	MSSIM	0.78	0.79	0.8	0.95	0.92	**0.96**
	EPI	0.07	0.25	0.27	0.29	0.34	**0.38**
	RMSE	27.66	24.81	24.98	16.44	15.75	**14.62**
Zh3	MPSNR	24.75	25.99	26.39	27.58	27.97	**29.56**
	MSSIM	0.93	0.93	0.94	0.98	0.96	**0.98**
	EPI	0.09	0.27	0.31	0.31	0.34	**0.55**
	RMSE	14.94	12.98	12.41	10.65	10.18	**8.48**
Zh6	MPSNR	24	25.08	25.97	25.33	25.97	**28.04**
	MSSIM	0.93	0.92	0.94	0.98	0.94	**0.99**
	EPI	0.09	0.3	0.33	0.31	0.33	**0.7**
	RMSE	16.08	14.2	12.82	13.8	12.82	**10.1**
Zh19	MPSNR	20.07	21.13	21.28	23.23	23.8	**24.55**
	MSSIM	0.82	0.82	0.84	0.94	0.91	**0.95**
	EPI	0.07	0.25	0.28	0.3	0.32	**0.37**
	RMSE	25.29	22.38	22	17.58	16.46	**15.1**

interpolación (INN, Bilineal y Bicúbica), se contrastó su rendimiento con los algoritmos basados en aprendizaje NSSR [10] y WDML-SR [24], al igual que con el método SR-WDIEE-SR [8] que está fundamentado en las transformaciones de dominio. A cada imagen de prueba se le aplicó un sub-muestreo de 4 veces simulando una imagen LR para utilizarse como entrada de los algoritmos. La evaluación cualitativa de los métodos se realizó por medio de una imagen de error entre la imagen original Y_i y la imagen reconstruida $F(x)_i$ a través de la siguiente ecuación:

$$I_e = l_i - c\,|Y_i - F(x)_i|\,, \tag{6}$$

donde i corresponde a la i-ésima banda, l_i es el rango del valor máximo de las imágenes y $c = 3$ es una constante para amplificar el error. Se tomó en cuenta que un valor de EPI y SSIM con valor igual a 1 es una imagen idéntica a la original y que un valor SAM cercano a 0 tiene una relación espectral igual a la original.

En la Tabla 1 se observan los resultados cuantitativos obtenidos de la imagen Pavia Center, Pavia University y Washington DC Mall. Se puede observar en Pavia Center y Pavia University que los métodos clásicos de interpolación y el método SR-WDIEE-SR resultaron en valores de EPI cercanos a 0, indicando que no tienen una buena preservación de bordes, también, presentaron valores SAM altos que señalan la baja relación espectral entre las bandas de las imágenes en comparación con los otros métodos.

Adicionalmente, sus valores MPSNR y MSSIM fueron inferiores a los del método NSSR y el sistema propuesto. Para el caso de Pavia Center, NSSR proporciona un MPSNR de 32.57 y RMSE de 5.99, MSSIM de 0.98, SAM de 7.12 y EPI de 0.5 mostrando más texturas y detalles finos en la imagen de error (Fig. 5), mientras que Pavia University con el método propuesto muestra un SAM de 5.09, EPI de 0.56 y RMSE de 7.76 resultando en bordes más nítidos y percepción de mayores detalles en

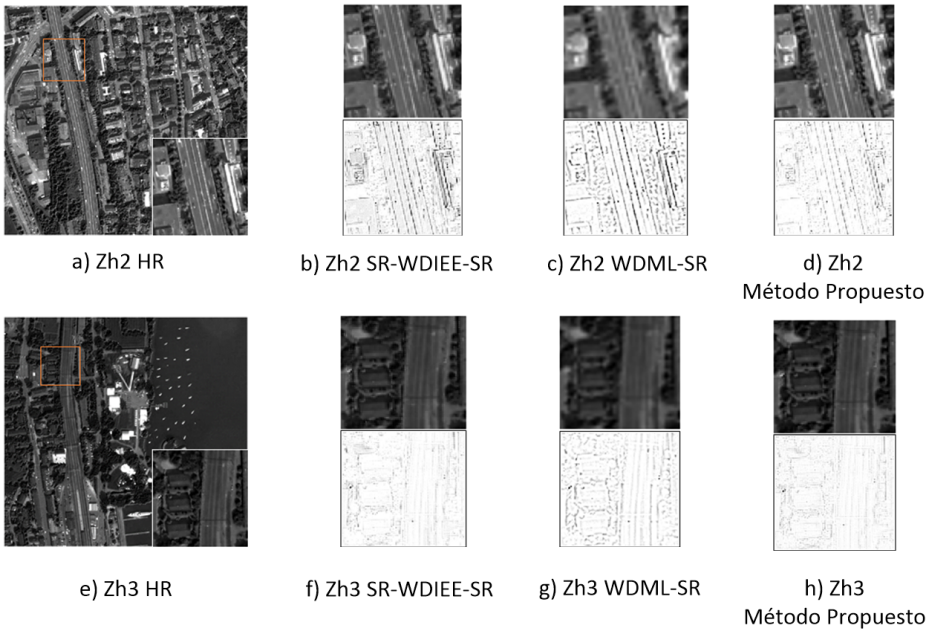


Fig. 7. Resultados visuales de las imágenes Zh2 y Zh3 (Zurich).

la imagen recortada. Por otro lado, en el caso de Washington DC Mall, los métodos clásicos, así como SR-WDIEE-SR y NSSR mostraron valores de MPSNR, RMSE, MSSIM y EPI menores al método propuesto. Esto se ve reflejado en la Fig. 6 donde la imagen de error (e) indica una reconstrucción con bordes más definidos incluso en comparación con el método WDML-SR.

En la Tabla 2 se muestran los valores promedio (MPSNR, MSSIM, EPI, SAM y RMSE) obtenidos de la base de datos CAVE. El método propuesto genera imágenes con menor error en los bordes para el caso de Balloons (Fig. 7 (e)) a comparación de los otros métodos. Sin embargo, para las demás imágenes, el método NSSR (Fig. 7 (i), (n)) disminuye de mejor forma este error, dándole nitidez a las imágenes reconstruidas.

No obstante, cabe mencionar que este método requiere ser entrenado de acuerdo con las características del sensor que capturó las imágenes híper-espectrales. Los resultados obtenidos en la base de datos de Zurich muestran que el método propuesto preserva la nitidez dado que, visualmente, la imagen reconstruida se parece en gran medida a la imagen original (Fig. 8 (d)) obteniendo RMSE de 14.62 el cual se ve reflejado en la imagen de error.

En cambio SR-WDIEE-SR (Fig. 8 (b)) muestra una imagen de error con zonas poco limpias mientras que el método WDML-SR presenta un emborronamiento en la imagen recortada. Cuantitativamente, los valores obtenidos en la Tabla 3 corroboran los resultados visuales, ya que los métodos clásicos así como SR-WDIEE-SR y WDML-SR obtienen resultados en MPSNR, MSSIM, EPI marginalmente bajos y RMSE altos con respecto al método propuesto. Los resultados experimentales obtenidos de las diferentes bases de datos de imágenes HS y MS demuestran que el método propuesto es efectivo para aumentar la calidad de las imágenes, preservando los detalles de los bordes y

resultando en imágenes claras. No obstante, cabe recalcar que el método propuesto se entrena con imágenes RGB para reconstruir imágenes de baja resolución a alta resolución, a diferencia del método NSSR, el cual es entrenado con imágenes HS de baja resolución e imágenes RGB tomadas por el mismo sensor.

Por lo que, NSSR toma en cuenta las características de las imágenes HS de un sensor específico por cada entrenamiento, es decir, se entrena un diccionario para cada base de datos de un determinado sensor y se ajustan los diferentes parámetros de regularización. En comparación, el método propuesto sólo necesita de un diccionario para el proceso de superresolución, permitiendo su uso en diferentes bases de datos y generando resultados competitivos para imágenes MS e HS como en el caso de las bases de datos de Zurich, Pavia University y Washington DC Mall.

5. Conclusiones

En este trabajo se propuso un método efectivo de superresolución de imágenes MS e HS basado en representación dispersa y reconstrucción mediante interpolación. El diccionario creado a partir de imágenes RGB mejora la precisión espacial de las imágenes MS e HS y la reconstrucción basada en interpolación de sub-bandas de alta frecuencia permite mantener la información de los bordes y características finas de la imagen.

Los resultados experimentales obtenidos de las diferentes bases de datos mostraron que este método puede lograr una menor tasa de error en reconstrucción para imágenes HS, como Pavia University y Washington DC Mall, así como en las imágenes MS utilizadas de la base de datos de Zurich, obteniendo una mejor calidad visual.

Sin embargo, los resultados obtenidos para las imágenes MS varían considerablemente de los registrados para imágenes HS y los valores SAM en imágenes HS distan de ser ideales. Por lo que es necesario mejorar el diccionario empleado con el fin de aumentar la calidad espacial-espectral en la reconstrucción de este tipo de imágenes.

Referencias

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing, vol. 54, no. 11, pp. 4311–4322 (2006) doi: 10.1109/TSP.2006.881199
2. Alvarez-Ramos, V. Ponomaryov, V., Sadovnychiy, S.: Image super-resolution via wavelet feature extraction and sparse representation. Radioengineering, vol. 27, no. 2, pp. 602–609 (2018) doi: 10.13164/re.2018.0602
3. GIC: Hyperspectral remote sensing scenes. Grupo de Inteligencia Artificial (GIC), Universidad del País Vasco (2021)
4. Remote sensing laboratory: Remote sensing datasets. School of Surveying and Geospatial Engineering (2022)
5. Kaggle.: Starter: T91 image dataset 5362b8e1-6. Data Science platform (2019)
6. Ayas, S., Ekinci, M.: Single image super resolution based on sparse representation using discrete wavelet transform. Multimedia Tools and Applications, vol. 77, no. 13, pp. 16685–16698 (2019) doi: 10.1007/s11042-017-5233-5

7. Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. M., Chanussot, J.: Hyperspectral remote sensing data analysis and future challenges. IEEE Geoscience and Remote Sensing Magazine, vol. 1, no. 2, pp. 6–36 (2013) doi: 10.1109/MGRS.2013.2244672
8. Chavez-Roman, H., Ponomaryov, V.: Super resolution image generation using wavelet domain interpolation with edge extraction via a sparse representation. IEEE Geoscience and remote sensing Letters, vol. 11, no. 10, pp. 1777–1781 (2014) doi:10.1109/LGRS.2014.2308905
9. Dengwen, Z.: An edge-directed bicubic interpolation algorithm. In: 3rd International congress on image and signal processing, vol. 3, pp. 1186–1189 (2010) doi: 10.1109/CISP.2010.5647190
10. Dong, W., Fu, F., Shi, G., Cao, X., Wu, J., Li, G., Li, X.: Hyperspectral image super-resolution via non-negative structured sparse representation. IEEE Transactions on Image Processing, vol. 25, no. 5, pp. 2337–2352 (2016) doi: 10.1109/TIP.2016.2542360
11. Feng, L., Suen, C. Y., Tang, Y. Y., Yang, L. H.: Edge extraction of images by reconstruction using wavelet decomposition details at different resolution levels. International journal of pattern recognition and artificial intelligence, vol. 14, no. 6, pp. 779–793 (2000) doi: 10.1142/S0218001400000519
12. Goklani, H. S., Sarvaiya, J. N., Fahad, A. M.: Image reconstruction using orthogonal matching pursuit (OMP) algorithm. In: 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking, pp. 1–5 (2014) doi: 10.1109/ET2ECN.2014.7044960
13. Justin, J., Sivaraman, J., Rajadurai, P., Simi, V. R.: An edge preservation index for evaluating nonlinear spatial restoration in MR images. Current Medical Imaging, vol. 13, no. 1, pp. 58–65 (2017)
14. Kirkland, E. J.: Bilinear interpolation. Advanced Computing in Electron Microscopy, 2nd edition. Springer, pp. 261–263. (2010) doi: 10.1007/978-3-030-33260-0
15. Mei, S., Yuan, X., Ji, J., Zhang, Y., Wan, S., Du, Q.: Hyperspectral image spatial super-resolution via 3D full convolutional neural network. Remote Sensing, vol. 9, no.11, pp. 1–22 (2017) doi: 10.3390/rs9111139
16. Nason, G. P., Silverman, B. W.: The stationary wavelet transform and some statistical applications. Antoniadis, A., Oppenheim, G., eds., Wavelets and statistics. Lecture Notes in Statistics, vol. 103, pp. 281–299 (1995) doi: 10.1007/978-1-4612-2544-7_17
17. Nguyen, H. V., Banerjee, A., Chellappa, R.: Tracking via object reflectance using a hyperspectral video camera. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pp. 44–51 (2010) doi: 10.1109/CVPRW.2010.5543780
18. Parker, J. A., Robert V. K., Donald E. T.: Comparison of interpolating methods for image resampling. IEEE Transactions on medical imaging, vol. 2, no. 1, pp. 31–39 (1983) doi: 10.1109/TMI.1983.4307610
19. Partridge, M., Calvo, R. A.: Fast dimensionality reduction and simple PCA. Intelligent data analysis, vol. 2, no. 3, pp. 203–214 (1998) doi: 10.3233/IDA-1998-2304
20. Rukundo, O., Hanqiang, C.: Nearest neighbor value interpolation. arXiv:1211.1768v2 (2012) doi: 10.48550/arXiv.1211.1768
21. Tarabalka, Y., Chanussot, J., Benediktsson, J. A.: Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 40, no. 5, pp. 1267–1279 (2010) doi: 10.1109/TSMCB.2009.2037132
22. Uzair, M., Mahmood, A., Mian, A. S.: Hyperspectral face recognition using 3D-DCT and partial least squares. British Machine Vision Conference (BMVC), pp. 57.1–57.10 (2013) doi: 10.5244/C.27.57

23. Volpi, M., Ferrari, V.: Structured prediction for urban scene semantic segmentation with geographic context. Joint Urban Remote Sensing Event (JURSE), pp. 1–4 (2015) doi: 10.1109/JURSE.2015.7120490
24. Wu, X., Fan, J., Xu, J., Wang, Y.: Wavelet domain multidictionary learning for single image super-resolution. Journal of Electrical and Computer Engineering, Vol. 2015, pp. 1–12 (2015) doi: 10.1155/2015/526508
25. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S. K.: Generalized assorted pixel camera: Post-capture control of resolution, dynamic range and spectrum: Technical Report CUCS-061-08. Department of Computer Science, Columbia University (2008)

ISSN 1870-4069

Optimización de hiperparámetros de una red Long Short Term Memory para pronósticos financieros

Francisco J. Pedroza-Castro, Alfonso Rojas-Domínguez, Martín Carpio, Manuel Ornelas-Rodríguez, Héctor Puga

Tecnológico Nacional de México,
Instituto Tecnológico de León,
México

alfonso.rojas@gmail.com

Resumen. El proceso de optimización de hiperparámetros de una red neuronal artificial, usualmente se lleva cabo de manera manual usando Grid Search o Random Search. En este artículo, estudiamos una metodología para optimizar hiperparámetros de una red Long Short Term Memory (LSTM) usando dos metaheurísticas bio-inspiradas, Particle Swarm Optimization, y Flower Pollination Algorithm y una Estimation of Distribution Algorithm (EDA), Low Number of Function Evaluation, para que la red realice un pronóstico financiero del precio de cierre del siguiente día de las acciones de Google y Nike. Los resultados muestran que las redes LSTM obtenidas mediante el proceso de optimización de hiperparámetros son más simples, resultando en menor tiempo de entrenamiento, y mayor rendimiento, que las redes no optimizadas con metaheurísticas. Los errores de prueba obtenidos de las soluciones son de 10E-4 hasta 10E-6. Adicionalmente realizamos una comparación de las metaheurísticas, concluyendo que la EDA encuentra una solución esperada con menor cantidad de llamadas a función, lo que se traduce en menor costo computacional y tiempo de ejecución, con resultados satisfactorios.

Palabras clave: Long Short Term Memory, estimation of distribution algorithm, bio-inspired metaheuristics, algorithm for a low number of function evaluations, particle swarm optimization, flower pollination algorithm, pronóstico financiero.

Hyperparameter Optimization of a Long Short Term Memory Network for Financial Forecasting

Abstract. The hyperparameter optimization process of an artificial neural network is usually carried out manually using Grid Search or Random Search. In this paper, we study a methodology to optimize hyperparameters of a Long Short Term Memory (LSTM) network using two bio-inspired metaheuristics, Particle Swarm Optimization, and Flower Pollination Algorithm, and an Estimation of Distribution Algorithm (EDA), Low Number of Function Evaluation, for the

network to make a financial forecast of the next day's closing price of Google and Nike stock. The results show that the LSTM networks obtained through the hyperparameter optimization process are simpler, resulting in less training time, and higher performance, than the networks not optimized with metaheuristics. The resulting proof errors of the solutions are 10E-4 up to 10E-6. Additionally, we perform a comparison of the metaheuristics, concluding that the EDA finds an expected solution with fewer function calls, which translates into lower computational cost and execution time, with satisfactory results.

Keywords: Long Short Term Memory, estimation of distribution algorithm, bio-inspired metaheuristics, algorithm for a low number of function evaluations, particle swarm optimization, flower pollination algorithm, financial forecast.

1. Introducción

Inversionistas, academia e investigadores, del campo de pronósticos financieros se han visto interesados por el campo de la inteligencia computacional, ya que las investigaciones muestran ventajas significativas en contraste con los métodos clásicos, como análisis técnico, análisis fundamental, y métodos estadísticos [12, 13, 21].

Las investigaciones en inteligencia computacional aplicadas a pronósticos financieros, muestran que el uso de redes neuronales híbridas con algoritmos evolutivos, obtienen un mayor rendimiento en comparación con las redes no-híbridas. Dichas hibridaciones van desde la optimización de parámetros hasta hiperparámetros [12,13,21]. Este último es un campo del Automated Machine Learning, conocido como Hyperparameter Optimization (HPO). Por lo que en este artículo nos referimos a la hibridación para ajuste de hiperparámetros como HPO [11].

De hecho, el proceso de ajuste de hiperparámetros de una Red Neuronal Artificial, comúnmente se lleva a cabo usando Random Search o Grid Search [15]. HPO busca realizar este proceso de manera automatizada utilizando, comúnmente, diferentes técnicas como Algoritmos Genéticos y Métodos Bayesianos [11, 15].

El presente artículo, muestra una metodología propia para llevar a cabo la optimización de hiperparámetros, probando 3 metaheurísticas: 2 algoritmos bio-inspirados, Particle Swarm Optimization y Flower Pollination Algorithm, y 1 algoritmo perteneciente a las Estimation of Distribution Algorithm, Estimation of Distribution Algorithm Low Number of Function Evaluation (EDALNFE, en este artículo nos referiremos a él como LNFE).

Estos algoritmos se compararon y adicionalmente se contrastaron con una red LSTM que fue ajustada de manera manual por [16]. Los resultados obtenidos muestran que la optimización de hiperparámetros con metaheurísticas diseñan una red más simple con mejores rendimientos en comparación de la hechas manualmente.

El trabajo está organizado como sigue: en la Sección 2, presentamos diferentes trabajos que se han utilizado para realizar pronósticos financieros utilizando redes neuronales artificiales, investigaciones que optimizan hiperparámetros en pronósticos financieros y de otros campos. En la Sección 3, se describe la teoría de la red LSTM y

las metaheurísticas. En la Sección 4, exponemos la metodología propuesta para realizar la optimización de hiperparámetros y comparación de metaheurísticas.

En la sección 5 se muestran los resultados experimentales y discusiones, y finalmente en la Sección 6 las conclusiones de nuestro trabajo.

2. Trabajos relacionados

Los investigadores en pronósticos financieros han encontrado que el uso de inteligencia computacional genera resultados satisfactorios que superan a los métodos clásicos [4, 12–14, 16, 21]. Estos métodos van desde un Algoritmo Genético (AG) [4, 7], Lógica Difusa [4, 6], Sistemas Expertos [4], y Redes Neuronales Artificiales (RNAs) [18], entre otros. Las RNAs se han utilizado mayormente, especialmente la red LSTM, que de acuerdo a la literatura, son las que mayor rendimiento tienen [12–14, 19, 21, 24].

En la literatura para pronósticos financieros usando redes neuronales, del periodo de 2017 a 2019, cerca del 51 % de los artículos revisados utilizan algún tipo de Red Neuronal Recurrente (RNR), de ese porcentaje, el 61 % utilizan redes LSTM, la razón es la misma que se menciona en el párrafo anterior [21], además de ser relativamente fácil de aplicar.

Kumar *et. al* (2021) optimizan hiperparámetros de una red LSTM para pronósticos de tendencias de activos financieros, utiliza PSO y FPA. Su trabajo lo comparan con un modelo sin uso de metaheurísticas, los resultados muestran que utilizar metaheurísticas para la optimización, genera resultados superiores a los casos donde no se usa metaheurísticas [13].

En otro trabajo se realiza un sistema de comercio automatizado usando un AG y un RNA, obteniendo 72.5 % de precisión y 23.3 % de Rendimiento Anual Neto. En su trabajo encontraron que existen tres factores importantes que afectan el rendimiento del pronóstico: Pre-procesamiento de datos adecuado y presentación de los datos, estrategias óptimas de comercio, y la estructura del modelo a pronosticar [7].

En otros trabajos se realiza un sistema de comercio utilizando una red LSTM [22], su red la llevan a un sistema simulado considerando los costos de transacción, y diferentes versiones de la red LSTM, con lo cual llegan a tener rendimientos de hasta 228.94 % en el mejor de los casos.

Sezer *et. al* (2020), además, identifica que en diversos trabajos de pronósticos financieros, con redes neuronales artificiales, el ajuste de hiperparámetros es de vital importancia para un adecuado rendimiento de la red al realizar el pronóstico de activos financieros. Consideran que encontrar los mejores hiperparámetros de las redes neuronales es un problema significativo.

De hecho HPO optimizan los hiperparámetros usando Métodos Bayesianos (MB) como Sequential Model-Based Global Optimization (SMBGO), The Gaussian Process Approach (GPA), Tree-structured Parzen Estimator Approach (TSPEA) [2, 3], entre otros algoritmos. Por ejemplo, se ha realizado optimización de hiperparámetros en campos diferentes a pronósticos financieros, donde se usa GA y MB [11, 15], con resultados satisfactorios. También se ha hecho uso de Aprendizaje por Refuerzo Profundo [5], pero en este caso los resultados superan a GA y MB, incluso, en trabajos relacionados, se ha

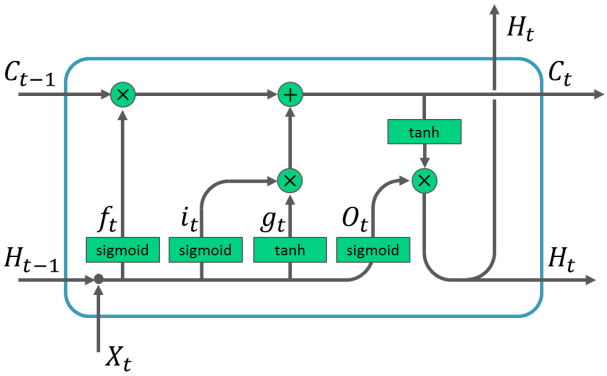


Fig. 1. Estructura de una celda LSTM (Elaboración propia).

hecho uso de una red LSTM para la optimización, teniendo resultados satisfactorios y en un tiempo reducido [13].

Otro trabajo de optimización de hiperparámetros de una red tipo LSTM fue el de [1], quien consideran que es una área de oportunidad para las metaheurísticas con el objeto de buscar los hiperparámetros de la red mencionada. Ellos proponen utilizar Grey Wolf Optimizer (GWO), con lo cual encuentran que una red LSTM de arquitectura simple, la cual puede tener el mismo rendimiento que una red más compleja. Atribuyéndolo a la adecuada combinación de hiperpárametros.

En este trabajo, siguiendo la línea marcada por [12], utilizaremos algoritmos evolutivos para la optimización de hiperparámetros, los cuales son dos bio-inspirados ya utilizados en la literatura, PSO, y FPA, los cuales han tenido resultados satisfactorios al optimizar los hiperparámetros de una red LSTMA; adicionalmente utilizaremos una EDA, EDALNFE que de acuerdo a su diseño hace una menor cantidad de llamadas a función, lo cual puede ser una ventaja, ya que las otras metaheurísticas tiene que hacer llamadas a función a toda la población en cada iteración; en el presente caso, que se usa RNAs, la llamada a función implica entrenar la red en cada iteración, generando un alto costo computacional. El proceso de optimización se seguirá utilizando una metodología propuesta por nosotros. Al final contrastamos cada uno de los algoritmos con una red LSTM que se ajustó sin metaheurísticas por [16].

3. Teoría

3.1. Long Short Term Memory (LSTM)

La red LSTM es un tipo de RNR que puede trabajar con hasta 1000 retrasos en el tiempo, evitando los problemas del *exploding-gradient* y el *vanishing-gradient* [20]. La red trabaja simulando compuertas lógicas que permiten eliminar o agregar información a un *Cell State* (5) como se describe a continuación[1]: primero Forget gate (2) decide que información se olvida o se mantiene; después, la Input gate (1) y Candidate gate

[1] En este trabajo, $\otimes$ representa *element-wise* o el producto Hadamard, sig$(\cdot)$ es la función sigmoide y tanh$(\cdot)$ función tangente hiperbólica.

(4) deciden que información se agrega; finalmente se genera el Hidden state (6) usando Output gate (3) y la Cell state (5) [10]. Las ecuaciones para llevar dicho proceso se muestran a continuación. En la Figura 1 se muestra el funcionamiento de la red LSMT:

$$\textbf{Input gate:} \quad i_t = \text{sig}(x_t U_i + h_{t-1} W_i + b_i), \quad i_t \in \mathbb{R}^{d_H}, \tag{1}$$

$$\textbf{Forget gate:} \quad f_t = \text{sig}(x_t U_f + h_{t-1} W_f + b_f), \quad f_t \in \mathbb{R}^{d_H}, \tag{2}$$

$$\textbf{Output gate:} \quad O_t = \text{sig}(x_t U_o + h_{t-1} W_o + b_o), \quad O_t \in \mathbb{R}^{d_H}, \tag{3}$$

$$\textbf{Candidate gate:} \quad g_t = \tanh(x_t U_g + h_{t-1} W_g + b_g), \quad g_t \in \mathbb{R}^{d_H}, \tag{4}$$

$$\textbf{Cell state:} \quad C_t = f_t \otimes C_{t-1} + i_t \otimes g_t, \quad C_t \in \mathbb{R}^{d_H}, \tag{5}$$

$$\textbf{Hidden state:} \quad H_t = O_t \otimes \tanh(C_t), \tag{6}$$

$$\textbf{State:} \quad s_t = R_{LSTM}(s_{t-1}, x_t) = [c_t; H_t], \quad s_t \in \mathbb{R}^{2d_H}, \tag{7}$$

donde U y $W \in \mathbb{R}^{d_x \times d_H}$ son los pesos, $x_t \in \mathbb{R}$. Si se utiliza tamaños de lote d_b, entonces: $s_t \in \mathbb{R}^{2d_H \times d_b}$, C_t, H_t, i_t, f_t, O_t, $g_t \in \mathbb{R}^{d_H \times d_b}$, y $b \in \mathbb{R}^{d_b \times d_H}$.

3.2. Particle Swarm Optimization (PSO)

Esta metaheurística se inspira en el comportamiento social de insectos, animales y humanos. El algoritmo inicia con una población aleatoria, donde cada partícula de la población se considera una solución potencial, las cuales se van desplazando en el espacio de búsqueda con el objeto de mejorar la mejor solución encontrada $\mathbf{g}^*$ en la iteración previa, usando la información de los vecinos $\mathbf{g}^*$ y su mejor posición $\mathbf{p}_i$ (en la que alguna vez estuvo) [9]. Este concepto se presenta en la regla de actualización (8):

$$\mathbf{v}_i^{New} = \chi(w\mathbf{v}_i + \varphi_1 \otimes (\mathbf{p}_i - \mathbf{x}_i) + \varphi_2 \otimes (\mathbf{g}^* - \mathbf{x}_i)), \tag{8}$$

donde $\mathbf{v}_i$ es la velocidad de la partícula i, $\mathbf{x}_i$ es la posición de la partícula i, w es la inercia o peso de inercia, $\mathbf{g}^*$ es la mejor posición global, $\varphi_1 = R_1 c_1$, $\varphi_2 = R_2 c_2$, donde c_1 es el coeficiente cognitivo, c_2 el coeficiente social, $R_1, R_2 \sim \mathcal{U} \in [0, 1]$, y χ el coeficiente de constricción dado por la Ec. (9). La Ec. (10) es la regla actualización de la posición de la partícula. El pseudocódigo del PSO se muestra en Algoritmo 1:

$$\chi = \frac{2}{\left|2 - \varphi' - \sqrt{\varphi'^2 - 4\varphi'}\right|}, \quad \varphi' = c_1 + c_2, \varphi' > 4, \tag{9}$$

$$\mathbf{x}_i^{New} = \mathbf{x}_i + \mathbf{v}_i^{New}. \tag{10}$$

3.3. Flower Pollination Algorithm (FPA)

El FPA es un algoritmo inspirado en el proceso de polinización de las plantas bajo la óptica del proceso de optimización [25]. Tanto FPA como PSO son algoritmos bio-inspirados, los cuales son similares al seguir un proceso de generación de población

Algorithm 1 PSO, asumiendo minimización.

Generación de población inicial de manera aleatoria $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
Encontrar la mejor solución $\mathbf{g}^*$ en X.
while *Criterio de paro* no se cumpla, **do**
    for partícula $\mathbf{x}_i$ **do**
        if $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ **then**
            $\mathbf{p}_i \leftarrow \mathbf{x}_i$; actualizar la mejor posición de la partícula i.
        end if
        Identificar la mejor posición de la iteración $\mathbf{g}^*$.
        Actualizar velocidad $\mathbf{v}_i$ vía (8).
        Actualizar posición $\mathbf{x}_i$ vía (10).
    end for
end while
return La mejor solución encontrada $\mathbf{g}^*$.

inicial y mover las soluciones en el espacio de búsqueda, no obstante FPA añade ruido usando una distribución Lévy para explorar, mientras que PSO solo usa números aleatorios uniformes. El FPA usa dos conceptos de manera general, polinización global y polinización local, para realizar la búsqueda. Las Ecs. (11) a (13) representan dichas ideas. El pseudocódigo se muestra en el Algoritmo 2.

Algorithm 2 FPA, asumiendo minimización.

Crear población inicial de manera aleatoria $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
Encontrar la mejor solución $\mathbf{g}^*$.
Define el sego de polinizavión $\rho \in [0, 1]$.
while *Criterio de paro* no es encontrado, **do**
    for por cada flor/gameto $\mathbf{x}_i$ ($i = 1$ to n) **do**
        if $rand < \rho$ **then** ▷ Donde $rand \in [0, 1]$.
            Crea números aleatorios por cada variable de la solución usando el vuelo de Lévy.
            Crea solución temporal $\mathbf{x}'$ con la polinización global Ec. (11).
        else
            Crea un número aleatorio ϵ de manera uniforme ($\mathcal{U} \in [0, 1]$) por cada variable.
            Crea solución temporal $\mathbf{x}'$ con la polinización local Ec. (14).
        end if
        if $f(\mathbf{x}') < f(\mathbf{x}_i)$ **then**
            $\mathbf{x}_i \leftarrow \mathbf{x}'$
        end if
    end for
    Identifica la mejor solución $\mathbf{g}^*$ de la iteración.
end while
return La mejor solución encontrada $\mathbf{g}^*$.

La polinización global se define por (11), donde $\mathbf{x}_i^t$ es una flor/gameto, $\mathbf{g}^*$ la mejor solución (flor/gameto) global, γ es el factor de escalamiento del vuelo Lévy, y $L(\lambda)$ es un número aleatorio obtenido mediante la distribución del vuelo de Lévy de acuerdo a la Ec. (12):

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma L(\lambda) \otimes (g^* - \mathbf{x}_i^t), \tag{11}$$

$$L(\lambda) \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad s \gg s_0 > 0, \quad s = \frac{U}{|V|^{1/\lambda}}, \tag{12}$$

donde $V \sim \mathcal{N}(0,1)$ y $U \sim \mathcal{N}(0, \sigma_{fpa}^2)$, con desviación estándar σ_{fpa} obtenida con la Ec. (13):

$$\sigma_{fpa}^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right]^{1/\lambda}. \tag{13}$$

La polinización local se define por (14), donde $\mathbf{x}_k^t$ y $\mathbf{x}_j^t$ son dos flores/gametos tomadas de manera aleatoria de la población. Y ϵ es un número aleatorio en el intervalo $[0, 1]$:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon \otimes (\mathbf{x}_j^t - \mathbf{x}_k^t). \tag{14}$$

3.4. Algorithm for a Low Number of Function Evaluations (EDALNFE)

LNFE[2] pertenece a la familia de Estimation of Distribution Algorithms (EDAs). La idea de una EDA es iniciar con una población la cual se usa para crear una media y varianza que es utilizada para generar una nueva muestra poblacional mediante una distribución de probabilidad previamente definida.

La idea central del LNFE es crear un vector direccional con la correlación de Pearson, para crear una nueva población. La ventaja de este algoritmo es que hace una menor cantidad de llamadas a función [17]. El pseudocódigo se muestra en el Algoritmo 3[3].

Este algoritmo inicia con una población, cada solución es evaluada y se le asigna una probabilidad de acuerdo a su fitness. La mejor solución y su fintess, X_{elite}, F_{elite}, se usa para generar una media y varianza (ver [23]). Si el fitness no mejora después de n iteraciones la población se reinicia preservando la mejor solución. La nueva población se genera mediante dos distribuciones (ver línea 8 en Algoritmo 3), una distribución normal multivariada y una distribución sobre el vector dirigido a una dirección prometedora. En cada iteración se verifica que el vector proyección obtenido por la correlación de Pearson mejore al mejor fitness, de lo contrario se reinicia la población. Este proceso se repite hasta cumplir con el criterio de paro (para más detalles ver [17]).

4. Metodología

Para realizar la optimización de hiperparámetros de la red LSTM y pueda así pronosticar los precios de cierre del siguiente día de las acciones de Google y Nike.

[2] En este articulo nos referimos a EDALNFE como LNFE.

[3] Este algoritmo trabaja en el espacio $[0, 1]$, lo cual separa al problema del algoritmo.

Algorithm 3 Algorithm for a Low Number of Function Evaluations (EDALNFE).

$N_p \leftarrow 8D + 2$; $C_{elite} \leftarrow 0$; $C_{resets} \leftarrow 0$
$X \leftarrow$ initialize(d, N_p) ▷ Inicia población de manera aleatoria X.
$F \leftarrow$ evaluate$(X, f(\cdot))$ ▷ Donde $f(\cdot)$ es la función fitness.
$[ibest, ps] \leftarrow$ selection(F)
$[X_{elite}, F_{elite}] \leftarrow [X_{ibest(1)}, F_{ibest(1)}]$
$f_{reset} \leftarrow F_{elite(1)}$ ▷ Guardar la mejor solución.
while *Criterio de paro* ($var_c > var_{min}$) no se cumple, **do**
    $\left[\hat{X}, U\right] \leftarrow$ generateNewIndividuals$(X, F, X_{elite}, ps, ibest, U \leftarrow rand(D))$
    $\hat{F} \leftarrow$ evaluate$(\hat{X}, f(\cdot))$ ▷ Evaluar la nueva población.
    if $\left|F_{elite} - F_{ibest(1)}\right| > th_{elite}$ **then** ▷ Si no mejora
        $C_{elite} \leftarrow 0$ ▷ Reiniciar población.
        $[X_{elite}, F_{elite}] \leftarrow \left[X_{ibest(1)}, F_{ibest(1)}\right]$ ▷ Actualizar la mejor solución elite.
    else
        $C_{elite} \leftarrow C_{elite} + 1$ ▷ Incrementar el contador de elite.
    end if
    if $C_{elite} = MaxC_{elite}$ **then** ▷ Si la petición de la mejor solución se repite en un limite, reiniciar.
        $C_{resets} \leftarrow C_{resets} + 1$ ▷ Incrementar el contador de reinicio.
        $X_{elite} \leftarrow X_{ibest(1,\ldots,C_{resets})}$
        $F_{elite} \leftarrow F_{ibest(1,\ldots,C_{resets})}$
        $X \leftarrow$ reinitialize$(D, Np - 1, C_{resets}, X_{elite})$ ▷ Reiniciar población.
        $F \leftarrow$ evaluate$(X, f(\cdot))$ ▷ Evaluar la nueva población.
        $[X, F] \leftarrow [[X, X_{elite}], [F, F_{elite}]]$ ▷ Agregar la mejor solución elite a la nueva población.
        $[ibest, F_{elite}] \leftarrow$ selection(F)
        $[X_{ibest}, ps] \leftarrow \left[X_{ibset(1)}, F_{ibest(1)}\right]$
        $C_{elite} \leftarrow 0$
        if $F_{elite(1)} = f_{reset}$ or $C_{resets} = MaxC_{resets}$ **then** ▷ Si no mejora, detener el algoritmo.
            break
        else
            $f_{reset} \leftarrow F_{elite(1)}$ ▷ Si la solución mejora actualizarla.
        end if
    end if
    $[X, F] \leftarrow [X(ibest), F(ibest)]$ ▷ Ordenar las soluciones de acuerdo a su valor fitness.
    $[X, F] \leftarrow [X(1, \ldots, Np/2), F(1, \ldots, Np/2)]$ ▷ Mantener la mitad de la población con mejor fitness.
    $[X, F] \leftarrow \left[\left[X, \hat{X}\right], \left[F, \hat{F}\right]\right]$ ▷ Añadir nuevas soluciones a la población.
end while
return La mejor solución encontrada, $[x_{elite}, f_{elite}]$

Utilizamos 2 algoritmos bio-inspirados (PSO y FPA) y una EDA (LNFE). Para realizar el entrenamiento de las redes LSTM usamos el optimizador Adam y, como función de pérdida el error cuadrático medio. Seleccionamos el 65 % de la serie de tiempo del precio de cierre para el conjunto de entrenamiento y 35 % para el conjunto de prueba.

Tabla 1. Parámetros usados en metaheurísticas.

Parámetro	Valor
PSO	
Coeficiente cognitivo, c_1	2.05
Coeficiente social, c_2	2.05
Peso de inercia, w	0.8
Número de partículas (Soluciones)	25 (Usado en [12])
Número de iteraciones	16
FPA	
Sesgo de polinización, ρ	0.8
Factor de escalamiento de vuelo Lévy, γ	0.1
Número Flores/gametos (Soluciones)	25 (Usado en [12])
Número de iteraciones	16
LNFE	
Número de soluciones	26

Los datos se obtuvieron de la web Yahoo Finance[4]. Los periodos seleccionados para las acciones de Google fuero de 24/08/2004 a 21/01/2022; para el caso de Nike el periodo fue de 02/12/1980 a 24/01/2022. Estas acciones fueron las mismas que usaron [16]; se replicó el trabajo de estos últimos para compararlo con las metaheurísticas. Es menester resaltar que [16] en su trabajo únicamente muestra 4 configuraciones de red LSTM, cambiando únicamente el número de épocas, sin mostrar otras configuraciones con demás hiperparámetros. Al modelo de [16] nos referiremos como Optimización sin Metaheurística (OSM).

En cada experimento se decidió normalizar los datos usando $min - max$ en el intervalo entre $[0, 1]$. Los hiperparámetros que se optimizaron se muestran en la primera columna del Cuadro 2, y su símbolo en la segunda columna. Los hiperparámetros Cl_s, H_s, α, W_s, B_s, y E_s, fueron seleccionados como lo hace [13] (ver Cuadro 2).

Los parámetros de control del PSO fueron seleccionados como sugiere [9] en el caso de FPA se seleccionaron como sugiere [25]. En ambos casos (PSO y FPA) el número de soluciones iniciales se seleccionan como sugiere [13]. En el caso de LNFE la población fue de 26, ya que el diseño del algoritmo sólo así lo permite. Estos parámetros son mostrados en el Cuadro 1.

Tanto PSO como FPA trabajan sobre el espacio de los números continuos. Por lo que decidimos mapear los valores al espacio de los discretos redondeando los números como lo sugiere [8] para solucionar problemas de variables mixtas. De tal modo las soluciones tienen valores de acuerdo al Cuadro 2.

El espacio de tamaño de lote se seleccionó como se usa típicamente en RNAs. Las épocas se seleccionaron de acuerdo a experimentos previos y el trabajo de [16]. El tamaño de la ventana se usa de acuerdo a las investigaciones realizadas por [21]. El número de estados ocultos se seleccionó para reducir el espacio de búsqueda, no obstante, se incluye la solución de [16]. El número de celdas LSTM se selecciona como sugiere [13] (ver Cuadro 2). Para realizar una comparación justa de las metaheurísticas

[4] https://finance.yahoo.com/

Tabla 2. Hiperparámetros y espacio de búsqueda.

hiperparámetro	Simbolo	Espacio de búsqueda	Limite
Celda LSTM	Cl_S	$\{1, 2, 3, 4, 5, 6, 7, 8\}$	$[1, 8]$
Estado oculto	H_S	$\{1, 20, 50, 80, 96, 110, 150\}$	$[1, 7]$
Tamaño de la ventana	W_S	$\{20, 40, 60, 80, 90, 100\}$	$[1, 6]$
Factor de aprendizaje	α	$\left\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\}$	$[1, 5]$
Tamaño de lote	B_S	$\{8, 16, 32, 64\}$	$[1, 4]$
Número de épocas	E_S	$\{12, 25, 50, 100, 150\}$	$[1, 5]$

decidimos utilizar un mínimo de 400 llamadas a función, dicho número se fijó por experimentos previos.

Para entrenar la red se consideró añadir condiciones de paro para disminuir el costo computacional, consistiendo detener el entrenamiento de no mejora la pérdida de la red después de 5 épocas. La metodología de optimización de hiperparámetros se puede ver en la Figura 2.

Por otro lado, también se decidió utilizar la misma semilla de generación de número aleatorios de las soluciones iniciales para comparar a los algoritmos bajo condiciones similares.

Una vez obtenidas las soluciones por cada metaheurística, se realizaron 31 experimentos de entrenamiento de cada solución para obtener evidencia estadística de cada solución.

Por último, para revisar el fitness de convergencia, realizamos gráficas de convergencia, así como comparar sus desviaciones estándar de la primera generación a la última iteración de cada metaheurística.

5. Resultados experimentales y discusiones

Los experimentos se realizaron en una computadora con procesador RYZEN 5 5600x, una tarjeta gráfica GPU-GTX-1050TI, y RAM de 16GB. Se utilizó el lenguaje Python, usando principalmente NumPy, matplotlib y Keras de TensorFlow. El tiempo de ejecución fue de entre 4 y 7 horas tanto para PSO, FPA y LNFE, para cada uno de los conjuntos de datos (Google y Nike).

Los hiperparámetros encontrados por cada una de las metaheuristicas se encuentra en el Cuadro 3. Se analizó la convergencia de cada uno de los algoritmos. En la Figura 3 se muestran algunas gráficas de convergencia ya que de agregar todas las imágenes el número de páginas sería mayor a lo requerido.

Adicionalmente en el Cuadro 4 se muestran las diferencias de las dispersión de la primera población y la última población de cada una de las variables, de cada algoritmos y de cada acción[5]. En el Cuadro 4 se aprecia que en PSO la convergencia en cada variable disminuyó a excepción de FPA, que incluso en algunos casos aumenta la dispersión. En apariencia PSO muestra mejor convergencia que LNFE, no obstante como se aprecia en la Figura 4 y 3 en 400 llamadas a función PSO llegó a la solución

[5] Las soluciones fueron escaladas en el intervalo $[0, 1]$ para poder hacer una comparación relativa.

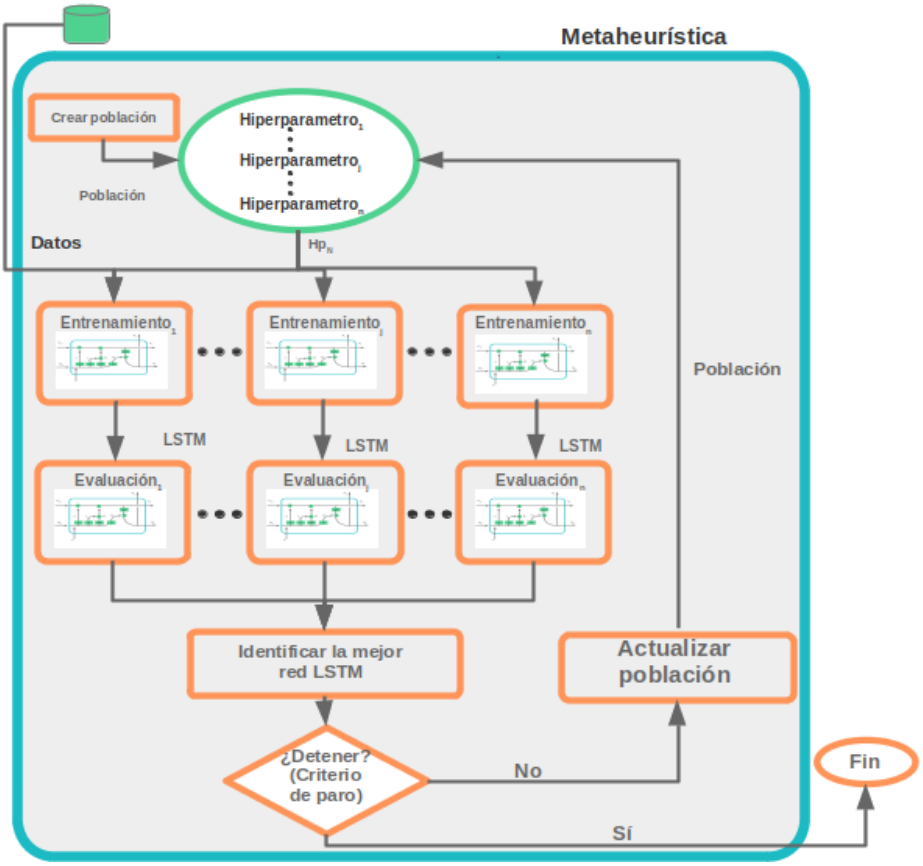


Fig. 2. Metodología de optimización. Hp_N representa el N hiperparámetro (Elaboración propia).

mostrada en el Cuadro 3, no obstante LNFE, llegó a la solución en una menor cantidad de iteraciones y con menor cantidad de llamadas a función.

En la última iteración LNFE tiene mayor dispersión a comparación de PSO, empero el algoritmo esta diseñado para renovar su población si se repite el fitness después de n iteraciones, por lo que explicaría una mayor dispersión que PSO en su población final (ver Cuadro 4).

Por otro lado, FPA no disminuyó en gran medida su dispersión en la población final, la cual se mantuvo en cada iteración como se muestra en la Figura 3 y el Cuadro 4, incluso en algunos casos aumentó su dispersión. En el Cuadro 3 se muestran las soluciones encontradas, así como los resultados de 31 experimentos de entrenamiento de cada una de las redes.

Como se aprecia en las soluciones encontradas por las metaheurísticas, la arquitectura de la red LSTM es similar en cada uno de los casos a excepción de LSTM-PSO-Nike[6]. Empero, cada solución es más simple que las redes LSTM-OSM-Google y

[6] Cuando nos referimos a LSTM-PSO-Nike significa, red LSTM optimizada con PSO para pronosticar las acciones de Nike.Del mismo modo para la combinación de las diferentes siglas.

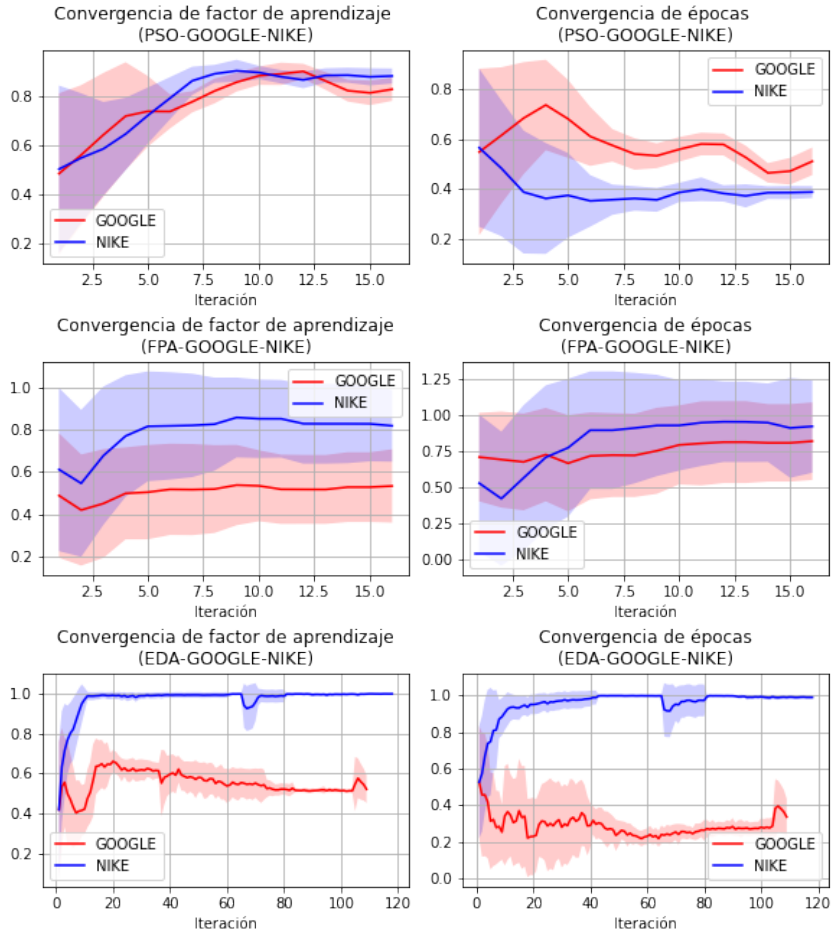


Fig. 3. Convergencia del factor de aprendizaje y número de épocas. Las gráficas están escaladas en el intervalo $[0, 1]$.

LSTM-OSM-Nike, inclusive la media de la pérdida es menor en todos los casos.

En el caso de LSTM-PSO-Google, se muestra que es el mejor fitness entre LSTM-FPA-Google, LSTM-LNFE-Google y LSTM-OSM-Google, no obstante la media de los 31 experimentos muestra que LSTM-LNFE-Google tiene mejor rendimiento, mientras que en el caso de LSTM-FPA-Nike y LSTM-LNFE-Nike muestra un mejor rendimiento que LSTM-PSO-Nike, con fitness similar como se aprecia en la Figura 6.

Empero LNFE, encontró la solución con una menor cantidad de llamadas a función como se ven Figura 3. Los resultados promedio de pronósticos de 31 experimentos de LNFE y OSM se muestra en la Figura 5. En dichas imágenes se aprecia la mejora sustancial de usar LNFE, lo cual sucede de igual manera con los demás pronósticos y se puede apreciar en el Cuadro 3.

No se presentan los demás experimentos para evitar tener un número excesivo de imágenes. Dichos pronósticos se logran con una red LSTM más simple que las redes

Tabla 3. LSTM optimizada para realizar el pronóstico de las acciones.

	Google				Nike			
hiperparámetro	**PSO**	**FPA**	**LNFE**	**OSM**	**PSO**	**FPA**	**LNFE**	**OSM**
Cl_s	1	1	1	4	1	1	1	4
H_s	150	150	150	96	110	150	150	96
W_s	60	80	40	50	90	100	40	50
α	10^{-4}	10^{-3}	10^{-4}	10^{-2}	10^{-4}	10^{-4}	10^{-4}	10^{-2}
B_s	16	32	8	16	16	16	8	16
E_s	50	150	100	100	50	150	150	100
Datos de solución	**PSO**	**FPA**	**LNFE**	**OSM**	**PSO**	**FPA**	**LNFE**	**OSM**
Llamadas a función:	400	400	401	-	400	400	402	-
Fitness:	3.18E-5	3.41E-5	3.32E-5	-	5.25E-6	4.23E-6	1.24E-7	-
Media de pérdida:	3.53E-5	2.91E-5	2.91E-5	1.86E-3	5.35E-6	3.96E-6	3.97E-6	1.02E-3
Media de prueba:	2.60E-4	3.28E-4	1.49E-4	2.22E-3	1.78E-4	1.04E-4	1.08E-4	2.97E-3
Desviación estándar de pérdida:	2.10E-6	1.58E-6	1.04E-6	7.80E-5	1.93E-7	9.13E-8	1.51E-7	5.61E-5
Desviación estándar de prueba:	1.29E-4	1.88E-4	6.47E-5	2.01E-4	6.01E-5	4.05E-5	2.38E-5	5.48E-4

Tabla 4. Diferencia de desviación estándar entre la primera y última generación de cada variable.

	PSO		FPA		LNFE	
hiperparámetro	**Google**	**Nike**	**Google**	**Nike**	**Google**	**Nike**
Cl_s	0.23	0.26	0.19	0.13	0.26	0.23
H_s	0.27	0.25	0.32	0.34	0.29	0.28
W_s	0.29	0.26	0.10	0.09	0.25	0.23
α	0.31	0.28	0.12	0.20	0.27	0.33
B_s	0.23	0.20	-0.03	-0.03	0.17	0.18
E_s	0.29	0.27	0.04	0.12	0.21	0.30

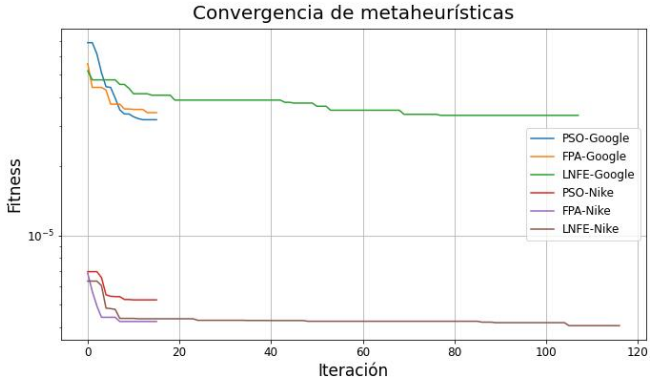


Fig. 4. Convergencia (fitness vs. Iteración) de la optimización de hiperparámetros de la red LSTM.

LSTM-OSM, similar a lo que refiere [1] quién asegura que encontrando las hiperparámetros adecuados se puede obtener la misma precisión a una red más compleja, e incluso se supera el rendimiento, como se aprecia en en este caso en el diagrama de caja y bigote de la Figura 6 y en la gráfica de entrenamiento de la Figura 7.

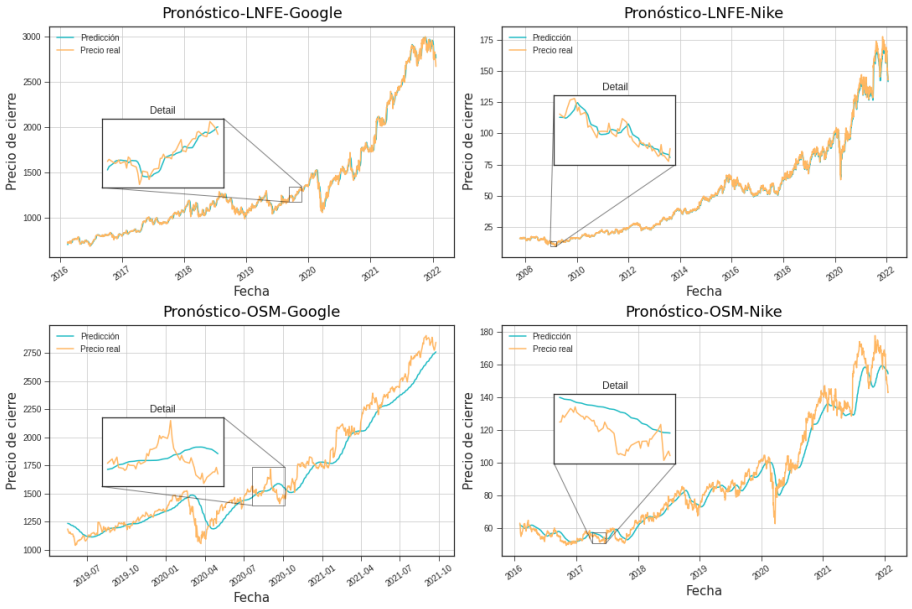


Fig. 5. Pronóstico promedio de 31 experimentos de LNFE y OSM.

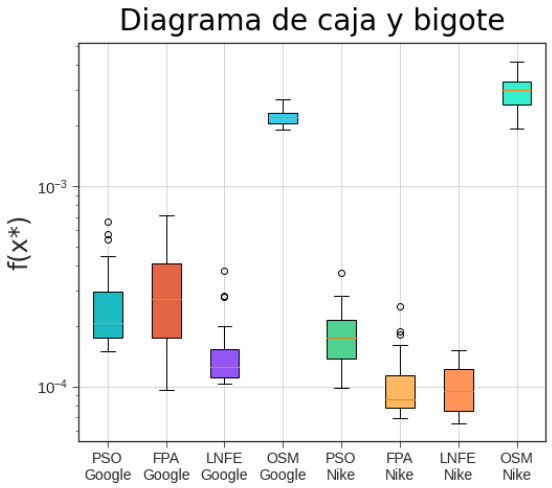


Fig. 6. Diagrama de caja y bigote de prueba de 31 experimentos.

6. Conclusiones

Las topologías que retornaron cada una de las metaheurísticas fueron similares, a excepción de PSO-Nike, sin embargo esto da indicios de que se puede usar una misma arquitectura tanto para Nike como Google, y usar con menor cantidad de Cl_s, usando más estados ocultos, y diferentes configuraciones de hiperparámetros; de la misma manera en el caso de α que se repite en todos a excepción de FPA-Google.

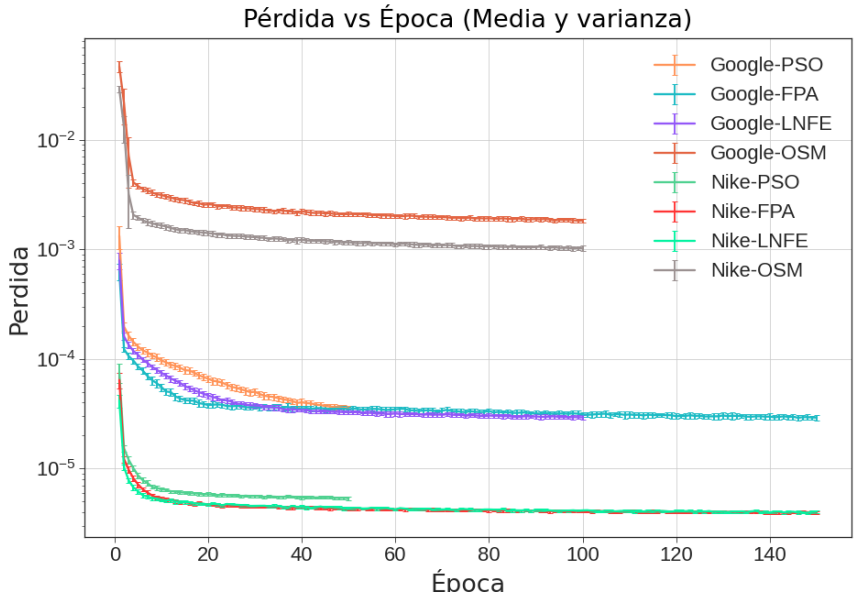


Fig. 7. Media y varianza de 31 experimentos de entrenamiento de la red optimizada con PSO, FPA, LNFE y OSM.

Mientras que LNFE retorna soluciones iguales tanto para Google como Nike, y convergencia claramente definida en una menor cantidad de llamadas a función logrando rendimientos similares a PSO-Google, lo cual sugiere continuar probando con más acciones y diferentes EDAs, por ejemplo, CMA-ES.

Por otro lado la convergencia de LNFE puede aumentar su dispersión en la población final, empero con soluciones consistentes, pues el mismo algoritmo guarda la mejor solución aunque se reinicie su población, por lo cual se puede correr el algoritmo con una menor cantidad de llamadas a función e iteraciones obteniendo resultados satisfactorios.

En referente a la metodología para optimizar la metaheurística, muestra resultados satisfactorios. Se sugiere probar la metodología fijando el número de E_s, y reduciendo el conjunto de entrenamiento, siguiendo un proceso similar de optimización Random Search, pero automatizado; de tal modo que se puede reducir el tiempo de búsqueda, y al final el resultado solo entrenarlo con un mayor número de E_s y con un conjunto de entrenamiento mayor.

Es menester puntualizar que mantener a los algoritmos trabajando en el espacio de los continuos y luego mapear cada solución a números enteros da resultados satisfactorios. Se sugiere experimentar con un número mayor de acciones, para saber si la arquitectura se repite, quizás en tales casos solo será necesario usar una LNFE, ya que implica menor costo computacional.

Por último, usar gráficas de convergencia permite un análisis más detallado del comportamiento de los algoritmos para saber que algoritmo usar para determinado caso, en el presente, pronósticos.

Agradecimientos. Este trabajo es parcialmente financiado por CONACYT de México a través: Beca de Posgrado No. 774627 (F. Pedroza) y Proyecto CÁTEDRAS-2598 (A. Rojas).

Referencias

1. Aufa, B. Z., Suyanto, S., Arifianto, A.: Hyperparameter setting of LSTM-based language model using grey wolf optimizer. In: International conference on data science and its Applications (ICoDSA). pp. 1–5. IEEE (2020) doi: 10.1109/ICoDSA50139.2020.9213031
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. Advances in neural information processing systems, vol. 24 (2011)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research, vol. 13, no. 2 (2012)
4. Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., Oliveira, A. L.: Computational intelligence and financial markets: A survey and future directions. Expert systems with applications, vol. 55, pp. 194–211 (2016) doi: https://doi.org/10.1016/j.eswa.2016.02.006
5. Chen, S., Wu, J., Chen, X.: Deep reinforcement learning with model-based acceleration for hyperparameter optimization. In: IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). pp. 170–177. IEEE (2019) doi: 10.1109/ICTAI.2019.00032
6. Chourmouziadis, K., Chatzoglou, P. D.: An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. Expert Systems with Applications, vol. 43, pp. 298–311 (2016) doi: 10.1016/j.eswa.2015.07.063
7. Evans, C., Pappas, K., Xhafa, F.: Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation. Mathematical and Computer Modelling, vol. 58, no. 5–6, pp. 1249–1266 (2013) doi: 10.1016/j.mcm.2013.02.002
8. Filomeno-Coelho, R., Xiao, M., Guglielmetti, A., Herrera, M., Zhang, W.: Investigation of three genotypes for mixed variable evolutionary optimization. In: Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences, pp. 309–319. Springer, Cham (2015)
9. Gendreau, M., Potvin, J.-Y.: Metaheuristics in combinatorial optimization. Annals of Operations Research, vol. 140, no. 1 (2005) doi: 10.1007/s10479-005-3971-7
10. Goldberg, Y.: Neural network methods for natural language processing. Synthesis lectures on human language technologies, vol. 10, no. 1, pp. 1–309 (2017) doi: 10.2200/S00762ED1V01Y201703HLT037
11. Gorgolis, N., Hatzilygeroudis, I., Istenes, Z., Gyenne, L. G.: Hyperparameter optimization of LSTM network models through genetic algorithm. In: 10th International Conference on Information, Intelligence, Systems and Applications (IISA). pp. 1–4. IEEE (2019) doi: 10.1109/IISA.2019.8900675
12. Kumar, G., Jain, S., Singh, U. P.: Stock market forecasting using computational intelligence: A survey. Archives of Computational Methods in Engineering, vol. 28, no. 3, pp. 1069–1101 (2021) doi: doi.org/10.1007/s11831-020-09413-5

13. Kumar, K., Haider, M., Uddin, T.: Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network. New Generation Computing, vol. 39, no. 1, pp. 231–272 (2021) doi: 10.1007/s00354-020-00104-0
14. Li, A. W., Bastos, G. S.: Stock market forecasting using deep learning and technical analysis: a systematic review. IEEE, vol. 8, pp. 185232–185242 (2020) doi: 10.1109/ACCESS.2020.3030226
15. Li, W., Ng, W. W., Wang, T., Pelillo, M., Kwong, S.: Help: An LSTM-based approach to hyperparameter exploration in neural network learning. Neurocomputing, vol. 442, pp. 161–172 (2021) doi: 10.1016/j.neucom.2020.12.133
16. Moghar, A., Hamiche, M.: Stock market prediction using LSTM recurrent neural network. Procedia Computer Science, vol. 170, pp. 1168–1173 (2020) doi: 10.1016/j.procs.2020.03.049
17. Mora, K. M. F., Marín, J. A., Cerda, J., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., Olvera-López, J. A.: Pattern Recognition: 12th Mexican Conference, MCPR'20, vol. 12088. Springer Nature (2020)
18. Ozbayoglu, A. M., Gudelek, M. U., Sezer, O. B.: Deep learning for financial applications: A survey. Applied Soft Computing, vol. 93, pp. 106384 (2020) doi: 10.1016/j.asoc.2020.106384
19. Qu, Y., Zhao, X.: Application of LSTM neural network in forecasting foreign exchange price, vol. 1237, no. 4, pp. 042036 (2019) doi: 10.1088/1742-6596/1237/4/042036
20. Schmidhuber, J., Hochreiter, S.: Long short-term memory. Neural Comput, vol. 9, no. 8, pp. 1735–1780 (1997) doi: 10.1162/neco.1997.9.8.1735
21. Sezer, O. B., Gudelek, M. U., Ozbayoglu, A. M.: Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, vol. 90, pp. 106181 (2020) doi: 10.1016/j.asoc.2020.106181
22. Silva, T. R., Li, A. W., Pamplona, E. O.: Automated trading system for stock index using LSTM neural networks and risk management. In: International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2020) doi: 10.1109/IJCNN48605.2020.9207278
23. Valdez-Peña, S. I., Hernández-Aguirre, A., Botello-Rionda, S.: Approximating the search distribution to the selection distribution in EDAs. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. pp. 461–468 (2009) doi: 10.1145/1569901.1569965
24. Wen, Y., Lin, P., Nie, X.: Research of stock price prediction based on PCA-LSTM model, vol. 790, no. 1, pp. 012109 (2020) doi: 10.1088/1757-899X/790/1/012109
25. Yang, X. S. (ed): Nature-inspired computation and swarm intelligence: Algorithms, theory and applications. Academic Press (2020)

ISSN 1870-4069

Clasificación de la señal de audio cardiaco mediante la transformada de Fourier de tiempo corto y aprendizaje profundo

Leonel Orozco-Reyes[1], Miguel-Ángel Alonso Arévalo[1], Eloísa García-Canseco[2], Roilhi F. Ibarra-Hernández[3]

[1] Centro de Investigación Científica y de Educación Superior de Ensenada, Departamento de Electrónica y Telecomunicaciones, México

[2] Universidad Autónoma de Baja California, Facultad de Ciencias, México

{orozcorl,aalonso}@cicese.edu.mx, roilhi.ibarra@universidaddeensenada.edu.mx eloisa.garcia@uabc.edu.mx

Resumen. La auscultación es una herramienta de diagnóstico no invasiva, de bajo costo y de sencilla implementación, que actualmente provee información importante en el diagnóstico de patologías cardíacas. Con ayuda de la auscultación se obtiene el sonido cardíaco o fonocardiograma, que es el elemento principal del análisis de este trabajo. Los errores de diagnóstico debido a la falta de médicos experimentados y las limitaciones del sistema auditivo humano han llevado al avance en el área de procesamiento digital de señales y el desarrollo de técnicas para el análisis de sonidos cardíacos asistidos por computadora. El presente trabajo tiene como objetivo clasificar señales de fonocardiogramas usando redes neuronales convolucionales. Haciedo uso de la transformada de Fourier de tiempo corto. Para generar matrices de características que mejor representen la señal de audio cardíaco. Las matrices obtenidas serán clasificadas usando redes neuronales convolucionales, en este trabajo se usarán dos arquitecturas de redes neuronales que han demostrado tener un desempeño notable en tareas de clasificación. Estas redes son AlexNet y VGG16. Los resultados obtenidos reflejan un mejor desempeño para la red convolucional AlexNet con una exactitud de 82.2 % sobre otras redes evaluadas.

Palabras clave: Transformada de Fourier de tiempo corto, sonidos cardiacos, redes neuoronales convolucionales.

Cardiac Audio Signal Classification Using the Fourier Transform of Short Time and Deep Learning

Abstract. Auscultation is a non-invasive, low-cost and easy-to-implement diagnostic tool that currently provides important information in the diagnosis of

cardiac pathologies. With the help of auscultation, the heart sound or phonocardiogram is obtained, which is the main element of the analysis of this work. Diagnostic errors due to a lack of experienced clinicians and the limitations of the human auditory system have led to advancement in the area of digital signal prosessing and the development of techniques for computer-aided analysis of heart sounds. The present work aims to classify phonocardiogram signals using convolutional neural networks. Making use of the short-time Fourier transform. To generate feature matrices that best represent the cardiac audio signal. The obtained matrices will be classified using convolutional neural networks, in this work two neural network architectures will be used that have shown to have a remarkable performance in classification tasks. These networks are AlexNet and VGG16. The results obtained reflect a better performance for the AlexNet convolutional network with an accuracy of 82.2% over other evaluated networks.

Keywords: Short-time Fourier transform, heart sounds, networks convolutional neurons.

1. Introducción

De acuerdo a la Organización para la Cooperación y Desarrollo Económicos (OCDE), en México los gastos económicos provocados por la epidemia de enfermedades cardiovasculares y de diabetes representan una amenaza para el futuro del país y para la viabilidad del sistema público de salud [20]. Además, según las estadísticas reportadas por el Instituto Nacional de Estadística y Geografía (INEGI), en México las enfermedades cardiovasculares son la primer causa de mortalidad en el país, las cuales provocaron más de 300 mil decesos en 2020 [12].

A nivel mundial la situación es similar ya que las enfermedades cardiovasculares siguen siendo la principal causa de morbilidad y mortalidad en todo el mundo, con un estimado de más 20 de millones de personas que fallecieron por enfermedades relacionadas con males cardiovasculares en 2020, lo que representa aproximadamente un tercio de todas las muertes a nivel mundial [21].

La auscultación es un método de diagnóstico primario de patologías cardiacas por medio del análisis de sonidos cardiacos. Esta técnica tiene su origen en el uso del tacto y la aplicación del oído en el tórax de los pacientes y no fue hasta la invención del estetoscopio por René Laënnec en 1819 cuando la auscultación como práctica clínica se estableció y diseminó [11]. Existen, además de la auscultación, otros métodos más sofisticados para la detección de patologías cardíacas, tales como la ecocardiografía y la resonancia magnética.

Sin embargo, la auscultación cardíaca persiste debido a ser una técnica de diagnóstico primario, es sencillla, no invasiva y de muy bajo costo. Mediante la auscultación se proporciona una perspectiva de apoyo al médico para conocer de manera inmediata el estado de un paciente y a decidir si es necesario realizar otra prueba más sofisticada. Cardiólogos experimentados pueden incluso distinguir con gran precisión varios tipos de patologías cardiacas y estimar su severidad utilizando como única herramienta un estetoscopio [17].

Sin embargo, el dominio del estetoscopio para lograr un oído clínico bien entrenado requiere de una larga curva de aprendizaje, provocando que el uso de esta herramienta pueda incluso considerarse como un arte en vías de desaparición. Desde finales del siglo XX, se comenzaron a popularizar los estetoscopios electrónicos (también llamados estetoscopios digitales).

Estos aparatos facilitan considerablemente la adquisición digital de los sonidos producidos por el corazón. En la literatura especializada, a la forma de onda de los sonidos producidos por el corazón se le conoce como fonocardiograma (FCG) o también audio cardiaco. Aunque tanto el FCG como el electrocardiograma (ECG) describen la actividad cardiaca, ambos trabajan en dominios diferentes [1].

La señal de ECG es producida por la actividad eléctrica del corazón, mientras que la señal de FCG es producida por la actividad mecánica del corazón. Además, debido a la diferente naturaleza del origen de estas señales, la existencia de un problema en una señal FCG no implica necesariamente la existencia del mismo u otros problemas en el ECG correspondiente. Por estas razones, la mayoría de los algoritmos basados en la señal de ECG no se pueden aplicar directamente al FCG [25].

1.1. Antecedentes

Generalmente, una señal de FCG normal consta de dos sonidos cardíacos fundamentales llamados primer sonido cardíaco (S1) y segundo sonido cardíaco (S2), que se generan debido al cierre de las válvulas auriculoventriculares y semilunares, respectivamente [1]. El FCG es una señal no estacionaria, es decir una señal cuyas propiedades y estadísticas cambian en función del tiempo.

El intervalo desde el punto inicial de S1 hasta el punto de partida de S2 se denomina intervalo de sístole, y el intervalo desde el punto inicial de S2 hasta el punto inicial de S1 se denomina intervalo de diástole. En la Fig. 1-a) se presenta una señal de FCG normal y la Fig. 1-b) ilustra una señal de FCG patológica correspondiente a un daño en la válvula mitral, es visible en la forma de onda el murmullo asociado a este evento.

El intervalo de diástole suele ser más largo que el intervalo de sístole. Los sonidos cardiacos S1 y S2 suelen tener una duración entre 40 y 150 ms y su contenido espectral se encuentra principalmente en el rango de 20 a 150 Hz [2]. En el caso de las patologías (chasquidos, fricciones y murmullos), su duración varía considerablemente dentro del ciclo cardiaco y su contenido espectral se encuentra en el rango de 25 a 700 Hz [5]. Durante la última década han sido propuestos un gran número de trabajos que buscan llevar a cabo la clasificación automática del FCG.

Una revisión exhaustiva de la literatura sobre el análisis y la clasificación de los sonidos cardiacos está fuera del alcance de este artículo, pero esta puede ser consultada en [4, 8]. Una tendencia predominante en el análisis de la señal de FCG es la realizar primeramente una etapa de segmentación. Es decir, localizar S1 y S2, antes de proceder a la clasificación de los sonidos [8].

Con respecto a las técnicas de clasificación, las primeras propuestas se basaron en análisis estadísticos y métodos clásicos de aprendizaje de máquinas tales como máquinas de soporte vectorial (SVM), k-vecino más cercano (k-NN), bosques aleatorios, modelos ocultos de Markov o perceptrón multicapa (MLP) [4, 8].

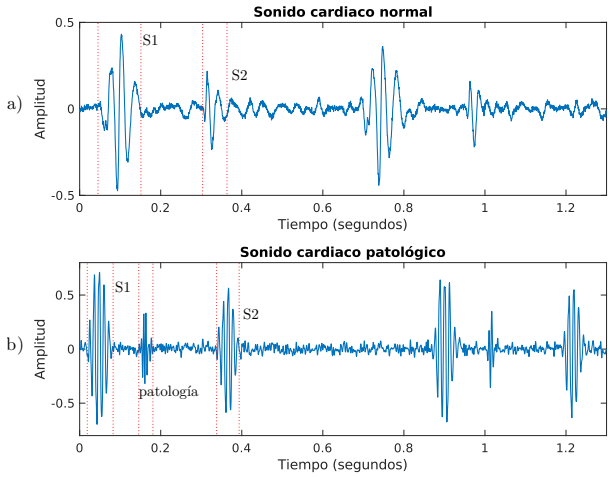


Fig. 1. Dos ejemplos de sonidos cardiacos, en a) se muestra una señal de FCG correspondiente a una persona sin patología, mientras que b) presenta una señal de FGC donde se puede apreciar una patología entre los sonidos S1 y S2.

Estos métodos suelen combinarse con el procesamiento digital de señales para la extracción de características basadas en el dominio del tiempo, en el dominio de frecuencia o en el dominio tiempo-frecuencia [6, 8, 10, 28].

Entre las técnicas de extracción de características las más comunes que han sido previamente utilizadas se encuentran: la transformada rápida de Fourier (FFT), coeficientes cepstrales en escala de frecuencia Mel (MFCC), la Transformada de Fourier de Tiempo Corto (TFTC), la transformada ondeleta discreta (DWT), la transformada ondeleta continua (CWT), la transformada ondeleta a Q-constante (TQWT), la transformada chirplet (CT), la transformada S, Transformada de Hilbert-Huang (HHT), entre otras [8]. En años recientes han aparecido las arquitecturas de aprendizaje profundo (DL), particularmente las de tipo red neuronal convolucional (CNN) y las redes neuronales recursivas (RNN) [4, 8].

Artículos recientes demuestran las ventajas de utilizar técnicas de análisis tiempo-frecuencia y métodos de DL para lograr la detección eficaz de patologías cardíacas únicamente utilizando la señal de FCG [6, 8, 10, 28]. En el presente trabajo se propone utilizar el análisis tiempo–frecuencia del FCG combinado con métodos de clasificación supervisada basados en DL.

Específicamente, se propone utilizar la Transformada de Fourier de Tiempo Corto (TFTC) para convertir la señal de FCG en una señal bidimensional la cual es clasificada por una CNN. Las CNNs que proponemos utilizar han sido específicamente diseñadas para clasificación de imágenes y tienen un alto desempeño: AlexNet, VGG16 [14, 26].

1.2. Bases de datos utilizadas

Para el desarrollo de este trabajo se propone utilizar dos bases de datos de sonidos cardiacos, ambas están disponibles al público. La primera base es la utilizada en “The PhysioNet/Computing in Cardiology (CinC) Challenge 2016” [7].

Tabla 1. Composición base de datos de sonidos cardiacos Physionet CinC [7].

Base de datos	**Pacientes**	**Grabaciones**	**Sonidos patológicos (%)**	**Sonidos sanos (%)**	**Sin clasificación (%)**
A	121	409	67.5	28.4	4.2
B	106	490	14.9	60.2	24.9
C	31	31	64.5	22.6	12.9
D	38	55	47.3	47.3	5.5
E	356	2054	7.1	86.7	6.2
F	112	114	27.2	68.4	4.4
Total	764	3153	18.1	73.0	8.8

Tabla 2. Composición de la base de datos de Yaseen et. al. [30].

Tipo	**Clase**	**Número de grabaciones por clase**
Normal	N	200
Patológico	AS	200
	MR	200
	MS	200
	MVP	200
Total		1000

Esta colección de sonidos proviene de siete grupos independientes de investigación. La base de datos contiene 3,153 grabaciones y un total de 233,512 sonidos cardiacos recolectados de pacientes sanos y pacientes con diferentes patologías cardiacas, como enfermedades de las válvulas del corazón y enfermedades de la arteria coronaria [7], entre otras.

La composición de la base de datos se presenta en la Tabla 1. La segunda base de datos utilizada es la propuesta por [30]. También está compuesta de sonidos patológicos y sonidos sanos. Los sonidos patológicos a su vez están divididos en cuatro categorías que corresponden a las siguientes enfermedades: estenosis aórtica (AS), estenosis mitral (MS), regurgitación mitral (MR) y prolapso de la válvula mitral (MVP). La composición de la base de datos se presenta en la Tabla 2.

Los sonidos de estas bases de datos han sido digitalizados usando diferentes frecuencias de muestreo y 16 bits de resolución. Para este trabajo, se remuestrearon la señales, se uniformizó la frecuencia de muestreo a $Fs = 2000$ Hz con 16 bits de resolución y después se usó un filtro pasa-banda Butterworth con frecuencias de corte de 25 Hz y 900 Hz. Los sonidos de la base de datos de PhysioNet que han sido marcados como no aptos para clasificación no fueron utilizados en este trabajo.

2. Metodología

En la Fig. 2 se describe a grandes rasgos el funcionamiento del método de clasificación propuesto. Utilizando únicamente la forma de onda de los sonidos cardiacos, primeramente se calcula el espectrograma por medio de la transformada de Fourier.

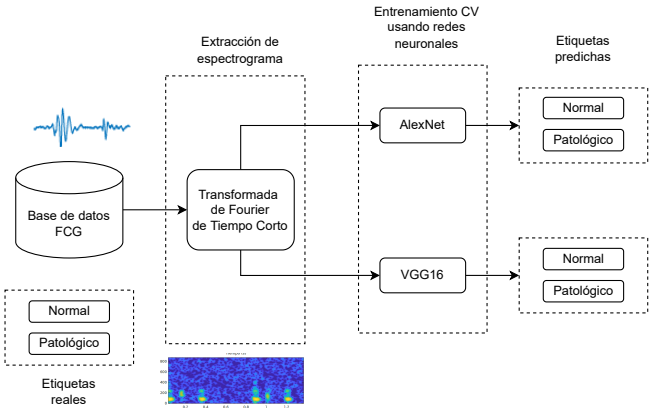


Fig. 2. Proceso de la clasificación de la señal de fonocardiograma usando la TFTC y redes neuronales convolucionales.

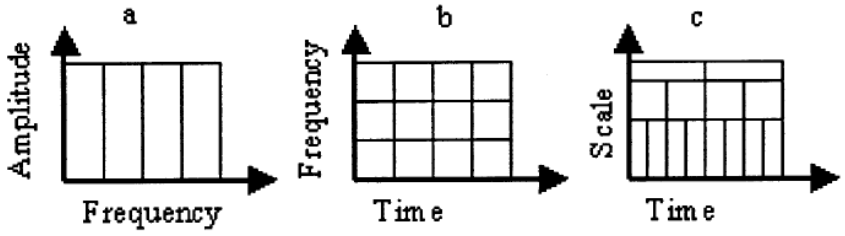


Fig. 3. Las representaciones usadas son (a) transformada de Fourier, (b) trasformada de Fourier de tiempo corto, y (c) transformada ondeleta. Imagen tomada de [19].

Después, el espectrograma es analizado por medio de una red neuronal convolucional que estima si la señal corresponde a un sonido cardiaco normal o a un sonido patológico.

2.1. Transformada de Fourier de tiempo corto

En procesamiento digital de señales es de gran utilidad considerar la dualidad tiempo–frecuencia que tienen las señales, ya que al representarlas en un dominio o en el otro se obtiene información distinta y complementaria a la vez. La herramienta más común para conocer el contenido frecuencial de una señal continua $x(t)$ en el dominio del tiempo es la Transformada de Fourier (TF), la cual puede ser definida como:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t}\, dt, \tag{1}$$

donde $j = \sqrt{-1}$ y $X(\omega)$ puede considerarse como una medida de cuánto oscila $x(t)$ a la frecuencia angular ω [18].

Este operador matemático, sin embargo, tiene una gran limitante: el tiempo y la frecuencia son excluyentes entre sí. Es decir, al representar una señal en uno de los dominios se pierde la información sobre el otro [3]. De esta manera, al analizar mediante la TF señales de tipo no estacionarias o señales cuyas características en frecuencia varían en función del tiempo, como sucede con el FCG, no sería posible identificar los instantes de tiempo donde se presenten ciertos componentes determinados de la señal; como son S1, S2 y los sonidos patológicos.

Tradicionalmente la forma de solventar esta limitación ha sido mediante una representación en ambos dominios al mismo tiempo. Específicamente, en la transformación de una señal unidimensional en el dominio del tiempo a una bidimensional donde se incorporen tiempo y frecuencia [3]. A este tipo de transformación se les denomina representación tiempo–frecuencia (RTF) [3].

Existen una gran variedad de RTFs, entre los métodos lineales uno de los más utilizados por su sencillez y fácil implementación es la llamada Transformada de Fourier de Tiempo Corto (TFTC o STFT por sus siglas en inglés). En comparación con otras herramientas como la transformada ondeleta es que se puede tener una mayor facilidad en la interpretación con la TFTC, esto puede ser visto en la Figura 3.

La mayoría de los resultados obtenidos usando ondeletas pueden ser obtenidos de la misma manera usando la TFTC [13]. Cuando se usa la transformada ondeleta es común tener dificultades al extraer los componentes fundamentales o cualquier otro componente armónico de la señal. Aunque la TFTC tiene una resolución fija en todas las frecuencias, una vez el tamaño de la ventana está dado, permite una interpretación más fácil en términos de las armónicas [13]. La TFTC se calcula a partir de la traslación (τ) y modulación (ω) de una ventana de análisis $w(t)$ [18]:

$$X(t,\omega) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-j\omega t}\,dt, \tag{2}$$

donde $w(t) \in \mathbb{R}$, es simétrica y además $\|w_{t,\omega}\| = 1$. Al calcular la TFTC se obtiene un resultado complejo, i.e., $X(t,\omega) \in \mathbb{C}$. En nuestro caso nos interesa particularmente la distribución de la energía en función de la frecuencia y del tiempo, a este parámetro se le conoce como el espectrograma y está definido como $S(\tau,\omega) = |X(\tau,\omega)|^2$ [3, 18]. En la práctica, la señal de FCG, $x(n)$, que utilizamos es de tiempo discreto y puede ser vista como la versión muestreada de $x(t)$ cada nT instantes de tiempo, donde $n \in \mathbb{Z}$ y T es el periodo de muestreo. La versión discreta de la TFTC [27] puede escribirse como:

$$X(m,\omega) = \sum_{n=-\infty}^{+\infty} x(n)w(n-mR)e^{-j\omega n}, \tag{3}$$

donde $w(n)$ es la versión discreta de la ventana de análisis mencionada anteriormente y tiene una longitud de L_w muestras, m es el índice de los intervalos de tiempo correspondientes a cada segmento de la ventana de análisis, R es el número de muestras de traslape entre el instante m y el instante $m+1$. La Fig. 4 ilustra la manera en que se calcula la TFTC por medio de la Transformada Rápida de Fourier (FFT), suponiendo que se calcula una transformada de longitud par de N puntos. El espectrograma se obtiene calculando la expresión $S(m,\omega) = |X(m,\omega)|^2$.

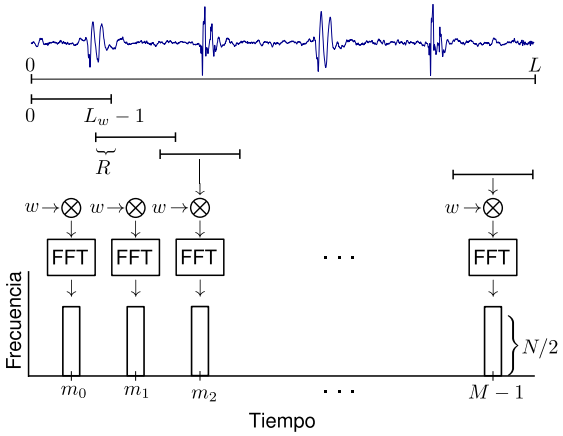


Fig. 4. Representación gráfica de cómo se calcula la TFTC discreta a partir de la FFT. El resultado del cálculo es una matriz compleja de dimensiones $\frac{N}{2} \times M$.

Los parámetros de implementación de la TFTC en este trabajo son los siguientes: se procesaron bloques de FCG de $L = 2800$ muestras, usando una ventana de análisis $w(n)$ de tipo Hamming de longitud $L_w = 100$ muestras, un traslape de $R = 88$ muestras y la longitud de la FFT de $N = 512$ muestras. Finalmente, de la matriz de TFTC se calcula el valor absoluto y se eleva al cuadrado para obtener el espectrograma $S(\tau, \omega)$, el cual tiene dimensiones 257×224.

Dado que las arquitecturas de CNN que se propone utilizar en este trabajo requieren una imagen de entrada de dimensiones distintas a la de $S(\tau, \omega)$, el espectrograma fue recortado para ajustarlo al tamaño 224×224. En este caso se eliminan las filas 225 a la 257, es decir, el análisis posterior únicamente considera frecuencias de hasta ≈871 Hertz. La Fig. 5-a) y b) ilustran el tipo de imagen utilizada para detectar la presencia o ausencia de patologías cardiacas.

La ventaja que presenta la metodología descrita es que no se requiere conocer de antemano la segmentación de la señal cardiaca. Es decir, a diferencia de otras propuestas que existen en la literatura [8], este método no requiere conocimiento de la posición exacta de los sonidos S1, S2 o los silencios de sístole y diástole, para llevar a cabo la detección.

3. Clasificación de las imágenes de sonidos cardiacos mediante aprendizaje profundo

El aprendizaje de máquinas clásico busca que las computadoras puedan actuar con poca intervención humana. Por otro lado, el aprendizaje profundo se trata de que las computadoras aprendan a actuar usando estructuras inspiradas en el cerebro humano y han demostrado que requieren todavía menos intervención humana. Gracias al poder computacional de los dispositivos de cómputo modernos es posible implementar las redes neuronales convolucionales (CNN) para tareas de clasificación relacionadas con la salud.

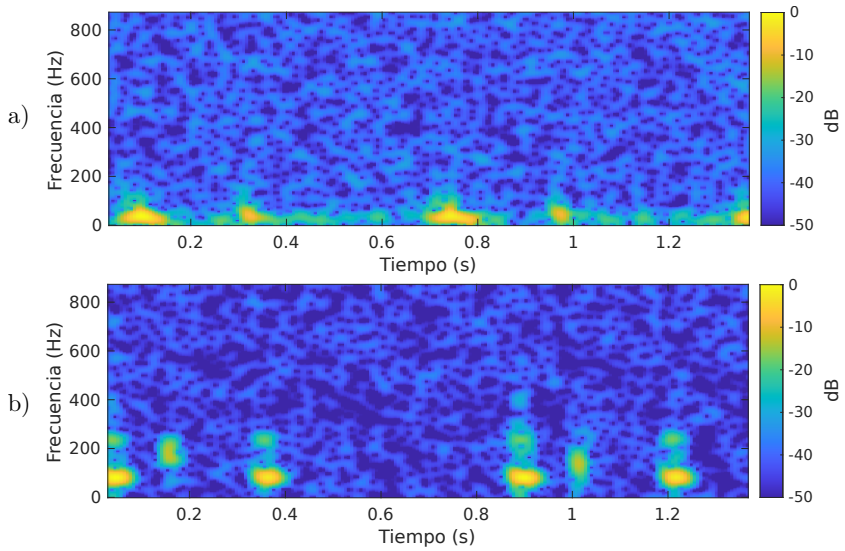


Fig. 5. Ejemplo de espectrograma obtenido de los sonidos cardíacos presentados en la Fig. 1, a) para un sonido cardiaco normal y b) para un sonido cardiaco patológico.

En el caso de este trabajo, la finalidad es desarrollar un clasificador de sonidos cardíacos que tenga un alto nivel de confiabilidad en la detección de patologías.

3.1. Aprendizaje profundo

El aprendizaje profundo es un método de representación-aprendizaje con múltiples niveles de representaciones, obtenidas con la descomposición de módulos simples, pero no lineales que transforman la representación de un nivel bajo a un nivel alto y más abstracto. En esencia son redes neuronales compuestas de tres capas o más usadas para tareas de clasificación, las representaciones de las capas de alto nivel amplifican aspectos de la entrada que son importantes para la discriminación y supresión de variables irrelevantes [15].

Para comprender el funcionamiento de las redes neuronales, estas pueden verse como una versión simplificada del cerebro humano. Las neuronas están organizadas en capas, cada neurona recolecta información de la capa anterior, realiza un cálculo simple y comunica el resultado a la siguiente capa. En las redes más eficientes se pueden tener docenas de capas, por lo que el modelo puede ser llamado de aprendizaje profundo [22]. Una introducción detallada a los conceptos y aplicación del aprendizaje profundo y las CNNs está disponible en [9].

El elemento más importante de las CNN son las capas convolucionales. Se puede entender que una convolución es la aplicación de un filtro, también llamado kernel, a una entrada que da como resultado una activación. La aplicación sistemática del mismo filtro en la entrada resulta en un mapa de activaciones o también llamado mapa de características, que indican la localización y magnitud de una característica detectada en la entrada de los datos.

Las CNN tienen la habilidad de aprender una gran cantidad de filtros que en conjunto pueden llegar a resolver problemas complejos [24]. Esto deriva en su utilidad en tareas de clasificación de imágenes con una capacidad que incluso puede mejorar la pericia de los seres humanos.

Por lo que haciendo uso de las representaciones obtenidas a partir de los sonidos cardíacos, se detectará la presencia de anomalías cardíacas. Las CNNs que fueron seleccionadas para el presente trabajo son AlexNet y VGG16 [14, 26].

Estas redes han sido específicamente diseñadas para clasificación de imágenes y presentan un alto desempeño. El modelo de AlexNet está compuesto por una arquitectura de ocho capas, las primeras cinco son capas convolucionales y las últimas tres son capas densas, fue la primer red en cambiar la función de activación de sigmoide por una de tipo ReLu.

Originalmente fue entrenada con 1.2 millones de imágenes, para evitar el sobre-entrenamiento se implementaron técnicas de Data Augmentation y capas de Dropout. Esta red participó y ganó el reto de ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC), rompiendo así el paradigma de que las características usadas para los filtros tenían que ser hechas a mano.

El modelo de CNN llamado VGG16 también es una arquitectura diseñadas para clasificación de imágenes, la cual ganó la competencia ILSVRC en 2014. Esta red está compuesta por trece capas convolucionales y tres capas densas. La abreviación VGG significa Visual Geometry Group, en referencia al grupo de investigadores que la propuso. A su vez, el número 16 indica el número de capas que la compone [29, 26].

Los modelos de CNN seleccionados fueron diseñados para clasificar cientos de diferentes tipos de imágenes y utilizan matrices de entrada de tres canales para tomar en cuenta los colores rojo, verde y azul. Como se explicó en la sección anterior, las imágenes creadas a partir de los espectrogramas de los sonidos cardiacos tienen solamente un canal.

Por esta razón fue necesario modificar ligeramente las redes en lo que respecta a las capas de entrada y la de salida. Específicamente, las modificaciones que se realizaron fueron la adaptación de la capa de entrada para recibir imágenes de tamaño $224 \times 224 \times 1$ y en la capa de salida se modificó a dos posibles resultados, la predicción de que se trate de una imagen que corresponde a un sonido patológico o a un sonido normal.

Para el entrenamiento de la red se utilizaron un total de 10,200 imágenes, estas fueron balanceadas con una relación 1:1 entre imágenes provenientes de sonidos normales y patológicos. Los hiperparámetros usados fueron: 150 épocas para el entrenamiento, así como el optimizador Stochastic Gradient Descent (SGD) con una tasa de aprendizaje de $0{,}008$, la función de pérdida categorical_crossentropy y un tamaño de lote de 64 muestras.

Estas actividades se realizaron bajo el ambiente virtual en la plataforma de Google Colaboratory en su versión de pago. Las características del sistema utilizado para realizar las pruebas son las siguientes: procesador Intel(R) Xeon(R) @ 2.3GHz, memoria RAM de 26 GB y una tarjeta gráfica Tesla P100 de 16 GB. Los resultados del entrenamiento se presentan en la Fig. 6.

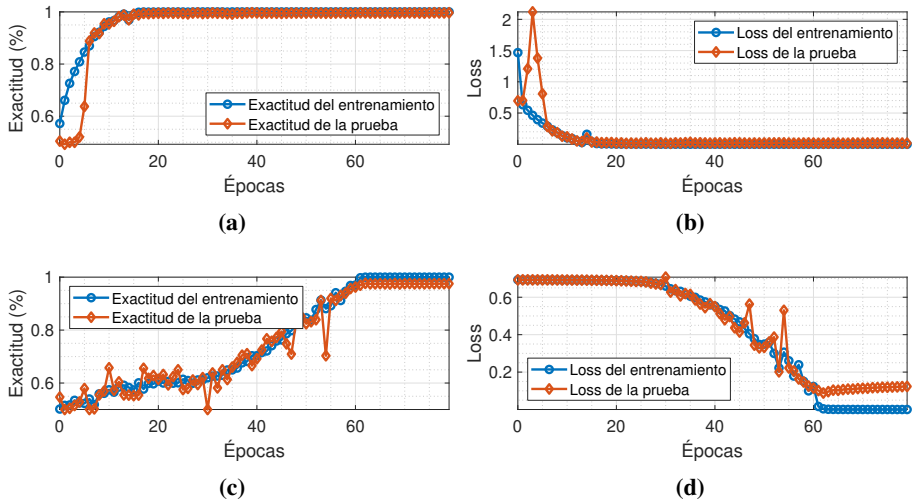


Fig. 6. Exactitud de AlexNet (a) y VGG16 (c); y Loss de AlexNet (b) y VGG16 (d) durante el entrenamiento. Sólo se muestran las primeras 80 épocas para mejorar la visualización de los resultados.

4. Resultados

En esta sección evaluamos el rendimiento de la metodología propuesta con respecto a su capacidad para distinguir con precisión entre sonidos cardiacos normales y patológicos. Se hará uso de la validación cruzada, que es un método estadístico para la evaluación de del rendimiento de un modelo de aprendizaje automático. Este consite en dividir los datos en dos partes, la primera se usará para entrenar al modelo y la segunda se usará para la evaluación del modelo.

Entre los diferentes tipos de validación cruzada, se eligió la validación cruzada de K iteraciones, el valor de K fue establecido a 10, por lo que se entiende que cada iteración era entrenada usando el 90 % de los datos y se usaban 10 % de los datos para la validación, este proceso de entrenamiento/evaluación fue repetido 10 veces. Entre las ventajas de este tipo de validación cruzada está una estimación precisa del rendimiento del modelo, así como evitar el sobre-entrenamiento [23].

La obtención de la exactitud, precisión, sensibilidad, y especifidad serán usados para la evaluación del rendimiento de los modelos usados. La exactitud (Accuracy) hace referencia a lo cerca que está el resultado de una medición del valor verdadero. La sensitividad (Sensitivity) mide la proporción de sonidos patológicos que fueron correctamente identificadas como positivos.

La especificidad (Specificity) mide la proporción de sonidos normales que fueron correctamente identificados como negativos. La precisión (Precision) hace referencia a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Estos valores están detallados en la Tabla 3. Se puede observar que los mejores resultados fueron obtenidos por el modelo de AlexNet con una exactitud de 84.2 %.

Tabla 3. Métricas de la matriz de confusión de las imágenes espectrograma.

	Exactitud	**Precisión**	**Sensibilidad**	**Especificidad**
AlexNet	0.828	0.842	0.807	0.848
VGG16	0.746	0.755	0.729	0.763

Tabla 4. Matrices de confusión de los modelos AlexNet y VGG16 en la clasificación de las imágenes espectrograma.

	Verdadero negativo	**Falso positivo**	**Falso negativo**	**Verdadero positivo**
AlexNet	865	155	197	823
VGG16	778	242	276	744

Para poder comprender de mejor manera los resultados obtenidos, se hará uso de la matriz de confusión. Esta matriz permite analizar el desempeño de un algoritmo de clasificación, decribiendo cómo se distribuyen los valores reales así como las predicciones. Una predicción correcta positiva es un verdadero positivo, una predicción incorrecta positiva es un falso positivo, una predicción correcta negativa es un verdadero negativo y una predicción incorrecta negativa es un falso negativo.

Estos valores están detallados en la Tabla 4. El modelo de AlexNet cuenta con una mayor cantidad de verdaderos negativos en comparación del modelo VGG16 con un 11.18 % más de casos verdaderos negativos. Los falsos positivos del modelo VGG16 son mayores en un 56 % que AlexNet. Los falsos negativos del modelo VGG16 son mayores también en un 40 % que Alexnet.

Finalmente, los casos verdaderos positivos de AlexNet son mayores en un 10 % que el modelo VGG16. Esto nos da un mejor entendimiento de los resultados obtenidos de la Tabla 3, en la que el modelo AlexNet tiene mejor rendimiento al clasificar las patologías cardiacas, al mismo tiempo que tiene menos errores que el modelo VGG16.

Los resultados obtenidos por el algoritmo propuesto se consideran satisfactorios ya que concuerdan con los reportados por otros algoritmos en la página oficial del challenge de PhysioNet realizado en 2016 [7]. Sin embargo, actualmente hay otros algoritmos del estado del arte que reportan valores de exactitud mayores [4, 8].

Es necesario explorar otros métodos de analisis tiempo-frecuencia tales como la Transformada Ondeleta [16, 25] o métodos cuadráticos como la distribución de Wigner-Ville [3, 18, 28]. Con respecto a las CNNs evaluadas, AlexNet exhibe un mejor desempeño que VGG16.

5. Conclusiones y trabajo futuro

En este trabajo se ha propuesto la comparación de dos modelos de redes neuronales convolucionales para la detección de patologías cardíacas usando el espectrograma obtenido a partir de la transformada de Fourier de tiempo corto aplicado a la señal de fonocardiograma. Los resultados obtenidos por el modelo de AlexNet son de una exactitud de 82.8 %, y el modelo VGG16 obtuvo una exactitud de 74.6 %, mostrando que el modelo de AlexNet es más apropiado para llevar a cabo esta clasificación.

La utilidad de este trabajo es resaltar la importancia de una correcta elección de la red neuronal convolucional. Otra ventaja de AlexNet es su complejidad computacional durante la etapa de entrenamiento, ya que es considerablemente más rápida.

En el trabajo futuro sería interesante explorar el uso de otras representaciones tiempo-frecuencia, así como otros modelos de redes neuronales convolucionales adecuados a la clasificación de imágenes médicas.

Los resultados obtenidos por el algoritmo propuesto se consideran buenos ya que concuerdan con los reportados por otros algoritmos sometidos al PhysioNet Challenge CinC 2016 [7]. Sin embargo, actualmente existen algoritmos que reportan mejor desempeño [4, 8].

Referencias

1. Abbas, A. K., Bassam, R.: Phonocardiography signal processing. Synthesis Lectures on Biomedical Engineering, vol. 4, no. 1, pp. 1–194 (2009) doi: 10.1007/978-3-031-01637-0
2. Arnott, P. J., Pfeiffer, G. W., Tavel, M. E.: Spectral analysis of heart sounds: relationships between some physical characteristics and frequency spectra of first and second heart sounds in normals and hypertensives. Journal of biomedical engineering, vol. 6, no. 2, pp. 121–128 (1984) doi: 10.1016/0141-5425(84)90054-2
3. Boashash, B.: Time-frequency signal analysis and processing: a comprehensive reference. Academic press (2015)
4. Chen, W., Sun, Q., Chen, X., Xie, G., Wu, H., Xu, C.: Deep learning methods for heart sounds classification: A systematic review. Entropy, vol. 23, no. 6, pp. 667 (2021) doi: 10.3390/e23060667
5. Choi, S., Jiang, Z.: Cardiac sound murmurs classification with autoregressive spectral analysis and multi-support vector machine technique. Computers in biology and medicine, vol. 40, no. 1, pp. 8–20 (2010) doi: 10.1016/j.compbiomed.2009.10.003
6. Chowdhury, T. H., Poudel, K. N., Hu, Y.: Time-frequency analysis, denoising, compression, segmentation, and classification of pcg signals. IEEE Access, vol. 8, pp. 160882–160890 (2020) doi: 10.1109/ACCESS.2020.3020806
7. Clifford, G. D., Liu, C., Moody, B., Springer, D., Silva, I., Li, Q., Mark, R. G.: Classification of normal/abnormal heart sound recordings: The PhysioNet/Computing in Cardiology Challenge 2016. Computing in Cardiology, vol. 43, pp. 609–612 (2016) doi: 10.22489/cinc.2016.179-154
8. Dwivedi, A. K., Imtiaz, S. A., Rodriguez-Villegas, E.: Algorithms for automatic analysis and classification of heart sounds–a systematic review. IEEE Access, vol. 7, pp. 8316–8345 (2018) doi: 10.1109/ACCESS.2018.2889437
9. Géron, A.: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media Inc. (2019)
10. Ghosh, S. K., Ponnalagu, R. N., Tripathy, R. K., Acharya, U. R.: Automated detection of heart valve diseases using chirplet transform and multiclass composite classifier with PCG signals. Computers in biology and medicine, vol. 118, pp. 103632 (2020) doi: 10.1016/j.compbiomed.2020.103632

11. Hanna, I. R., Silverman, M. E.: A history of cardiac auscultation and some of its contributors. The American journal of cardiology, vol. 90, no. 3, pp. 259–267 (2002) doi: 10.1016/S0002-9149(02)02465-7
12. INEGI: Características de las defunciones registradas en México durante 2020 (2021)
13. Jurado, F., Saenz, J. R.: Comparison between discrete STFT and wavelets for the analysis of power quality events. Electric Power Systems Research, vol. 62, no. 3, pp. 183–190 (2002) doi: 10.1016/S0378-7796(02)00035-4
14. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017) doi: 10.1145/3065386
15. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, vol. 521, no. 7553, pp. 436–444 (2015) doi: 10.1038/nature14539
16. Lu, J., Jiang, Q., Li, L.: Analysis of adaptive synchrosqueezing transform with a time-varying parameter. Advances in Computational Mathematics, vol. 46, no. 5 (2020) doi: 10.1007/s10444-020-09814-x
17. Mahnke, C. B.: Automated heartsound analysis/computer-aided auscultation: a cardiologist's perspective and suggestions for future development. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 3115–3118. IEEE (2009) doi: 10.1109/IEMBS.2009.5332551
18. Mallat, S.: A wavelet tour of signal processing: The sparse way (2009)
19. Messer, S. R., Agzarian, J., Abbott, D.: Optimal wavelet denoising for phonocardiograms. Microelectronics Journal, vol. 32, no. 12, pp. 931–941 (2001) doi: 10.1016/S0026-2692(01)00095-7
20. OECD: Obesity update (2017)
21. Organization, W. H.: World health statistics 2022: Monitoring health for the SDGs, sustainable development goals. Tech. rep., World Health Organization (2022)
22. Peyré, G.: Mathematics of neural networks. École Normale Supérieure PSL (2020)
23. Refaeilzadeh, P., Tang, L., Liu, H., Angeles, L., Scientist, C. D.: Cross-validation. Springer New York (2016)
24. Rosebrock, A.: Deep learning for computer vision with python - Starter. Pyimageserach (2017)
25. Safara, F., Doraisamy, S., Azman, A., Jantan, A., Ramaiah, A. R. A.: Multi-level basis selection of wavelet packet decomposition tree for heart sound classification. Computers in biology and medicine, vol. 43, no. 10, pp. 1407–1414 (2013) doi: 10.1016/j.compbiomed.2013.06.016
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, vol. 6, pp. 1–14 (2014) doi: 10.48550/arXiv.1409.1556
27. Smith, J. O.: Spectral audio signal processing. W3K (2011)
28. Vyas, S., Patil, M. D., Birajdar, G. K.: Classification of heart sound signals using time-frequency image texture features. Computational Intelligence and Healthcare Informatics, pp. 81–101 (2021) doi: 10.1002/9781119818717.ch5
29. Wei, J.: AlexNet: The architecture that challenged CNNs. Towards Data Science, (2019)

30. Yaseen, Son, G. Y., Kwon, S.: Classification of heart sound signal using multiple features. Applied Sciences (Switzerland), vol. 8, no. 12, pp. 2344 (2018) doi: 10.3390/app8122344

ISSN 1870-4069

Implementación de CNN basada en una arquitectura VGG16 para detección y clasificación de árboles mediante la segmentación semántica en imágenes aéreas

Laritza Pérez-Enríquez, Raquel Díaz-Hernández,
Leopoldo Altamirano-Robles

Instituto Nacional de Astrofísica Óptica y Electrónica,
Posgrado en Ciencias y Tecnología del Espacio,
México

{laritza,raqueld,robles}@inaoep.mx

Resumen. Factores como, el calentamiento global, la contaminación, la sobre población, y la inconciencia humana, son algunas de las causas que han generado impacto de forma negativa en nuestro planeta, provocando consecuencias irreparables como la extinción de recursos naturales no renovables y poner algunas especies en peligro de extinción. Para combatir esta situación se han implementado situaciones legales que protegen áreas y especies en peligro, sin embargo, no es suficiente con realizar la protección a lo que ya se encuentra en riesgo de extinción, también es necesario tomar medidas que ayuden a impedir que más recursos naturales lleguen a estas circunstancias. La segmentación semántica es una técnica que se ha aplicado para la selección de píxeles pertenecientes a una clase, mediante la aplicación de algoritmos de redes neuronales convolucionales (CNN), los cuales son una herramienta que permite realizar mediante entrenamiento con conjuntos de datos el reconocimiento, identificación y selección de objetos. En este trabajo se propone aplicar una red neuronal convolucional basada en una arquitectura VGG16 para realizar segmentación semántica en árboles de palmeras, pinos y naranjos, con la intención de replicar esta técnica en otros tipos de especies de árboles aunado finalmente a la aplicación de la técnica de transferencia de aprendizaje (TL) implementada para mejorar los resultados de segmentación de forma exitosa y eficiente alcanzando una precisión del 94 % a 96.5 %.

Palabras clave: Percepción remota, CNN, VGG16, segmentación semántica, transferencia de aprendizaje.

Implementation of CNN based on a VGG16 Architecture for Detection and Classification of Trees through Semantic Segmentation in Aerial Images

Abstract. Factors such as global warming, pollution, overpopulation, and human unconsciousness are some of the causes that have generated a negative

impact on our planet, causing irreparable consequences such as the extinction of non-renewable natural resources and putting some species in danger. Danger of extinction. To combat this situation, legal situations have been implemented that protect endangered areas and species, however, it is not enough to protect what is already at risk of extinction, it is also necessary to take measures that help prevent more resources natural come to these circumstances. Semantic segmentation is a technique that has been applied to the selection of pixels belonging to a class, through the application of convolutional neural network (CNN) algorithms, which are a tool that allows training with data sets to perform recognition, identification and selection of objects. In this work, it is proposed to apply a convolutional neural network based on a VGG16 architecture to perform semantic segmentation in palm, pine and orange trees, with the intention of replicating this technique in other types of tree species, finally coupled with the application of the transfer learning (TL) technique implemented to improve segmentation results successfully and efficiently reaching an accuracy of 94

Keywords: Remote sensing, CNN, VGG16, semantic segmentation, transfer learning.

1. Introducción

Actualmente, resulta difícil poder realizar inspección de campo de zonas rurales, debido al poco o nulo acceso en algunas de estas áreas. Es mediante recorridos que se puede obtener información y datos para estadísticas de las especies vegetales de la zona y tener un registro que se encuentre disponible para un control eficiente de estás, que en su momento, se encuentren en peligro de extinción o puedan llegar a estarlo.

Hoy en día, el uso de técnicas como la inspección aérea mediante vehículos aéreos no tripulados (VANT) está revolucionando y facilitando la forma de obtener las imágenes de estas zonas. Para realizar la identificación y clasificación de especies se han aplicado, también, técnicas de Inteligencia Artificial (IA). La IA ha evolucionado en los últimos años, y se ha aplicado en distintas áreas tales como: minería espacial, medicina, vehículos autónomos, entre otras.

Con la aplicación de CNNs [1, 2, 3, 5, 6, 7] se ha logrado principalmente realizar análisis de imágenes en cuyas escenas se puede detectar, identificar y clasificar todo tipo de objetos y formas para los cuales la red es entrenada. Estas redes neuronales, son implementadas mediante distintas arquitecturas, las cuales se encuentran formadas por numerosas estructuras y capas convolucionales que es lo que principalmente las hace diferentes una de otra, sin embargo, siguen un mismo objetivo, el detectar el objeto deseado.

No obstante es mediante la técnica de segmentación que se puede lograr obtener y visualizar las clases de objetos de interés [1, 3, 5, 6, 7], ya que la segmentación realizará un agrupamiento en los píxeles característicos que constituyen la forma de la clase indicada. Otro aspecto relevante, es que las arquitecturas pueden ser reentrenables, y además, se puede aplicar técnicas que ayuden a obtener mejor precisión en los resultados de segmentación como se muestra en [2, 3, 5, 8].

Algunos trabajos como [2, 6] implementan la arquitectura VGG16 para detección y selección mediante la aplicación de la técnica de transferencia de aprendizaje, con la cual logran utilizar parámetros adquiridos y aplicarlos para identificar otros objetos. Recientemente métodos basados en una red totalmente convolucional (FCN) realizaron un gran progreso en la segmentación semántica [1, 6, 7], para la clasificación de imágenes, utilizan características que no se encuentran en métodos tradicionales, sin embargo, la aplicación de estas técnicas para segmentación semántica a nivel de píxel trae como consecuencia un alto costo computacional.

Métodos como la segmentación de instancias [5] y segmentación débilmente supervisada [3] exploran la unión entre la supervisión a nivel de píxel y la supervisión mediante cuadros delimitadores con la finalidad de disminuir el uso de recursos computacionales, con esto se ha logrado obtener una mejora en la precisión de la detección y clasificación, sin embargo, aún quedan algunos rasgos sin identificar.

Por lo anterior, en este trabajo se propone la implementación de una red neuronal convolucional aplicando una arquitectura VGG16 para realizar la segmentación semántica en imágenes aéreas que incluyen zonas de vegetación para detectar árboles de naranjo y finalmente aplicar transferencia de aprendizaje para mejorar los resultados de precisión obtenidos.

2. Trabajos relacionados

El reconocimiento de objetos mediante la segmentación semántica, es un término general para describir una colección de tareas relacionadas con la visión por computadora, que involucran el reconocimiento de imágenes y análisis de datos cómo se menciona en [6, 12]. Esta acción ha sido la base del estudio de nuevos métodos para realizar estas tareas acercándose cada vez más a la automatización.

La segmentación semántica es realizada con un algoritmo de aprendizaje profundo como en [8, 10, 11] que es implementado usando redes neuronales convolucionales para realizar la tarea de identificación y clasificación en imágenes, como menciona Long J. at al. (2015) donde el enfoque principal de su trabajo está relacionado a la aplicación de redes totalmente convolucionales para extender la clasificación hasta la segmentación y mejorar la técnica realizando combinaciones de arquitecturas como AlexNet[14], VGG net [15], GoogLeNet [16], para obtener segmentación semántica con capas profundas gruesas y capas poco profundas finas para producir segmentaciones precisas y detalladas.

Ellos reportan mejora en los resultados y simplificó y acelero el aprendizaje de inferencia, sin embargo sus resultados reportados se encuentran alrededor del 62.2 % IoU promedio (métrica de intersección sobre la unión). Por otro lado, se ha abordado la segmentación semántica implementando una red SegNet [13], la cual se basa en una arquitectura de CNN [8, 11], para realizar la comprensión de escenas mediante la segmentación semántica por medio de una red de codificación.

Esta red es idéntica topológicamente a la arquitectura VGG16, solo se omiten las capas completamente conectadas para que esta sea más pequeña y más fácil de entrenar. He, K. et al. (2017) [5], introduce técnicas de Mask R-CNN, que realiza instancias de segmentación en las imágenes con algoritmos que pueden ser reentrenables.

Con esta técnica se mejoró de forma significativa los resultados de precisión de la segmentación aplicando Faster R-CNN [9], pues realizan máscaras de segmentación en cada Región de Interés (RoI) para mejorar la predicción. Por otro lado, Guo, R., et al. (2020) [3], realiza segmentación semántica con múltiples etiquetas usando algoritmos que utilizan cuadros delimitadores y técnicas que mejoran la detección de objetos tales como: Grab Cut, GradCAMD y GrabCutC.

El entrenamiento es realizado con un modelo R-CNN y utilizan una red troncal Resnet50. Con esta técnica demostraron mejoraras en la segmentación y rendimiento aplicando la técnica al conjunto de datos iSAID. Lobo Torres, at al (2020) [17], evalúo cinco redes totalmente convolucionales: SegNet, U-Net, FC-DenseNet y dos variantes de DeepLabv3+ para realizar segmentación semántica de una especie de árbol; además, verificó los beneficios del fully connected conditional random (CFR) como un paso de procesamiento posterior para mejorar los mapas de segmentación.

En su análisis experimental mostró resultados promedio de precisión de entre 88.9 % a 96.7 %. Además indico que CFR mejoró el rendimiento pero a un alto costo computacional. Osco, L., et al. (2021) [1], resalta la importancia de la cartografía en áreas agrícolas; utilizó vehículos aéreos no tripulados (VANT) con la integración de cámaras multiespectrales que facilitan el mapeo de plantas en paisajes agrícolas.

En [1] aplican métodos de aprendizaje profundo para realizar segmentación semántica a imágenes espectrales de árboles cítricos, utilizando cinco arquitecturas diferentes para realizar la segmentación: FCN, U-Net, SegNet, dynamic dilated convolution network (DDCN) y Deep-LabV3 +. Con estos métodos de aprendizaje profundo, obtuvieron precisiones que van desde el 94.88 % al 95.46 %.

De esta forma se puede ver que se ha implementado CNN con las distintas arquitecturas conocidas y mencionadas en este trabajo con aplicaciones a distintas áreas, pero sobre un mismo objetivo, realizar segmentación para la detección y clasificación de distintos objetos de diferentes clases, una ventaja de la red VGG16 respecto a otras es que su arquitectura es fácil de comprender y de implementar, logró excelentes resultados en la competencia ImageNet (ILSVRC-2014) entre el 96 % y 97 %, ademas podemos encontrar esta red disponible en Keras, también fue entrenada para resolver problemas de clasificación de 1000 clases en el conjunto de datos de ImageNet el cual se conforma por mas de 1.4 millones de imágenes, por esto es utilizada para realizar entrenamiento en áreas a fines a los objetos etiquetados o para continuar con la segmentación de otras clases, se reduce el tiempo de entrenamiento de la red gracias al uso de los pesos ImageNet. La finalidad principal es mejorar los tiempos de procesamiento de la información, obtener una mejor precisión en el resultado y reducir el uso de recursos computacionales durante el procesamiento.

3. Método propuesto

Existen asociaciones que se dedican al cuidado y preservación de las especies vegetales en peligro de extinción y es mediante la recolección de datos que pueden obtener información del lugar y la especie para poder realizar registros y obtener estadísticas de los cambios pertinentes de la evolución de la diversidad de la flora en el área.

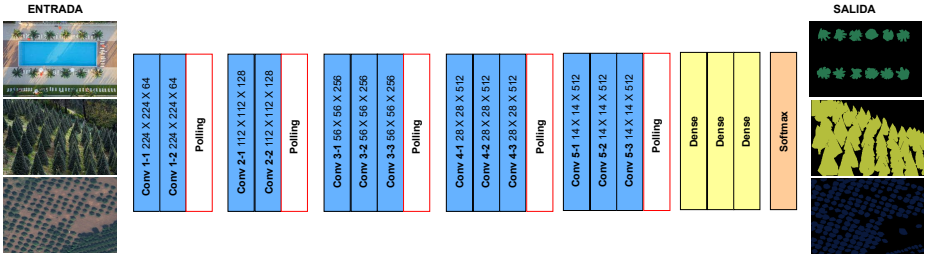


Fig. 1. Diagrama de arquitectura VGG16.

Sin embargo, esta actividad resulta no ser tan fácil, cómoda y segura de realizar en la forma tradicional, es por ello que gracias al uso de la tecnología hoy en día se puede explorar la zona de manera segura sin exponer la integridad del personal a cargo de esta labor, ya que con el uso de drones podemos obtener imágenes de estas áreas y analizar e identificar mediante algoritmos de inteligencia artificial los elementos que se encuentran presente en las imágenes.

Es por esto que principalmente se propone la implementación de una CNN basada en un algoritmo VGG16 pre-entrenado con el conjunto de datos ImageNet para la segmentación semántica de imágenes con vegetación, cuya forma general de funcionamiento se puede ver en la Fig. 1 que se muestra a continuación.

Como se observa en la Fig. 1, a la entrada de la arquitectura se tiene la imagen re-dimensionada a $224 \times 224 \times 3$ que contiene la escena donde se pueden apreciar zonas con árboles de palmeras, pinos y naranjos. Se realiza con cada una de las imágenes el recorrido en la arquitectura por cada una de las capas donde se aplican filtros convolucionales que permitirán la extracción de características particulares de cada una de las clases.

Según se avanza en la arquitectura el número de convoluciones será mayor, esto para poder extraer características más particulares que puedan lograr hacer la distinción entre un objeto y otro. Así pues, se lleva a cabo el proceso hasta finalizar con la última imagen que conforma el conjunto de datos de entrenamiento para enseñar al algoritmo a identificar cada clase.

Este procedimiento es repetido con el conjunto de datos de validación para poder saber y medir la precisión que obtuvo en el entrenamiento, así se puede determinar si se debe volver a entrenar y/o ajustar los parámetros de número de pasos, épocas u otro dato.

Una vez que se obtiene la matriz que contiene los resultados del entrenamiento, se guardan los pesos para usarlos durante la aplicación de la transferencia de aprendizaje. A continuación se enumeran los pasos que se realizaron previo para realizar este trabajo:

1) Adquisición de datos: En esta etapa se conformó el conjunto de datos, mediante la recolección de imágenes de distintas fuentes para el caso de datos de palmeras y pinos; para el conjunto de datos de árboles de naranjo se pudieron obtener las imágenes de una fotografía aérea capturada de una plantación de naranjos en la zona del estado de Veracruz,

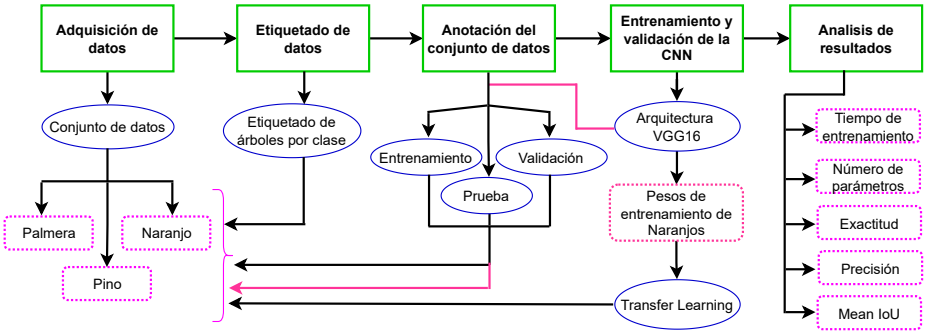


Fig. 2. Esquema del flujo de trabajo.

2) Etiquetado de datos: Esta acción se realiza de forma manual con la ayuda del software: ImageAnnotator, este permite realizar el etiquetado de las clases (palmera, pino y naranjo) mediante la selección de forma delimitada con puntos alrededor del borde del árbol, generando polígonos y finalmente se le asigna el nombre que le corresponde a la clase de árbol,
3) Anotación del conjunto de datos: A partir del etiquetado de las imágenes se realizaron máscaras de segmentación utilizando los puntos de intersección en las imágenes etiquetadas. Con estas imágenes y sus máscaras de segmentación de cada clase, se forman los conjuntos de datos de entrenamiento y validación,
4) Entrenamiento y validación de la CNN: Se ejecuta la CNN, a partir de la arquitectura VGG16 seleccionada y se establecen y/o ajustan parámetros específicos para la ejecución del entrenamiento de la red con los distintos conjuntos de datos que hemos formado para este trabajo. Una vez realizado el entrenamiento se emplea la validación para determinar el correcto funcionamiento del algoritmo,
5) Análisis de resultados: Finalmente se analizan los resultados obtenidos mediante las métricas que nos indican la precisión del algoritmo dependientes de valores como el tiempo de entrenamiento, número de parámetros, etc.

En la Fig. 2 se detalla el procedimiento de forma esquematizada.

3.1. Adquisición de datos

Para formar los conjuntos de datos, se realizó una recopilación de imágenes aéreas en las cuales se visualizará árboles de palmeras, pinos y en el caso de naranjos, las imágenes fueran obtenidas de una captura aérea de una zona de cultivos de naranjos en el estado de Veracruz. Estas imágenes se encuentran en formato RGB y de distintas dimensiones. En el caso de las palmeras y pinos las imágenes tenían un tamaño inicial de 480×480, y las de los naranjos 255×255. En la Fig. 3 se muestra algunas imágenes que conforman la base de datos de cada clase.

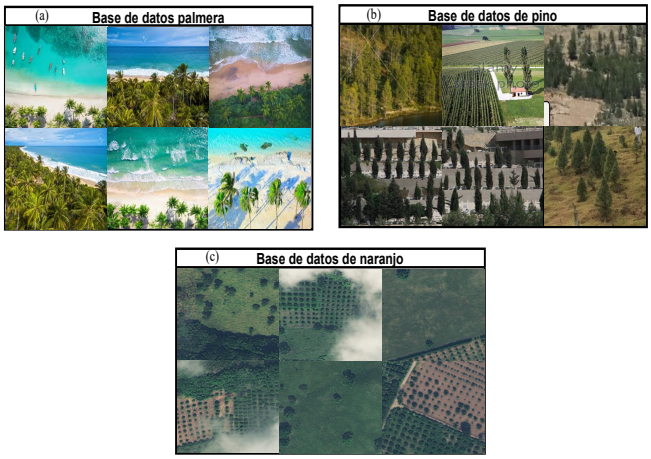


Fig. 3. Imágenes aéreas: (a) palmeras, (b) pinos, (c) naranjos.

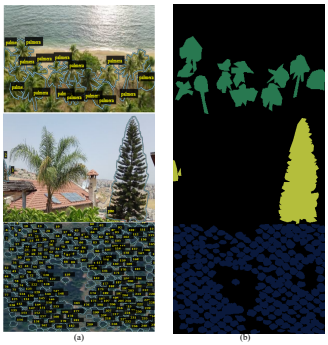


Fig. 4. (a) Etiquetado de las clases, (b) Máscaras de las clases.

3.2. Etiquetado de datos

El etiquetado de los árboles de las imágenes consiste en ir asignando a cada imagen la clase a la que pertenecen para cada tipo: palmera, pino y naranjo. Se utilizó el sotfware Image Annotator [18] para realizar esta tarea.

Una vez finalizado el etiquetado de cada imagen se genera un archivo descargable que contiene los puntos que indican y seccionan el área que abarca cada árbol en la escena. Este archivo es procesado en un programa realizado en lenguaje Python para obtener las máscaras de segmentación de las clases. En la Fig. 4 se muestra ejemplo del etiquetado por clase, y la máscara obtenida.

3.3. Anotación del conjunto de datos

Para la generación de los conjuntos de datos una vez que se ha realizado el etiquetado de las 420 imágenes y generado las máscaras, se forma el conjunto de datos de entrenamiento, este contiene 340 imágenes de distintas zonas y el objeto principal

Tabla 1. Datos de entrenamiento y validación de red con imágenes de pino.

Prueba	Batch	Epoch	Precisión de pixel	IoU	Tiempo
1	6	50	94 %	86 %	1560 seg
2	6	80	96 %	90 %	1980 seg
3	10	120	96.5 %	92 %	2975 seg

Tabla 2. Datos de entrenamiento y validación de red con imágenes de naranjo.

Prueba	Batch	Epoch	Precisión de pixel	IoU	Tiempo
1	6	50	87 %	64 %	2280 seg
2	6	80	89 %	70 %	2640 seg
3	10	120	91 %	73 %	3040 seg

son árboles de palmeras, pinos y naranjos lo que corresponde al 80 % del conjunto de datos, estas se utilizan para realizar el entrenamiento de la red. El conjunto de datos para validación, está conformado por 80 imágenes del conjunto de datos que corresponde a el 20 % del conjunto, estas son utilizadas para evaluar la precisión de la segmentación de las distintas clases de árboles trabajados.

Para el conjunto de datos de prueba se utilizaron 15 imágenes RGB que contienen en la escena las 3 clases de árboles, las imágenes de prueba son empleadas finalmente para probar el funcionamiento de la red mediante la segmentación de las clases palmera, pino y naranjo y así verificar de forma visual la precisión en el trabajo, estos conjuntos de datos podrán estar disponibles para uso público(una vez que el comité correspondiente acredite su publicación, la cual ya esta en proceso) en la plataforma GitHub.

3.4. Entrenamiento y validación

Una vez conformado el conjuntos de datos, se inicia la ejecución del programa de CNN para poder realizar el entrenamiento, validación y finalmente la prueba, con la cual podremos ver si se cumple el objetivo que es obtener la segmentación semántica de la imagen con clase de árbol que corresponde, esto toma un tiempo que va de 30 a 50 min.

La ejecución del programa se realiza en Python con la paquetería de TensorFlow [1] utilizando la arquitectura VGG16 pre-entrenada con datos de ImageNet. Para este trabajo se implementa finalmente la técnica de transferencia de aprendizaje [2], con la cual se pretende usar bajos recursos computacionales, disminuir el tiempo de procesamiento durante el entrenamiento y adaptar pesos del modelo para aplicarlos en el aprendizaje de las clases trabajadas.

3.5. Análisis de resultados

Para realizar la evaluación del algoritmo, se utilizaron las siguientes métricas:

– Tiempo de entrenamiento.- El tiempo de entrenamiento es el tiempo que le tomó al programa realizar el entrenamiento del algoritmo, este varía dependiendo del número de imágenes a procesar y va desde los 30 min a 1:40 min,

Tabla 3. Datos de entrenamiento y validación de red con imágenes de palmera.

Prueba	Batch	Epoch	Precisión de pixel	mean IoU	Tiempo
1	6	50	91 %	76 %	1680 seg
2	6	80	93 %	82 %	2040 seg
3	10	120	95 %	86 %	2400 seg

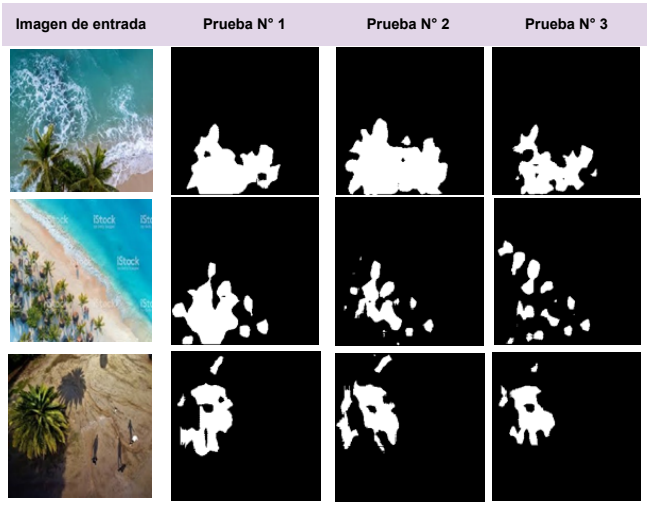


Fig. 5. Resultado de segmentación semántica de árboles de palmera.

– Número de pasos (batch size).- Define el número de muestras que se propagara a través de la red,
– Número de épocas (epoch).- Este determina el número de veces que el conjunto completo de imágenes será procesado por la red,
– Exactitud (accuracy).- La exactitud se refiere a lo cerca que esta el resultado de una medición del valor verdadero. En términos estadísticos esta se relaciona con el sesgo de una estimación,
– Precisión.- Se representa por la proporción de verdaderos positivos dividido entre todos los resultados positivos (tanto verdaderos positivos como falsos positivos), de esta manera se puede ver de forma porcentual que tan precisa fue la predicción respecto al conjunto de datos,
– IoU.- Es una métrica de evaluación utilizada comunmente para la segmentación semántica de imágenes. Esta métrica se define como:

$$IoU = \frac{VP}{VP + FP + FN}, \tag{1}$$

donde VP son los verdaderos positivo, FP los falsos positivo y FN representados por los falsos negativo.

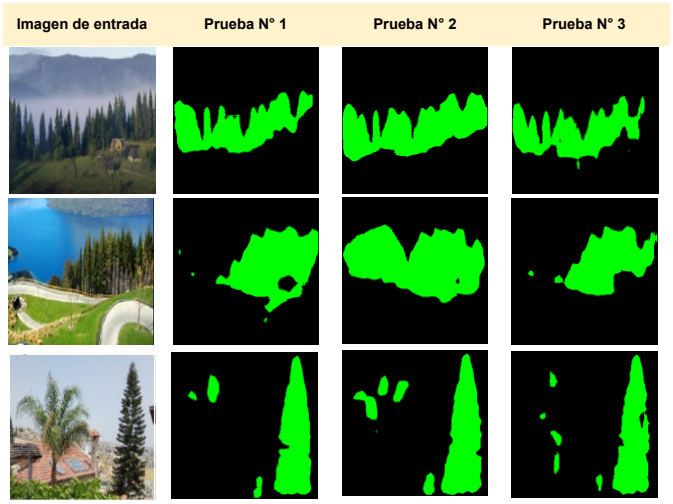


Fig. 6. Resultado de segmentación semántica de árboles de pino.

4. Pruebas y resultados

Se realizaron las pruebas con los 3 conjuntos de datos de árboles de palmeras, pinos y naranjos. Ahora se explica y muestra los resultados obtenidos en cada prueba.

4.1. Resultados con conjunto de datos de palmeras usando pesos ImageNet

Para el entrenamiento con el conjunto de datos de palmeras, se considero en el programa 3 pruebas estableciendo los parámetros que se muestran en la Tabla 3. Además se puede ver el resultado obtenido reflejado en porcentajes, teniendo entonces 91 %, 93 % y 95 % de precisión de píxel en las pruebas 1, 2 y 3.

En la Fig. 5 se pueden visualizar los resultados obtenidos en las imágenes del conjunto de datos de palmeras. El tiempo de ejecución para obtener la segmentación semántica en cada caso estuvo alrededor de los 15 a 20 min.

4.2. Resultados con conjunto de datos de pinos usando pesos ImageNet

Para el entrenamiento con el conjunto de datos de pinos también fueron consideradas 3 pruebas, estableciendo los parámetros que se muestran en la Tabla 1. Se refleja de forma porcentual la mejora de la precisión del resultados teniendo 94 %, 96 % y 96.5 % de precisión de píxel en las pruebas 1, 2 y 3 respectivamente. A continuación, en la Fig. 6 se muestran parte de los resultados obtenidos para cada una de las pruebas en las imágenes del conjunto de datos de pino.

4.3. Resultados con conjunto de datos de naranjos usando pesos ImageNet

Para el entrenamiento con el conjunto de datos de naranjo también se consideraron las 3 pruebas con los parámetros establecidos en los conjuntos de palmera y pino,

Tabla 4. Datos aplicando transferencia de aprendizaje con base de datos de naranjo.

Prueba	Batch	Epoch	Precisión de pixel	IoU	Tiempo
1	6	50	93 %	81 %	1896 seg
2	6	80	95 %	85 %	2338 seg
3	10	120	96 %	87 %	2720 seg

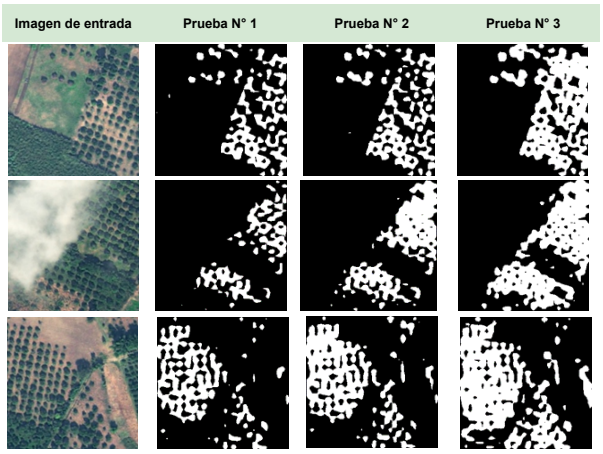


Fig. 7. Resultado de segmentación semántica de árboles de naranjo.

ver Tabla 2, donde los resultados de precisión de píxel obtenidos fueron 87 %, 89 % y 91 % respectivamente. En la Fig. 7 son mostrados algunos resultados de segmentación semántica obtenidos para cada una de las pruebas con los parámetros establecidos en la Tabla 3. Una vez analizado los casos para los 3 conjuntos de datos de árboles, se toma la decisión de realizar la implementación de la transferencia de aprendizaje para el caso del conjunto de datos de árboles de naranjo.

4.4. Resultados aplicando transferencia de aprendizaje a conjunto de datos de naranjos

Se decidió aplicar la técnica de transferencia de aprendizaje con la finalidad de mejorar la precisión de la detección y segmentación de la clase e implementar una técnica que resulte significativa para trabajos futuros y que puedan ayudar a reducir los tiempos de ejecución costos y recursos computacionales. En la Tabla 4, se muestran los parámetros establecidos para realizar el entrenamiento utilizando los pesos que se obtuvieron del entrenamiento con el conjunto de datos de naranjos pero ahora implementando la transferencia del aprendizaje con árboles de naranjos.

Se observa como se mejora los resultados de precisión en cada prueba comparada con el obtenido en la prueba usando pesos de ImageNet. Ver Tabla 2. Ahora en la Fig. 8 se muestran los resultados que se obtuvieron aplicando la transferencia de aprendizaje en la arquitectura VGG16 para obtener la detección de los árboles de naranjo mediante la segmentación semántica.

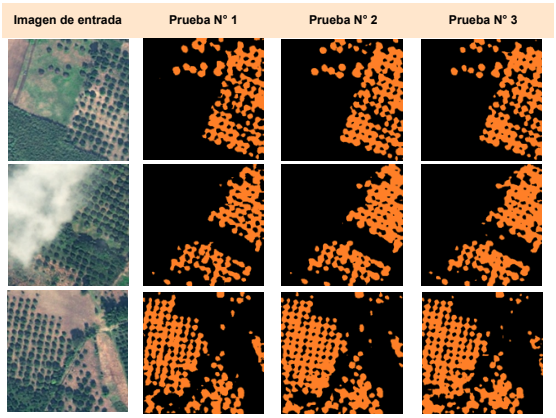


Fig. 8. Resultado de segmentación semántica en árboles de naranjo aplicando transferencia de aprendizaje.

En la imagen de entrada 1 se puede ver que en la prueba 1 se detectan algunos píxeles de árboles que están pegados a otros y no se pueden visualizar como uno solo, sin embargo para esa misma imagen en la prueba 2 aún se siguen visualizando árboles segmentados no separados pero en menor cantidad, para el caso de la prueba 3 de la misma imagen se visualiza una mejor precisión y se observa cómo se ha definido un poco mejor entre un árbol y otro, esto rectifica los datos obtenidos en la Tabla 4 y por lo tanto también si comparamos la Tabla 4 con la Tabla 3 se demuestra que al aplicar la técnica de transferencia de aprendizaje se logra mejorar de forma significativa los resultados.

5. Conclusiones y trabajo a futuro

Se puede observar que la aplicación de las CNN mediante la arquitectura VGG16 es un procedimiento que logró el objetivo principal que era detectar y segmentar las clases de las especies de árboles utilizadas en este trabajo, obteniendo resultados con una precisión alrededor del 87 % hasta 96.5 %, lo cual demuestra que fue efectivo y eficiente el uso de los datos y la aplicación de la técnica de transferencia de aprendizaje.

Con esto se puede decir que se puede implementar la metodología propuesta entrenando con otros conjuntos de datos de áreas de vegetación principalmente para tratar con especies en peligro de extinción, y poder abordar y proponer soluciones relacionadas a este tema importante, mediante el monitoreo constante de zonas protegidas donde se encuentren estas especies.

Se demostró que la transferencia de aprendizaje ayuda a mejorar los resultados de detección de objetos, segmentando además de forma semántica las clases específicas. Es importante mencionar que también, esta técnica ayudo a reducir los tiempos de ejecución y a evitar el uso excesivo de recursos computacionales. Debido a que esta técnica utiliza el conocimiento obtenido mediante el entrenamiento para realizar la segmentación de otras clases, solo se deberá aplicar a un trabajo relacionado al área para no iniciar todo el procedimiento desde cero en el entrenamiento.

Como trabajo a futuro se propone realizar mediante la aplicación de la técnica de transferencia de aprendizaje, la detección y segmentación de distintas clases de árboles en imágenes aéreas de zonas con vegetación, además de implementar este método con arquitecturas como: DeepLabV3, SegNet, Faster R-CNN, aplicado a imágenes RGB y Multiespectrales. También se propone la aplicación de la técnica de Fine tuning.

Referencias

1. Osco, L. P., Nogueira, K., Marques-Ramos, A. P., Faita-Pinheiro, M. M., Furuya, D. E. G., Gonçalves, W. N., dos-Santos, J. A.: Semantic segmentation of citrus-orchard using deep neural networks and multispectral UAV-based imagery. Precision Agriculture, vol. 22, no. 4, pp. 1171–1188 (2021) doi: 10.1007/s11119-020-09777-5
2. Tammina, S. Transfer learning using VGG-16 with deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications (IJSRP), vol. 9, no. 10, pp. 143–150 (2019) doi: 10.29322/IJSRP.9.10.2019.p9420
3. Guo, R., Sun, X., Chen, K., Zhou, X., Yan, Z., Diao, W., Yan, M. Jmlnet: Joint multi-label learning network for weakly supervised semantic segmentation in aerial images. Remote Sensing, MDPI, vol. 12, no. 19, pp. 3169 (2020) doi: 10.3390/rs12193169
4. Campillo, L. M. G., Torres, R. A. C., López, H. M. D: Percepción remota: Elementos básicos. Kuxulkab', vol. 21, no. 40 (2015) doi: 10.19136/kuxulkab.a21n40.1001
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
6. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
7. Martins, J. A. C., Nogueira, K., Osco, L. P., Gomes, F. D. G., Furuya, D. E. G., Gonçalves, W. N., Junior, J. M., et al.: Semantic segmentation of tree-canopy in urban environment with pixel-wise deep learning. Remote Sensing, vol. 13, no. 16, pp. 3054 (2021) doi: 10.3390/rs13163054
8. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: International conference on machine learning, PMLR, vol. 32, no. 1, pp. 647–655 (2014)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, vol. 28 (2015)
10. Lateef, F., Ruichek, Y.: Survey on semantic segmentation using deep learning techniques. Neurocomputing, vol. 338, pp. 321–348 (2019) doi: 10.1016/j.neucom.2019.02.003
11. Audebert, N., Saux, B. L., Lefèvre, S.: Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. Asian conferen-

ce on computer vision, Springer, Cham, vol. 10111, pp. 180–196 (2016) doi: 10.1007/978-3-319-54181-5_12
12. Butler, M. J. A., Mouchot, M. C., Barale, V., LeBlanc, C.: Aplicación de la tecnología de percepción remota a las pesquerías marinas: manual introductorio. Organización de las Naciones Unidas para la Agricultura y la Alimentación, no. 639.2028 BUTa (1990)
13. Badrinarayanan, V., Kendall, A., Cipolla, R.: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495 (2017)
14. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017) doi: 10.1145/3065386
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) doi: 10.48550/arXiv.1409.1556
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
17. Lobo-Torres, D., Queiroz-Feitosa, R., Nigri-Happ, P., Cué-laRosa, L. E., Marcato-Junior, J., Martins, J., Liesenberg, V.: Applying fully convolutional architectures for semantic segmentation of a single tree species in urban environment on high resolution UAV optical imagery. Sensors, vol. 20, no. 2, pp. 563 (2020) doi: 10.3390/s20020563
18. Sydorenko, I.: Image annotation tools for machine learning. Label Your Data (2021)

ISSN 1870-4069

Esteganografía de códigos QR embebidos por medio de un modelo neuronal generativo adversario

Rodrigo Hernández Moncayo, José Martin Flores Albino,
Víctor Manuel Landassuri Moreno, Saturnino Job Morales Escobar

Universidad Autónoma del Estado del México,
Centro Universitario UAEM Valle de México,
México

rhernandezm403@alumno.uaemex.mx,
{jmfloresa,vmlandassurim,sjmoralese}@uaemex.mx

Resumen. La esteganografía a través de técnicas de aprendizaje profundo se ha visto ampliamente impulsada a través de la Redes Neuronales Generativas-Adversarias. Este trabajo presenta la implementación del modelo ISGAN para embeber (ocultar) un código QR en otro código QR, utilizarlo como una marca de agua para validación de la autoría del código original. Se presentan los resultados de las imágenes estego evaluadas por medio de las métricas de calidad de imagen: SSIM, PSNR, UQI y VIF de las imágenes estego y la recuperación y decodificación de las imágenes QR ocultas.

Palabras clave: Esteganografía, código QR embebido, red neuronal generativa antagónica.

Steganography of Embedded QR Codes based on a Generative Adversarial Neural Model

Abstract. Steganography through deep learning techniques has been widely driven through Generative-Adversarial Neural Networks. This paper presents the implementation of the ISGAN model to embed (hide) a QR code in another QR code, use it as a watermark for validation of the authorship of the original code. The results of the stego images evaluated by means of the image quality metrics: SSIM, PSNR, UQI and VIF of the stego images and the recovery and decoding of the hidden QR images are presented.

Keywords: Steganography, embedding QR code, generative-adversarial neural networks.

1. Introducción

La *esteganografía* ha sido utilizada en las últimas décadas para poder ocultar información en medios digitales públicos. Usado, por ejemplo, para la prevención del robo de información y la protección de la propiedad intelectual. Se ha utilizado para identificación de documentos de identidad, licencias y propiedad de información en formato de video o imágenes [1]. En la esteganografía se definen tres cualidades de la información que compiten entre sí, la Capacidad, Robustez y la Imperceptibilidad que deben de permanecer en equilibrio. Los elementos en la esteganografía de imágenes digitales son: *imagen cubierta,* es donde se oculta la información; la *imagen secreta*, es la imagen que se quiere ocultarse; y como resultado del proceso estenográfico se produce *imagen estego,* es la imagen cubierta con la información de la *imagen secreta* imperceptible; la *imagen recuperada* que la imagen resultante de aplicar el proceso esteganográfico en sentido inverso [2].

El *estegoanálisis* tiene como objetivo es comprobar si existe un mensaje secreto y de ser posible extraerlo. Hipotéticamente, el estegoanálisis, implementa métodos para lograr vulnerar la esteganografía, las cuales pueden reducir o eliminar la capacidad de imperceptibilidad [3].

Los *códigos QR* (Quick Response Code) son imágenes que codifican y almacena información en una matriz de puntos bidimensional en módulos sobre un fondo claro, que puede ser recuperada por medio del sensor de la cámara fotográfica y la capacidad de procesamiento de los dispositivos móviles.

Una de las características principales de los códigos QR, es el nivel del factor de corrección de errores (ECC) la cual usa una codificación tipo *Reed-Solomon,* en cuatro niveles: L (7%), M (15%), Q (20%) y H (30%), que define la capacidad de robustez para recuperar la información por deterioro de la imagen, pero reduce la capacidad de los datos que pueden ser codificados. Los caracteres que acepta la codificación QR son: numéricos, alfanuméricos, binarios y kanji y su capacidad depende del número de módulos que tenga considerando la versión del código QR [4].

Los códigos QR han sido empleados en la mercadotecnia para acceso a anuncios publicitarios en las redes sociales y páginas web. Se han usado para acceder a enlaces para descarga de catálogos alojados en la nube. En pagos de servicios por medio de aplicaciones bancarias. Para transmitir mensajes de texto. Para facilitar la descarga del menú en restaurantes. [5].

En la literatura se pueden encontrar diversos documentos que implementan esteganografía en códigos QR para aumentar su seguridad o como marcas de agua. En [6] se presenta la implementación de un proceso estenográfico por algoritmo LSB para embeber un código QR en otro, por medio de un algoritmo estenográfico heurístico se estableció la relación entre la información de imagen de los códigos QR cubierta y el QR secreto para ser embebido.

En [7] se implementa una aplicación para comunicación militar vía código QR el cual se utiliza para codificar el mensaje estenográfico por medio de la compuerta XOR y aplican técnicas de hashing para aprovechar las características de robustez en la codificación y facilidad de lectura que tiene el código QR y aprovechar la capacidad que tiene como imagen para incrustar información sin modificar la codificación.

En el trabajo de [8] se presenta una técnica para mejorar la capacidad, la seguridad y la solidez del proceso de incrustación de datos, el esquema propuesto utiliza una técnica de codificación criptográfica base 64 para transformar los datos secretos mediante una técnica de reemplazo dinámico en los bits de los canales RGB la cual mostró buenos resultados ante ataques de estegoanálisis basados en fuerza bruta, estadísticos y diferenciales.

Por los avances en la Inteligencia Artificial, así como el incremento del poder de cómputo, se han implementado nuevas técnicas basadas en redes neuronales artificiales aplicadas al aprendizaje profundo. Una de esas técnicas son las llamadas *Redes Generativas Adversarias* (GAN por sus siglas en inglés) propuestas en el año 2014 por Goodfellow y colegas [9].

Esta técnica basada en la teoría de juegos combinada con el aprendizaje profundo tiene dos elementos principales: un *discriminador* (D) y un *generador* (G); los cuales compiten durante la fase de entrenamiento para alcanzar un equilibrio en sus *funciones de perdida* como dos adversarios.

El uso de Inteligencia Artificial, y en particular de Redes Neuronales Adversarias, ha mostrado beneficiar el proceso estenográfico en sus características de Capacidad, Perceptibilidad y Seguridad.

La arquitectura HIGAN [10], se construye la red de codificación compuesta por bloques residuales para ocultar la imagen secreta de color en otra imagen de color del mismo tamaño, los resultados al crear la *imagen estego* son buenos y con baja distorsión, y en la imagen recuperada no presenta cambios significativos.

Zhu utiliza la arquitectura DGANS [11] y propone un doble modelo GAN conectado de manera consecutiva para fortalecer el entrenamiento utilizando el algoritmo XuNet para estegoanálisis como discriminador, las imágenes utilizadas son del mismo tamaño en escala de grises.

Dentro del estegoanálisis existen varios problemas de falsos positivos por lo que en el trabajo de [12], se propone usar una GAN para etiquetar imágenes como imágenes estego, se logre eliminar el mensaje oculto y, preservar la imagen estego sin destruirla, evitando así un posible ataque por medio de mensajes maliciosos.

En el presente trabajo se propone el uso de un modelo de Red Neuronal Generativa Adversaria llamado ISGAN para implementar la tarea de embeber una imagen de código QR en otra.

ISGAN fue propuesto por [13]. Se elige debido a que muestra la capacidad de ocultar una imagen dentro de otra del mismo tamaño utilizando la descomposición en el espacio de color YCbCr y ocultando la imagen secreta en escala de grises en el canal Y de luminancia de la imagen portadora.

De acuerdo a esta característica principal al utilizar imágenes con código QR que están compuesta por dos colores blanco y negro principalmente, cuando se realiza la conversión de este tipo de imágenes a tonos de grises no sufre cambios visuales y tampoco se pierde su capacidad de codificación es por ello que el modelo ISGAN puede ser idóneo para realizar esteganografía en imágenes con código QR. ISGAN puede lograr niveles de alta calidad visual de la imagen estego y de la imagen secreta lo que hace posible decodificar la información de contenido en ellos.

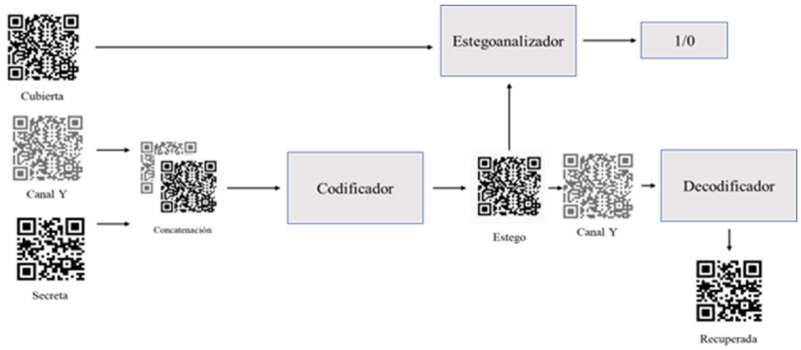


Fig. 1. Gráfico de bloques de la arquitectura de ISGAN.

1.1. Organización del trabajo

En la sección 2. *Desarrollo* se trata con profundidad la estructura del modelo base y el discriminador que forman la ISGAN, así como de la función de perdida basada en índices de calidad de imagen, además de observar la tendencia en función de épocas de entrenamiento de la ISGAN. En la Sección 3, *Resultados Experimentales*, se presentan los resultados del modelo para embeber un código QR secreto en otro QR de cubierta. Se realiza la evaluación de imágenes cubierta vs imágenes estego e imágenes ocultas vs imágenes recuperadas, a través de las métricas SSIM, PSNR, VIF y UQI. En la Sección 4. *Conclusiones* se muestra la capacidad que tuvo el modelo basado en ISGAN, además de valorarse visualmente los resultados. Los índices de calidad de imagen determinarán que no existen cambios importantes de estructura, contraste o textura en las imágenes estego y recuperadas.

2. Desarrollo

En esta sección se presenta la arquitectura del modelo de ISGAN, mostrando sus componentes y función de perdida para la red de Generación y Discriminante. El modelo de la GAN se divide en una Red Neuronal Generadora compuesto por una estructura *Codificador-Decodificador*, y una Red Discriminadora que tienen como rol ser el *estegoanalizador*. También se presenta la implementación para su aplicación experimental de embeber una imagen de un código QR en otro, el desempeño durante el entrenamiento y los requerimientos de software y hardware para su implementación.

La principal característica de este modelo de GAN es que usa el canal de luminancia en una *imagen de portada* para ocultar una imagen en escala de grises del mismo tamaño. Se describe a continuación el modelo ISGAN, compuesto por un modelo básico y un estegoanalizador representado en la Figura 1.

2.1. Modelo ISGAN (Esteganografía invisible con redes generativas adversarias)

El modelo ISGAN fue propuesto por [13]. y está inspirado en el trabajo de [14] y [15]; esta arquitectura ataca los principales problemas de distorsión que tiene la esteganografía por medio de CNN (Redes Convolucionales Profundas, por sus siglas en inglés) para crear las imágenes estego. Este modelo reporta mejoría en la calidad de la imagen resultante del proceso estenográfico, por medio de la propuesta de una función de perdida con base en índices de calidad de imagen, favoreciendo las características de robustez, seguridad y capacidad del proceso.

Modelo Básico: El modelo básico está compuesto de dos redes convolucionales. Al principio está el *Codificador*, que tendrá la tarea de generar la imagen estego. En serie a éste se encuentra la red convolucional del *Decodificador* que es la que recibe la imagen estego y genera la imagen recuperada, la cual debe ser similar a la imagen secreta.

Codificador: El modelo básico este compuesto por un codificador, que tiene como entrada la imagen de cubierta, que previamente se descompone de formato **RGB** al espacio color compuesto por un canal de luminancia y dos de croma (***YCbCr***). En el canal de Luminancia (Y) la imagen de cubierta se concatena con la imagen secreta (en escala de gris) para formar la entrada del decodificador.

Esta conversión del espacio de color tiene como objetivo evitar alterar la información de color de la imagen de cubierta reduciendo la percepción de cambios en los tonos de color de la imagen estego resultante. El codificador utiliza un *módulo de Inception* [16], con el objetivo de mezclar los mapas de características de la imagen cubierta y la imagen secreta, realizado en el mismo nivel de la red de la arquitectura del codificador, teniendo el efecto de ensanchar el filtrado de la imagen en paralelo y de esta forma apoyando para que la red convolucional no sea tan profunda, dando el mismo rendimiento y compactando el codificador. La salida del codificador es la imagen estego en canal Y. Para completar la imagen estego en color, se agregan los canales Cr y Cb de la imagen cubierta original.

Decodificador: El decodificador es una red CNN que se emplea para extraer la *imagen secreta* y utiliza normalización por lotes. Como la entrada del decodificador ingresa la imagen estego que a través de su propagación y filtrado recupera la imagen secreta, a esta imagen se denomina *imagen recuperada* debido que habrá algún grado de perdida de detalles de la imagen secreta original.

Así con el *Codificador-Decodificador* se integra el *modelo básico.* Como se observa en la Fig. 1. La ISGAN es un modelo basado en Redes Generativas Adversarias, el modelo base corresponde al Generador.

Por otro lado, el Discriminador es una red convolucional que opera como Estegoanalizador. Durante el entrenamiento se busca un equilibrio del desempeño de estos componentes.

Estegoanalizador: Este módulo, tiene como base a la red CNN XuNet [17], ha sido utilizado en la computación forense para estegoanálisis en los algoritmos de *esteganografía espacial*. Esta red incorpora una capa *Activación Absoluta* (ABS layer) a la salida de la primera capa convolucional cuyos valores absolutos permiten mejorar

el modelado estadístico del ruido residual en las capas siguientes, para las dos primeras capas también se aplica la activación de tangente hiperbólica (*Tanh*) para evitar el sobreajuste en la red.

En las capas siguientes se agrega una capa con el módulo de agrupación de *pirámides espaciales* (SPP) [18], donde se requiere que este módulo se alimente de otro clasificador para favorecer su entrenamiento, por lo que se coloca posterior a las capas basadas en XuNet. El módulo SSP permite el entrenamiento con imágenes de distintos tamaños facilitando utilizar también filtros más anchos o más largos dependiendo de las imágenes que se clasificará. En este proceso, SSP añade características que permiten mejorar el entrenamiento entre cada época del ajuste de pesos.

2.2 Implementación de la red generativa adversaria

El entrenamiento de una GAN se realiza buscando un equilibrio entre la Red Generadora y la Red Discriminante. La Red generadora debe producir ejemplares que progresivamente sean similares a las imágenes del conjunto de entrenamiento, y por otro lado la Red Discriminante debe competir con el Generador al criticar los ejemplares del Generador, clasificándolos como falsificaciones de las imágenes del conjunto de entrenamiento.

Por lo que el índice de desempeño (función objetivo) de la GAN muestra una dinámica competitiva entre el Generador y el Discriminante. Para la implementación de la GAN se toma como Generador (G) el modelo básico y para el Discriminador (D) el estegonalizador. El modelo ISGAN incluirá estos elementos y tendrá como función objetivo la siguiente ecuación:

$$\min_G \max_D = E_{x\sim P(x)}[logD(x)] + E_{x\sim P(x),s\sim P(s)}\left[logD\left(1 - D\big(G(x,s)\big)\right)\right], \quad (1)$$

donde: x es la imagen de portada, s es la imagen secreta, $G(x,s)$ es la imagen estego generada y P es la base de datos que contiene las imágenes.

Para poder calcular los índices de perdida que existen en el modelo, los autores de ISGAN proponen una función llamada mixta, basada en el Modelo Visual Humano (HSV), el cual básicamente evalúa los cambios en el color, contraste y textura. Las métricas implementadas en la función son: Índice de Similitud Estructural (SSIM) [19] y su variante el Índice de Similitud Multi Escalar (MS-SSIM) [20].

Además del Error Cuadrático Medio (MSE) que permite conocer la diferencia cuadrada entre los datos de los pixeles de dos imágenes.

Dentro de la función mixta se utilizan diferentes hiper parámetros $\alpha = 0.5$, $\beta = 0.3, \gamma = 0.85$ los cuales permiten cambiar la calidad de las imágenes estego e imágenes secretas; esta función mixta está representada en la siguiente ecuación:

Tabla 1. Características principales de los códigos QR implementados.

Versión	# Módulos	Nivel de corrección	Capacidad
1	21 x 21	M	34
2	25 x 25	M	63
3	29 x 29	M	101
4	33 x 33	M	111

$$F_{mix} = \alpha * \big(1 - SSIM(x,y)\big) + (1-\alpha) * (1 - \text{MSSIM}(x,y)) + \beta * MSE(x,y), \quad (2)$$

donde: x es la imagen cubierta o la imagen oculta, y es la imagen estego o la imagen recuperada de acuerdo con el momento del entrenamiento que se encuentre.

Para medir los resultados en las imágenes esteganográficas se ocupan dos métricas importantes SSIM y PSNR. El índice SSIM permite modelar las semejanzas que hay entre dos imágenes en este caso la imagen de portada y estego, en base al color, contraste y textura, la evaluación se realiza de 0 a 1 entre más cercano este a 1 tendrá mejor exactitud la imagen que va a ser comparada.

El PSNR toma como base el MSE para conocer la potencia máxima de una imagen cubierta. El objetivo es conocer la distorsión en la señal de una imagen, en este caso después de haber pasado por un proceso compresión o modificación, por lo regular para imágenes deben representar los pixeles usando 8 bits (0 a 255) esto es indispensable para evitar problemas en el formato de la imagen. Las unidades de PSNR son decibelios (dB) y ante la ausencia de ruido el valor será igual a 0.

La clasificación PSNR para una imagen se realiza con el siguiente criterio: $PSNR <$ 30 dB es no aceptable, $PSNR$ entre 30 a 40 dB es aceptable y $PSNR >$ 40 dB es muy buena. Para el proceso esteganográfico en la métrica SSIM un valor mayor a 0.90 es excelente y para el PSNR se mantiene el valor mayor a 30 dB.

2.3. Propuesta de implementación

Los autores del modelo ISGAN utiliza tres diferentes bases de datos para su implementación y pruebas: *Lfw people, ImageNet* y *Pascal VOC12*, con estos elementos mostró resultados importantes medidos por las métricas SSIM y PSNR. En la implementación para la propuesta de este proyecto se utiliza una base de datos retomada de [21], la cual se amplió a 12,000, de imágenes RGB de 256 x 256 de códigos QR con información contenida de un número elegido pseudoaleatoriamente entre 1 al 10,000, con un nivel de corrección medio (*M*) y versiones de la 1 a la 4.

Para el entrenamiento se usaron 10,000 imágenes y otras 2,000 imágenes como conjunto de prueba, en la Tabla 1**,** se muestra las principales características del código QR implementado según su versión.

El código del modelo ISGAN fue tomado del repositorio de *GitHub* [22] el cual está preparado para trabajar en un ambiente desarrollado con las bibliotecas *Tensorflow 1.15*

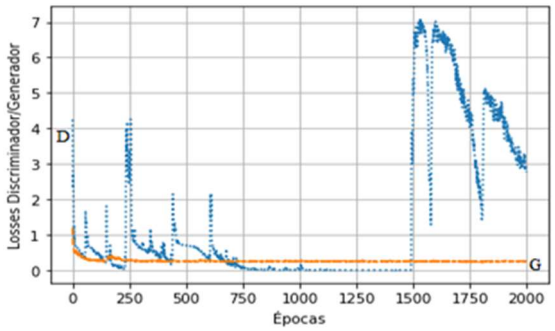


Fig. 2. Grafica de los índices de pérdida del Generador (G) y del Discriminador (D) para el modelo ISGAN en el entrenamiento.

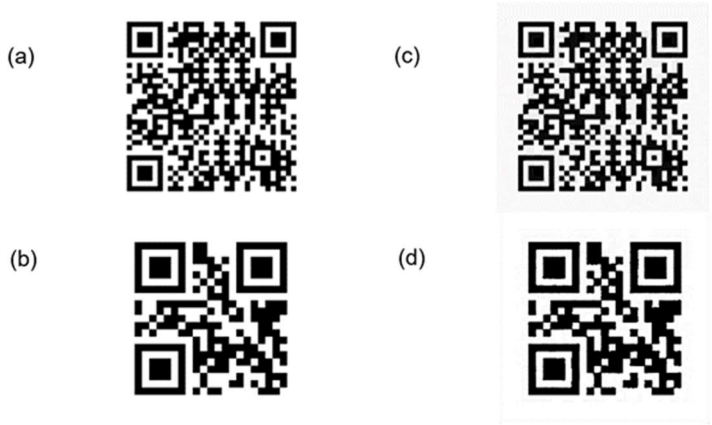


Fig. 3. Esteganografía de códigos QR Embebidos: a) QR de cubierta b) QR Estego c) QR Oculto (Secreto), d) QR Recuperado.

y *Keras 2.2.3* las cuales son de código abierto y permiten el desarrollo y entrenamiento de modelos para aprendizaje profundo.

El entrenamiento se realizó por 1,500 épocas utilizando lotes de 4 imágenes por época como se muestra en la Fig. 2. Los cambios en los *valores de pérdida* del Generador y Discriminador durante las épocas 0 a 700, muestran perdidas en el Discriminador altas no manteniendo un ritmo constante de descenso; en la época 750 se puede observar un descenso en los índices de perdida y este se mantiene contante.

En el caso del Generador, la perdida se mantiene constante desde la época 100, después de cruzar el umbral de 1,500 épocas muestra sobre entrenamiento parando en ese momento el entrenamiento de la GAN

Para evaluar el desempeño del discriminador se utilizó la prueba por medio del método predict implementado en la librería Keras. Al realizar el estegoanálisis las imágenes si no presentan esteganografía tendrán un valor de -1 del caso contrario un valor cercano a 1.

Al realizar las pruebas con imágenes estego obtenidas al finalizar el entrenamiento se obtuvieron valores en el estegoanálisis en el rango de 0.4 a 0.6 demostrando que el generador pudo confundir al estegoanalizador tal como debe funcionar una GAN.3

3. Resultados experimentales

En la Fig. 2, se muestran los resultados de la esteganografía realizada por el Modelo ISGAN cabe resaltar que el método esteganográfico utilizado en la red es el método de modificación de cubierta.

La primera impresión al analizar las imágenes de la Fig. 3**,** solo visualmente, se observa que la imagen estego sufrió una leve decoloración grisácea en el fondo, el cual debería de ser blanco, característico en la codificación QR, pero se alcanza a observar como la codificación no sufrió cambios importantes; la prueba de lectura se realizar sin ningún problema en una escala de 1:10. En el fondo no se alcanza a observar ningún tipo de sobre exposición que pueda hacer pensar que existe algún elemento oculto en ella.

En la comparación entre la imagen estego y la imagen recuperada se pueden ver cambios en la definición de los patrones de codificación a pesar de ello la lectura se puede realizar satisfactoriamente en la escala promedio de 1:10.

Para evaluar los resultados de las imágenes estego y las imágenes recuperadas se ocuparon las métricas propuestas por los autores de ISGAN que son el índice SSIM y PSNR, para enriquecer la evaluación se propone utilizar también el Índice de Calidad de Imagen Universal (UIQI) [23] y Fidelidad de la información Visual (VIF) [24] ya que es importante en los códigos QR conocer la calidad de la imagen para poder realizar una lectura y decodificación eficiente de la información.

La métrica UIQI tiene la capacidad para medir la pérdida de información ocurrida durante los procesos que impliquen la degradación de la imagen tomando como base tres índices: *pérdida de correlación*, *distorsión de luminancia* y *distorsión de contraste*. El rango de evaluación en la calidad de imagen es dinámico, va de -1 a 1, así entre más cercano es a 1, mejor será la calidad de la imagen evaluada.

La métrica VIF toma algunos conceptos de la teoría de la información como el modelado de escenas naturales (NSS) y el sistema visual humano (HVS) permite realizar una predicción de la calidad de una imagen dependiendo de la cantidad de información que compartan una imagen denominada perfecta y otra distorsionada, la evaluación de la imagen será de 0 a 1 entre más cercana sea a uno más características han podido ser extraídas de la imagen con distorsión

El valor idóneo en al evaluar imágenes esteganográficas es que los índices para las métricas VIF y UIQI superen el 0.90.

En la Tabla 2, se muestra los resultados para las métricas antes mencionadas la evaluación se llevó a cabo con los resultados de ISGAN en la época 1500 de donde se realizó la predicción para 10 imágenes de cubierta y 10 imágenes secretas tomadas de forma pseudoaleatoria de la base de datos para evaluación, los resultados se muestran en base al promedio de los resultados de los índices para cada imagen cubierta vs imágenes estego e imágenes ocultas vs imágenes recuperada.

Tabla 2. Índices para la evaluación de imágenes recuperadas y estego.

Imagen Secreta vs Imagen Recuperada			
SSIM	UIQI	PSNR (db)	VIF
0.9696	0.9521	22.3126	0.6182
Imagen Cubierta vs Imagen Estego			
SSIM	UQI	PSNR	VIF
0.9716	0.9687	30.4258	0.6205

De acuerdo con la Tabla 2**,** al comparar la imagen secreta vs la imagen recuperada el índice SSIM muestra resultados satisfactorios al presentar un valor muy cercano a 1 por lo que se puede inferir que no sufrió cambios muy relevantes en la luminancia ni tampoco presenta distorsión en la textura de la imagen.

El índice UIQI muestra un valor muy alto en la calidad de la imagen recuperada no mostrando degradación de la información en ella. Las métricas antes mencionadas son importantes ya que al tener la información completa con índices bajos de distorsión permite una precisa lectura y decodificación del código QR recuperado, puesto que esta información es la que podría fungir como una marca de agua invisible, es importante recuperar la información completa.

La métrica PSNR muestra en valor inferior a los 30db por lo que entonces se puede inferir que las distorsiones en la señal de la imagen recuperada fueron muy altas y estas muestran degradación a pesar de que visualmente no se observan cambios substánciales, sin embargo, este índice muestra una ventana de oportunidad para mejorar la red decodificadora y evitar estos problemas en cuanto a la señal para la recuperación de la imagen secreta.

El índice VIF muestra resultados malos y estos concuerdan con lo observado a simple vista en la imagen recuperada donde a pesar de recuperar la información completa los bloques de codificación QR muestran pequeños cambios. Esto demuestra que el nivel de corrección medio empleado en el código QR es suficiente para lograr una lectura y decodificación de la imagen recuperada no necesitando escalar a un nivel de corrección mayor.

De acuerdo con la Tabla 2 en los índices para la comparación de la Imagen Cubierta vs Imagen Estego, se puede observar que el índice SSIM muestra un valor alto muy cercano a 1, en aspectos estenográficos, indica que la imagen estego muestra gran semejanza con la imagen de portada a pesar de la modificación que se realizó en el canal de luminancia evitando problemas en el contraste y textura de la imagen estego.

La métrica UQI muestra un alto nivel de calidad ya que el índice es muy cercano al valor máximo 1, esto indica que no hay perdida de información substancial en la imagen estego. Como la imagen estego es la cara que se muestra a nivel público, es importante mantener una calidad alta para evitar las sospechas de modificación.

El índice PSNR muestra un valor apenas superior a 30dB por lo que se puede considerar aceptable, esto indica que a pesar de las modificaciones en las señales se pudo conservar varias características esenciales en la imagen estego, se puede señalar

que la seguridad de la imagen no está comprometida, sin embargo, es importante mejorar este aspecto para elevar los índices a valores superiores a 40dB para evitar la detección de información por modelos de estegoanálisis basados en aprendizaje profundo.

El índice VIF muestra resultados malos debido a la pequeña decoloración del fondo de la imagen estego, que, si bien no es substancial, si es perceptible.

4. Conclusiones

La esteganografía en las últimas décadas ha tenido un desarrollo importante y gracias a su implementación en las redes de generativas adversarias ha mejorado substancialmente.

Los códigos QR ya son parte de nuestra vida cotidiana, se encuentran en todo tipo de publicidad, se pueden realizar pagos por medio de distintas aplicaciones bancarias, pueden contener información personal como tarjetas de presentación entre otras, por lo que se puede utilizar elementos estenográficos que permitan aumentar su nivel de seguridad.

Con base en los resultados presentados en el modelo ISGAN, se puede observar que la técnica es eficiente para ocultar un QR en otro, porque a pesar de las modificaciones que se realizaron en la imagen QR de portada, los datos codificados no muestran alteraciones en el contraste y textura, permitiendo su lectura y decodificación. Además, la recuperación de la imagen secreta se logró sin cambios considerables en su estructura por lo que la codificación de esta información es recuperable.

La imagen estego sufre una pequeña decoloración, sin embargo, no influye en el proceso de lectura de la información del código QR.

Este modelo implementado muestra una gran capacidad para poder almacenar una imagen del mismo tamaño en tonos de grises porque la información no necesita ningún proceso de compresión.

De acuerdo con las métricas SSIM, UQI y VIF sus valores máximos que se pueden obtener al comparar dos imágenes es 1, lo cual significa que al comparar dos imágenes se alcanzó la igualdad, sin embargo, en esteganografía se puede alcanzar solo similitud en un porcentaje que puede variar de acuerdo con la técnica utilizada. Los valores obtenidos en la Tabla 2 para el modelo ISGAN demuestra que los índices SSIM y UQI son cercanos a uno acercándose al máximo superado el valor idóneo de 0.90 sin embargo el valor de la métrica VIF no alcanzo el umbral antes mencionado llegando solo a un valor de 0.60.

Por lo anterior, el modelo de ISGAN logro altos niveles de similitud entre las imágenes cubierta y estego lo que significa alto grado de imperceptibilidad; así como al comparar la imagen secreta con la recuperada, indicando alta robustez, al embeber una imagen con código QR en otra. Lo cual permite elevar la seguridad si se tiene información que pudiera ser susceptible y no quisiera ser vista por terceros o para verificación de donde proviene la información y validar la su autoría.

Referencias

1. Byrnes, O., La, W., Wang, H., Ma, C., Xue, M., Wu, Q.: Data hiding with deep learning: a survey unifying digital watermarking and steganography. (2021) arXiv preprint arXiv:2107.09287
2. Yahya, A.: Steganography techniques for digital images. Palapye, Botswana: Springer (2019)
3. Zhang, C.: Benz, P., Karjauv, A.: UDH: Universal deep hiding for steganography, watermarking, and light field messaging. In: 34th Conference on Neural Information Processing Systems, vol. 33, pp. 1–12 (2020)
4. Luque, J.: Códigos QR. Manual formativo de ACTA, vol. 63, pp. 9–28 (2012)
5. Denso Wave Incorpored.QRCode®. Essentials.: [En línea]. Available: https://www.qrcode.com/en/about/version.html. [Último acceso: 1 abril 2021] (2021)
6. Hernandez, R., Flores, J., Landassuri, V., Morales, S., Rodriguez, I.: Análisis de calidad en imágenes esteganográficas aplicando el algoritmo LSB en códigos QR embebidos. Research in Computing Science, vol. 150, no. 5 (2021)
7. Bhoskar, N., Ithape, P.: A Survey on secrete communication through QR code steganography for military application. International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 10 (2022)
8. Mathivanan, P.: QR code based color image stego-crypto technique using dynamic bit replacement and logistic map. Optik, vol. 225 (2021)
9. Goodfellow, I. J., Pouget-Abadie, J., B, X. B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
10. Fu, Z., Wang, F., Cheng, X.: The secure steganography for hiding images via GAN. EURASIP Journal on Image and Video Processing, no. 1, pp. 1–18 (2020)
11. Zhu, L., Yu, G., Mo, L., Zhang, D.: DGANS: Robustness image steganography model based on double GAN. Journal on Communications, vol. 41, no. 1, pp. 125–133 (2020)
12. Corley, I., Lwowskiy, J., Hoffmanz, J.: Destruction of image steganography using generative adversarial networks. arXiv preprint arXiv:1912.10070 (2019)
13. Zhang, R., Dong, S., Liu, J.: Invisible steganography via generative adversarial networks. Multimedia tools and applications, vol. 7, pp. 8559–8575 (2018)
14. Baluja, S.: Hiding images in plain sight: Deep steganography. In: Advances in Neural Information Processing Systems, vol. 30, pp. 2069–2079 (2017)
15. Atique, R., Rafia, R., Shahroz, N., Sibt, H.: End-to-end trained CNN encoder-decoder. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–15 (2015)
17. Xu, G.: Deep convolutional neural network to detect J-UNIWARD. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 67–73 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional. IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916 (2015)

19. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612 (2004)
20. Wang, Z., Simoncelli, E., Bovik, A.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, vol. 2, pp. 1398–1402 (2003)
21. Dieckhaus, C.: Kaggle [En linea]. Available: https://www.kaggle.com/datasets/coledie/qr-codes. [Último acceso: 21 abril 2022] (2022)
22. Lefranc, M.: GitHub [En linea]. Available: https://github.com/Neykah/isgan. [Último acceso: 21 abril 2022] (2022)
23. Wang, Z., Bovik, A.: A universal image quality index. IEEE signal processing letters, vol. 9, no. 3, pp. 81–84 (2002)
24. Sheikh, H. R., Bovik, A.: Image information and visual quality. IEEE Transactions on image processing, vol. 15, no. 2, pp. 430–444 (2006)

ISSN 1870-4069

Identificación de un individuo mediante la voz utilizando redes convolucionales y umbrales de aceptación

José Luis Medina Jiménez, Héctor Rodríguez Rangel,
Gloria Ekaterine Peralta Peñuñuri, Mario Alberto Román Garay,
Luis Alberto Morales Rosales

Tecnólogico Nacional de México Campus Culiacán,
División de posgrado,
México

Conacyt-Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Civil,
México

{jose_medinaj,mario_roman}@itculiacan.edu.mx,
{hector.rr,gloria.pp}@culiacan.tecnm.mx,
lamorales@conacyt.mx

Resumen. En la actualidad el uso de usuario y contraseña como método tradicional de acceso es un problema debido a que son fáciles de olvidar y no proporciona la certeza de que los datos de los usuarios están seguros. Por otro lado, el robo de identidad es otro de los problemas principales que se desea evitar. El robo de identidad en México y en el mundo ha crecido en los últimos años debido a la pandemia, ya que el uso de medios digitales ha aumentado por las medidas sanitarias que se han tomado. La exploración de la seguridad biométrica se ha ampliado gracias al avance de la inteligencia artificial y de los métodos de extracción de características, esto debido a que los rasgos biométricos son únicos y no cambian. Uno de los principales problemas de trabajar con la voz, es el timbre, en particular el tono, ya que de esta manera pueden existir voces muy similares en sus frecuencias capaces de confundir a los sistemas de identificación. En el artículo se describe un sistema de identificación de personas mediante la voz utilizando los espectrogramas. Además, mediante la aplicación de distintos umbrales se realizó una comparativa de estos, para reducir el impacto que tiene el timbre de voz al momento de identificar personas, llegando a obtener una precisión del 93 %.

Palabras clave: Acceso biométrico, reconocimiento de voz, aprendizaje profundo, red neuronal convolucional, VGGVox.

Identification of an Individual by Voice Using Convolutional Networks and Acceptance Thresholds

Abstract. Currently the use of username and password as a method traditional access is a problem because they are easy to forget and do not provide the

certainty that user data is safe. On the other hand, identity theft is another major problem you want to avoid. Identity theft in Mexico and in the world has grown in recent years due to the pandemic, since the use of digital media has increased due to the sanitary measures that have been taken. The exploration of biometric security has been expanded by the advancement of artificial intelligence and feature extraction methods, because biometric features are unique and do not change. One of the main problems of working with the voice is the timbre, in particular the tone, since in this way there can be voices that are very similar in their frequencies, capable of confusing the identification systems. The article describes a system for identifying people by voice using spectrograms. In addition, through the application of different thresholds, a comparison of these was made, to reduce the impact of the voice timbre when identifying people, reaching an accuracy of 93%.

Keywords: Biometric access, voice recognition, learning deep, convolutional neural network, VGGVox.

1. Introducción

Conforme ha crecido la era digital, el robo de identidad ha aumentado junto con ella, de tal manera que la falsificación de datos de personas ha afectado a millones. Durante el 2020, cuando dio lugar el inicio de la pandemia, el confinamiento aumentó el uso de internet, así como aumentaron los casos de falsificación con respecto a años anteriores. Según datos del Banco de México (Banxico), el país se encuentra en el octavo lugar en el mundo en delito de robo de identidad y es el segundo lugar en Latinoamérica, estimando que el 67 % es por perdida de documentos[1].

Actualmente, los métodos tradicionales como el usuario y contraseña ya no son suficientes, ya que son fáciles de olvidar o de robar para la suplantación de identidad, por ello los rasgos biométricos se utilizan en distintos sectores de la industria. La finalidad de utilizar los datos biométricos como sistema de acceso, es aumentar la seguridad y privacidad del usuario, ya que estos rasgos son universales y únicos [2].

La Biometría se define como las características únicas con las que cuenta un individuo. Estas se clasifican en dos categorías: físicas y de comportamiento. Las físicas engloban aquellas como la huella dactilar, iris, geometría de la mano y rostro y, por otro lado, tenemos las características de comportamiento, entre las cuales están, la firma, la escritura, la voz, el andar [4].

El reconocimiento de voz es una gran alternativa para la nueva modalidad que se encuentra actualmente la sociedad, ya que a diferencia de la huella dactilar, no es necesario tener contacto directo con los sensores.

El aprendizaje profundo se ha vuelto ampliamente utilizado para el reconocimiento de voz, debido a que estas técnicas permiten entrenamientos más rápidos y manejo de grandes datos, haciendo más eficiente la tarea de clasificación de voz [3]. Las redes profundas más utilizadas son redes convolucionales, redes recurrentes, memoria a corto plazo o LSTM por sus siglas en inglés, otras arquitecturas de aprendizaje profundo utilizadas son los autocodificadores y las redes generativas-adversarias (GAN) [4].

Los sistemas de reconocimiento de voz se pueden visualizar o separar en dos distintos enfoques. Por un lado, está el enfoque de los modelos tradicionales, llamados así, ya que pertenecen a modelos matemáticos-estadísticos, antes de la llegada de las redes neuronales profundas.

Estos modelos son los HMM (Hidden Markov Model), GMM (Gaussian Mixture Model) algoritmos basados en SVD (Singular Value Decomposition), DCT (Discrete Cosine Transform), Enfoque de agrupamiento iterativo, índice de probabilidad, entre otros derivados de los mencionados anteriormente. Ahora, por parte del enfoque del Aprendizaje profundo, algunos ejemplos son, DNN (Deep Neural Network), CNN (Convolutional Neural Network), SVM (Support Vector Machines) y entre otros derivados de estos mismos.

A manera general, un sistema biométrico tiene cuatro etapas. La primera es la de registro del usuario mediante el uso de un sensor apropiado ("Enrollment"). La segunda etapa se encarga de extraer las características del dato biométrico de entrada ("Feature Extractor") guardándola como plantilla en una base de datos.

La siguiente etapa es la nueva entrada del dato biométrico, donde se le realiza el proceso de extracción de característica y se compara con la que está guardada en la base de datos ("Matcher"). Por último, se hace la toma de decisión de verificación o autentificación de la persona [5].

La identificación de voz es una rama del reconocimiento de voz en donde se procesa una muestra de voz desconocida y compara con una base de datos de personas establecidas, es decir, una medición de 1 contra N personas. La voz desconocida se identifica como la que mejor se adapte, de esta manera la entrada para la identificación de voz es una voz desconocida y la salida es el nombre o la identificación del usuario [7].

Uno de los principales problemas que existe en la identificación de personas mediante la voz, es el timbre que tiene la voz, debido a que esta característica trabaja con frecuencias. Al tener una comparativa de 1 contra N personas, entre más grande sea el valor de N o dicho de otro modo entre más personas, sea con las que se compare la voz de entrada, los sistemas de identificación pueden confundirse detectando de manera errónea a otra persona.

Ante estas problemáticas, en este artículo se propone la implementación de un sistema de identificación de voz empleando Redes Neuronales Convolucionales utilizando como datos de entrada espectrogramas, es decir, trabajamos en el dominio de la frecuencia considerando el timbre de voz. Además, se hace un análisis de distintos umbrales para encontrar el más adecuado que permita identificar de manera correcta mediante la voz a un individuo.

Para ello, se realizaron dos pruebas distintas: 1) la identificación se realiza repitiendo la misma frase de registro del individuo y 2) se utiliza la frase de registro del individuo para comparar su similitud de la voz con una frase distinta mencionada.

A partir de estas pruebas, se llega a la conclusión que el uso de umbrales bajos para la medición de la similitud de la voz permite una mayor fiabilidad en la identificación de personas mediante la voz, obteniendo así un 93 % precisión.

El resto del artículo está organizado de la siguiente manera: Estado del arte, Materiales y Métodos, Resultados, Conclusiones y Trabajos Futuros.

2. Estado del arte

Durante los años 30, Francés McGehee se inspiró en un caso de secuestro y asesinato para el desarrollo del reconocimiento de voz, ya que uno de los afectados reconoció la voz del secuestrador. La idea de McGehee fue realizar una investigación de que tan fiable es el oído humano, que posteriormente esto daría partida a un tipo de investigación forense y psicológica [8].

Actualmente, este tipo de investigación de reconocimiento de voz sigue gracias a la Inteligencia Artificial, enfocado en técnicas de Aprendizaje-Máquina y Aprendizaje Profundo. Esto debido a que se han explorado distintos extractores de características tales como la Transformada Discreta de Wavelet donde principalmente sé utilizando los Modelos Gaussianos Mixtos (GMM) y el Perceptrón Multicapa (MLP) para el reconocimiento de voz [8].

Por otro lado, tenemos los Coeficientes Cepstrales en las Frecuencias de Mel utilizándose en los mismos modelos anteriores y además en DNN, SVM y CNN [8]. Existen otras técnicas de extracción con Vectores X, espectrogramas, características espectrales dinámicas normalizadas, Coeficientes cepstrales temporales basados en la energía de Teager, además de las combinaciones de estas [8].

En el trabajo de [9] proponen un sistema de verificación de voz dependiente de texto, donde se utiliza una DNN supervisada para la extracción de características a nivel cuadro, de esta manera cada cuadro se va apilando en un vector de izquierda a derecha, correspondiendo al número de voces de entrenamiento por cada cuadro.

En el artículo [10] se utiliza una modificación de la DNN, donde se emplea la combinación de un autocodificador con DNN, esto con la finalidad de mejorar el audio, donde la función principal del autoencoder es la eliminación del ruido y la reverberación.

Un modelo enfocado en CNN propuesto en [11] diseñado para optimizar el proceso de identificación de voz, basado en el dataset TIMIT. Para el preprocesamiento de datos se utilizan espectrogramas para mejorar las fuentes acústicas. Esta CNN contiene varias capas de convolución aplicando varios filtros a diferentes secciones locales de entrada y a su vez esa capa le sigue una capa de agrupación máxima, de esta manera emite una versión de más baja resolución de las activaciones de la capa de convolución eliminando la activación total del filtro.

Dentro de la categoría de las CNN existen las redes convolucionales 3D, donde el kernel encargado de la convolución se mueve en 3 direcciones. En el trabajo [12] se propone la utilización de las 3D-CNN, donde estas 3 dimensiones, además del dominio de la frecuencia como en otros trabajos, también se enfoca en el dominio del tiempo y el tamaño de enunciados que existe en un audio, con la finalidad de construir un modelo más robusto para la problemática de los cambios que existen en la voz.

Las Redes Siamesas están basadas en un modelo CNN, propuesto en [7], este trabajo consiste en dos entradas simultáneas aprendiendo una función de similitud y así muestra lo idénticas que son las dos entradas. Por lo tanto, el objetivo de la red siamesa no es clasificar la voz, sino distinguir o conocer la similitud entre las dos voces de entrada.

En el trabajo [13] proponen la utilización de la Máquina de Soporte Vectorial, enfocado en tratar el reconocimiento de voz como un problema de clasificación binaria.

Tabla 1. Características de micrófono HyperX QuadCast.

Características	Valores
Velocidad de muestreo/bits	48kHz/16-bit
Patrones polares	Estéreo, omnidireccional, cardiode, bidireccional
Respuesta de frecuencia	20Hz-20kHz
Sensibilidad	-36dB(1V/Pa a 1kHz)

Durante el proceso de reconocimiento de voz, se aplica el clasificador entrenado en distintos puntos para reconocer si la voz coincide o no. Existen algunas técnicas más recientes donde se proponen variantes y mejoras de extractores de características, ya que estas definen que tan bien van a detectar a la persona que habla.

En el artículo [14] se menciona como se combina el uso de los MFCC con características basadas en el tiempo, convirtiéndola en MFCCT, con la finalidad de mejorar la precisión de los sistemas de Identificación de Voz.

3. Materiales y métodos

Durante el desarrollo del sistema de identificación de voz, fue necesario utilizar un dispositivo de entrada de audio y un servido que se encargue de procesar y realizar el trabajo de identificación de la voz. Se describirá a continuación el hardware utilizado y posteriormente se mostrará a detalle los procesos que se realizan por parte del sistema para realizar la tarea de identificación de voz.

3.1. Hardware

Los sistemas de reconocimiento de voz como método de seguridad para cualquier tipo de acceso, siendo una buena alternativa o inclusive para realizar la fusión de sistemas que utilicen otro tipo de reconocimiento biométrico y así robustecer el método de acceso. En cualquier sistema biométrico es necesario contar con el sensor o dispositivo de entrada para capturar el rasgo de biometría a utilizar. En el caso del reconocimiento de voz se debe contar con un micrófono para capturar la voz.

Para tener un sistema controlado y no exista variaciones en los audios, se utilizó un micrófono HyperX QuadCast, ya que este cuenta con características ideales para tener mayor control en las ganancias de los audios y así evitar la mayoría de los ruidos ambientales o inclusive las voces de fondo que puedan entorpecer el proceso de reconocimiento de voz.

Para el sistema de reconocimiento de voz, es necesario que el audio cuente con ciertas características, como una tasa de 16 – Bit, audio mono y que tenga una frecuencia de 16 kHz, para esto se muestra las características del micrófono en la Tabla 1, la cual cuenta con los requisitos necesarios para nuestro sistema.

El procesamiento de software se realiza con un servido, la cual cuenta con las características necesarias para realizar la tarea de identificación de voz sin ningún problema. Las características esenciales para realizar el proceso se muestran en la Tabla 2.

Tabla 2. Características del servido.

Nombre	Características
Sistema Operativo	Windows 11
RAM	16 GB
Almacenamiento	500 GB SSD
Procesador	Intel Core i7 9° Gen
Tarjeta Grafica	Nvidia RTX 2060

3.2. Sistema de identificación de voz

El Sistema de Identificación de voz se divide en dos etapas, la etapa de registro de usuario y la etapa de identificación mediante la voz, las cuales a continuación se van a explicar sus procesos.

Etapa de registro de usuario

En la etapa de Registro, se introduce la voz de la persona a la base de datos con sus respectivos pre-procesamientos y se almacena, tal como se muestra en la Figura 1.

– **Frase de registro:** En la etapa de registro el audio llega en "crudo", es decir, que llega en un formato tipo "WAV", a una frecuencia de 16 kHz, una tasa de muestreo a 16-Bits y en canal mono. Debido a que el modelo fue entrenado con audios con las características descritas anteriormente, es necesario que los registros y las entradas de audio cumplan con este requisito.

  Ahora, para realizar el registro fue necesario que el usuario usara una frase en concreto de una duración alrededor de 5 segundos. La frase utilizada fue "El reconocimiento de voz es la llave de acceso al futuro como contraseña", esto con la finalidad que se hicieran pruebas con frases iguales y frases distintas y verificar si existía una variación en precisión entre usar la misma frase a una distinta.

– **Extractor de espectrogramas:** A partir de obtener el audio en "crudo", durante la etapa de extracción de espectrograma se realiza un preprocesamiento en el audio en el cual mediante la utilización de la Transformada Rápida de Fourier pasa del dominio del tiempo a dominio de la frecuencia.

  Una vez transformado el audio al dominio de la frecuencia, se extrae una imagen del audio, el cual representa el espectrograma como se puede observar en la Figura 2, esto para posteriormente pasar al modelo utilizado para obtener el vector de características, es decir, el "embedding".

– **Generación de embeddings:** El modelo utilizado para la extracción del vector de características fue del trabajo [15] llamada VGGVox, el cual está basada en una red convolucional VGG-M enfocada en la clasificación de imágenes, se hace una pequeña modificación donde se recibe como entrada las imágenes de espectrogramas extraídas de los audios. En la Figura 3 se muestra la estructura general de la Red neuronal Convolucional VGGVox.

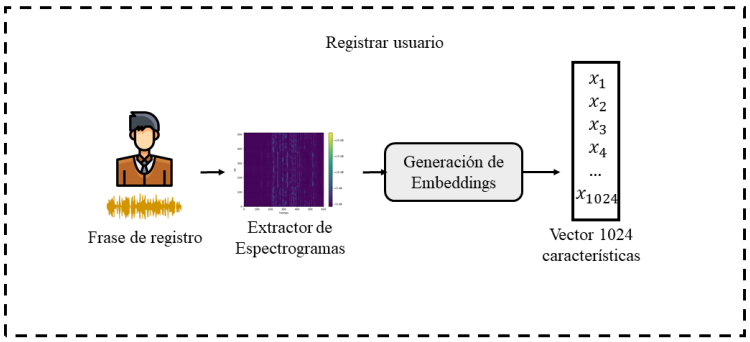


Fig. 1. Diagrama de registro de voz.

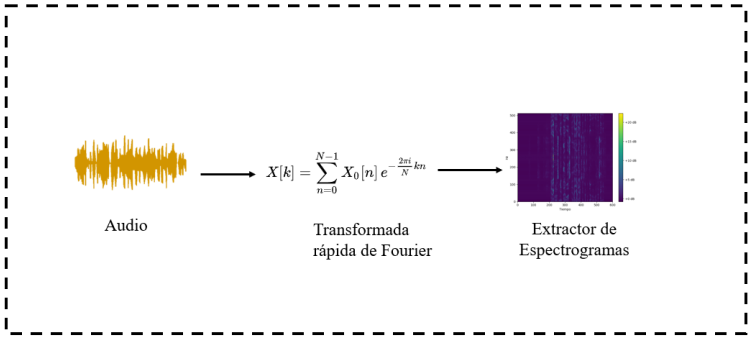


Fig. 2. Espectrogramas.

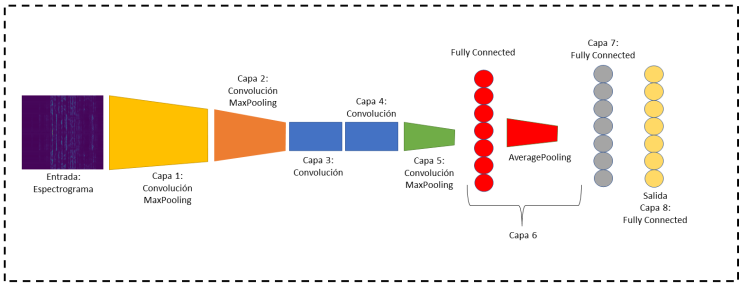


Fig. 3. Red Neuronal Convolucional VGGVox.

En la Tabla 3 se detalla la arquitectura de la VGGVox donde los autores hacen una modificación de la VGG-M, agregando a la capa “fully connected” (fc6) una capa de “average pool” el cual tiene un kernel que varía dependiendo de los segundos del audio esto para tener la flexibilidad de meter audios de diferentes tiempos. Observando la Tabla 3 de la arquitectura, en la última capa la dimensión de filtro es la salida entregada de la red VGGVox.

Tabla 3. Arquitectura VGGVox.

Capa	Kernel	Dimensión de filtro	Filtros	Stride	Tamaño de datos
conv1	7x7	1	96	2x2	256x148
mpool1	3x3	-	-	2x2	126x73
conv2	5x5	96	256	2x2	62x36
mpool2	3x3	-	-	2x2	30x17
conv3	3x3	256	384	1x1	30x17
conv4	3x3	384	256	1x1	30x17
conv5	3x3	256	256	1x1	30x17
mpool5	5x3	-	-	3x2	9x8
fc6	9x1	256	4096	1x1	1x8
apool6	1xn	-	-	1x1	1x1
fc7	1x1	4096	1024	1x1	1x1
fc8	1x1	1024	1251	1x1	1x1

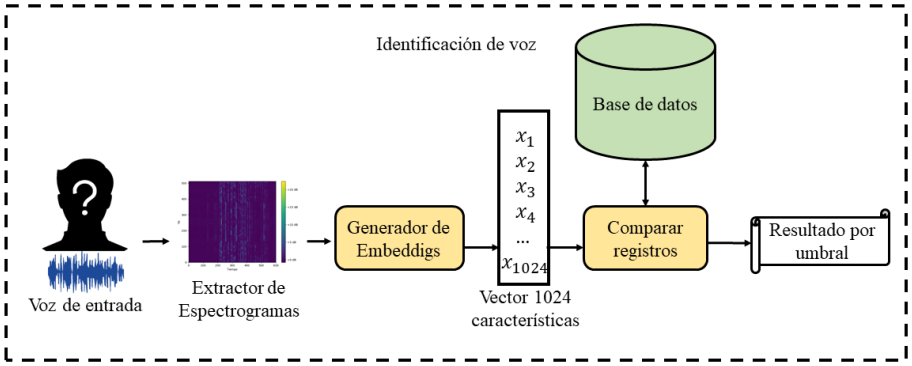


Fig. 4. Diagrama de identificación de voz.

– **Vector de características:** Una vez que el usuario realiza el proceso de grabar la frase con su voz, al pasar por el proceso de extracción de espectrograma y se obtiene un vector de características denominado “embedding” el cual cuenta con 1024 características extraídas de la imagen del espectrograma.

Etapa de identificación de usuario

La etapa de identificación mostrada en la Figura 4, llega una voz de entrada al sistema y este se encargará de clasificar esta voz entre los usuarios registrados en la etapa de registro, los cuales estos fueron almacenados en una base de datos.

– **Voz de entrada:** En la etapa de voz de entrada, tiene el mismo proceso que pasa al registrar un usuario, la persona introduce una frase con su voz, este pasa por una etapa de extracción del espectrograma y se extrae el “embedding”.

Durante la etapa de pruebas se recolectaron alrededor 768 audios de muestra de 128 personas, es decir, 6 audios por cada persona. De los 6 audios recopilados por personas, 3 eran con la finalidad de decir la misma frase que se hizo de registro y los otros 3 con frases distintas, para posteriormente separar estos dos grupos y someterlos al sistema.

– **Comparar registros:** Al tener el vector de características de la voz de entrada llega la etapa de comparar registros, es decir, identificar a que voz coincide con las registradas en la base de datos, es por eso que en esta etapa, al ser vectores de mismo tamaño de características y solo trabajar en un mismo plano (frecuencias) mediante la utilización de la ecuación 1 es considerada una de las métricas más utilizadas y de menor costo computacional al momento de comparar similitud entre vectores:

$$D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{1}$$

donde y representa el vector de entrada y x representa las voces registradas, esta última iterándose para medirse una a una con la voz a identificar y se guarda cada distancia en un vector para la siguiente etapa. La finalidad de la medición de la distancia euclidiana es que entre más cercana a 0 sea el valor obtenido, significa que más acertada es la identificación.

– **Resultado por umbral:** Al obtener el vector de distancias es necesario determinar si esa persona identificada es la correcta o no, es por eso que mediante un umbral se determina la identificación de la persona.

Al ir aumentando el tamaño de personas registradas, este umbral debe ir variando, es decir, ir disminuyendo este parámetro conforme va aumentando el registro de personas para así tener mayor tasa de precisión y evitar la identificación errónea.

Debido a que estamos utilizando la medida de distancia euclidiana para identificar a la persona, entre menor sea el umbral, significa que es más similar es la voz registrada a la de entrada al sistema, ya que el umbral va en función a los valores que se obtienen con la distancia euclidiana. Existen otras métricas que ayudan a conocer que tan fiable es el sistema de identificación.

La Exactitud que es interpretada por la tasa de porcentaje de la cantidad de predicciones positivas que fueron correctas, y se obtienen con la fórmula 2 ahora bien, para obtener el valor de precisión, el cual representa el porcentaje de casos positivos detectados, donde este porcentaje nos indica que tan fiable es el valor detectado como positivo, se define con la ecuación 3:

$$\text{exactitud} = \frac{(VP + VN)}{(VP + FP + FN + VN)}, \tag{2}$$

$$\text{precision} = \frac{VP}{(VP + FP)}. \tag{3}$$

Tabla 4. Matriz de confusión con audios diciendo la misma frase de registro.

Umbral	Verdaderos positivos	Falsos positivos	Verdaderos negativos	Falsos negativos
0.1	136	9	85	154
0.12	175	20	76	113
0.14	210	29	72	73
0.16	237	40	63	44
0.18	248	51	55	30
0.2	254	68	44	18
0.22	259	77	37	11
0.24	262	86	30	6
0.26	264	93	26	1
0.28	265	101	18	0
0.3	265	106	13	0

4. Resultados

En la Tabla 4 mediante la utilización de las 4 opciones que ofrece la matriz de confusión, se exploraron distintos umbrales para la toma de decisión de similitud de audios. Se obtuvieron los Verdaderos Positivos (VP) que son aquellos que el sistema detecta como verdaderos y su valor real es verdadero, Falsos Positivos (FP) se interpretan como los audios aceptados como una persona distinta, Verdaderos Negativos (VN) estos resultados se obtienen de personas que no están en el sistema y que realmente los rechaza y por último Falsos Negativos (FN) que son aquellos audios que si debieron asignarle un valor verdadero, pero que fueron rechazados por el sistema.

A partir de la Tabla 4 de los 4 puntos de la matriz de confusión, se determinaron las métricas por cada umbral de exactitud, precisión y F1 las cuales se pueden observar en la Tabla 5. En la Tabla 6 se muestra los resultados de la matriz de confusión para frases distintas con el mismo tamaño de audios, además que en la Tabla 7 se muestran las métricas obtenidas.

Al comparar resultados entre la Tabla 4 y 6 de matriz de confusión con la misma frase y frase distinta, se puede observar que existe una diferencia de aproximadamente un 20 % entre los Verdaderos positivos, es decir, que al utilizar la misma frase existe mayor exactitud a la hora de reconocer correctamente a la persona.

Por otro lado, un punto importante de la matriz de confusión en los sistemas biométricos son los falsos positivos, ya que estos determinan cuantas personas identificó erróneamente.

Al comparar ambas pruebas conforme va creciendo el umbral en el sistema empieza aumentar los falsos positivos, teniendo que la diferencia entre el umbral más bajo, no hay una diferencia significativa, pero al utilizar un umbral alto, como es el caso de 0.3 la diferencia entre ellos es casi de 15 %. Por lo tanto, no es conveniente en los sistemas biométricos una tasa alta en falsos positivos, ya que esto implica asignar erróneamente un valor correcto a una persona.

Por parte de las métricas de exactitud y precisión en ambas pruebas de las Tablas 5 y 7, se puede observar que el sistema es preciso con umbrales bajos, esto debido a que entre más bajo sea el umbral más cerca está de ser exacta a la voz de la base de datos.

Tabla 5. Métricas obtenidas de la matriz de confusión de audios diciendo la misma frase de registro.

Umbral	Exactitud	Precisión	F1
0.1	57.55 %	93.79 %	62.53 %
0.12	65.36 %	89.74 %	72.46 %
0.14	73.44 %	87.87 %	80.46 %
0.16	78.13 %	85.56 %	84.95 %
0.18	78.91 %	82.94 %	85.96 %
0.2	77.60 %	78.88 %	85.52 %
0.22	77.08 %	77.08 %	85.48 %
0.24	76.04 %	75.29 %	85.06 %
0.26	75.52 %	73.95 %	84.89 %
0.28	73.70 %	72.40 %	83.99 %
0.3	72.40 %	71.43 %	83.33 %

Tabla 6. Matriz de confusión con audios diciendo distinta frase a la de registro.

Umbral	Verdaderos positivos	Falsos positivos	Verdaderos negativos	Falsos negativos
0.1	60	5	90	227
0.12	96	21	86	179
0.14	125	41	78	138
0.16	149	64	69	100
0.18	173	93	55	61
0.2	189	112	47	34
0.22	196	128	34	24
0.24	200	141	26	15
0.26	205	149	21	7
0.28	207	156	15	4
0.3	208	160	11	3

La precisión en este caso indica la fiabilidad o certeza con la que el sistema está clasificando el valor positivo, es decir, para evitar que el sistema se confunda por la similitud en las personas en su timbre de voz, es mejor utilizar umbrales bajos para así asegurar mayor precisión a la hora de identificar a la persona de manera correcta. Es por eso que, el usar el umbral de 0.1 nos entrega un desempeño del 93 %.

Por último, la métrica F1 simplifica las medidas de precisión y exhaustividad, donde va de 0 a 1, o 0 a 100 en porcentaje, siendo 1 (100) el mejor caso. El decir la misma frase tiene mejor rendimiento en el sistema, evaluándose con la medida F1 debido a que hay mayores verdaderos positivos a la hora evaluarse el sistema.

En la Tabla 8 se compara el desempeño del trabajo de este artículo con respecto los trabajos de otros autores mostrados en el estado del arte. Se pueden observar las distintas características de entrada y los tipos de extractores de características, ya que estos elementos son clave para obtener una tasa alta de desempeño.

Tabla 7. Métricas obtenidas de la matriz de confusión de audios diciendo distinta frase a la de registro.

Umbral	Exactitud	Precisión	F1
0.1	39.27 %	92.31 %	34.09 %
0.12	47.64 %	82.05 %	48.98 %
0.14	53.14 %	75.30 %	58.28 %
0.16	57.07 %	69.95 %	64.50 %
0.18	59.69 %	65.04 %	69.20 %
0.2	61.78 %	62.79 %	72.14 %
0.22	60.21 %	60.49 %	72.06 %
0.24	59.16 %	58.65 %	71.94 %
0.26	59.16 %	57.91 %	72.44 %
0.28	58.12 %	57.02 %	72.13 %
0.3	57.33 %	56.52 %	71.85 %

Tabla 8. Sistemas desarrollados del estado del arte.

Autores	Base de datos utilizada	No. De hablantes	Entrada	Modelo	Desempeño
Variani et al. (2014) [9]	NA	646	Características energéticas del marco	DNN	EER : 2.00 (Por 20 expresiones)
Lukic et al. (2016) [11]	TIMIT	630	Espectrograma de datos de voz	CNN	AC: 97
Plchot et al. (2016) [10]	PRISM: Fisher corpora Switch Board SRE	13916 1991 2740	MFCC, PNCC	DNN auto encoder	NA
Chung et al. (2017) [15]	Voxceleb	1251	Espectograma	CNN	AC: 80.5 EER: 7.8
Torfi et al. (2018) [12]	WVU-Multimodal 2013	1083	Marco de la MFEC	3D-CNN	EER: 21.1
Dhakal et al. (2019) [13]	ELSDSR	22	Estadística, característica de Gabor y basada en CNN	SVM RF DNN	AC: 98.07 AC: 99.41 AC: 98.14
Jahangir et al. (2020) [14]	LibriSpeech	50 Hombres 50 Mujeres	Expresiones de los hablantes	MLDNN	AC: 92.9
Este trabajo (2022)	propio	128	Espectograma	CNN	AC: 93

Además, existe una gran diferencia entre las bases de datos utilizadas por los diferentes autores, ya que contienen diferencias acerca del ruido ambiental en los audios, el número de audios, algunos cuentan con grandes variedades de etnias, edades, nacionalidades y en contraste otros solo consideran el idioma Inglés-Americano. Por lo tanto, es difícil determinar cuál sistema de reconocimiento de voz es mejor. Existe variabilidad en las consideraciones de cada trabajo, por lo que tener un sistema único capaz de reconocer a una persona por la voz sigue siendo un reto abierto.

5. Conclusiones

El reconocimiento de voz es complejo debido a que existen diversos factores como el ruido ambiental que puede introducir frecuencias no deseadas al audio y confundir al sistema. Por otro lado, los factores de las emociones, el timbre de voz, también son importantes a considerar este tipo de problemáticas, ya que a pesar de que el

timbre es una de las características que hacen única a la voz, gracias a que con esta se mide la calidad con la que es producida, englobando la parte de la entonación donde mayormente ha tenido investigación el reconocimiento de voz por estar en el dominio de las frecuencias. La forma de como articulas las palabras y la intensidad, son clave al momento de utilizar este tipo de reconocimiento, debido a que al realizar pruebas, el sistema reconocía más fácilmente a las personas que hablaban de manera más clara y a una velocidad modulada.

A nivel hardware se trató de tener un sistema controlado, ya que la sensibilidad con la que se maneja el micrófono es importante, debido a que una alta sensibilidad puede afectar con mucho ruido al momento de hablar y los ruidos ambientales. Además, la distancia con la que hablas al micrófono va de la mano con la sensibilidad, ya que es más fácil para el reconocimiento de voz mantener una distancia corta al micrófono debido a que detecta mejor las frecuencias principales de tu voz.

La voz es un tema amplio de investigar, por los diversos factores que influyen con solo decir una frase, la variabilidad que existe de una persona a otra en es inmensa por las distintas características que puede contar la voz.

5.1. Trabajo futuro

A partir de los resultados obtenidos, y enfocados en el estado del arte, se puede explorar diversos extractores de características que existen, inclusive combinarse para aumentar la fiabilidad del sistema a la hora de identificar mediante la voz y así reducir los falsos positivos que arroja el sistema.

Por otro lado, una vez mejorado el sistema de identificación de voz, se propone agregar una nueva funcionalidad donde se detecten audios falsos, es decir, detectar cuando se reproducen audios de voz en la entrada del micrófono para así robustecer y atacar este tipo de problemas comunes en los sistemas de reconocimiento de voz.

Referencias

1. Banxico. Informe anual sobre el ejercicio de las atribuciones conferidas por la Ley para la Transparencia y Ordenamiento de los Servicios Financieros (2021)
2. Gayathri, M., Malathy, C., Prabhakaran, M.: A review on various biometric techniques, its features, methods, security issues and application areas. In: International Conference On Computational Vision and Bio Inspired Computing, vol. 1108, pp. 931–941 (2019) doi: 10.1007/978-3-030-37218-7_99
3. Boles, A. Rad, P.: Voice biometrics: Deep learning-based voiceprint authentication system. In: 12th System Of Systems Engineering Conference (SoSE). IEEE, pp. 1–6 (2017) doi: 10.1109/SYSOSE.2017.7994971
4. Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., Zhang, D.: Biometrics recognition using deep learning: A survey. ArXiv Preprint ArXiv:1912.00271 (2019) doi: 10.48550/arXiv.1912.00271
5. Jain, A., Nandakumar, K.: Biometric authentication: System security and user privacy. Computer, vol. 45, no. 11, pp. 87–92 (2012)

6. Muckenhirn, H., Doss, M. M., Marcell, S.: Towards directly modeling raw speech signal for speaker verification using CNNS. In: IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 4884–4888 (2018) doi: 10.1109/ICASSP.2018.8462165
7. Tandel, N. H., Prajapati, H. B., Dabhi, V. K.: Voice recognition and voice comparison using machine learning techniques: A Survey. In: 6th International Conference On Advanced Computing And Communication Systems (ICACCS), pp. 459–465 (2020) doi: 10.1109/ICACCS48705.2020.9074184
8. Hanifa, R., Isa, K., Mohamad, S.: A review on speaker recognition: Technology and challenges. Computers & Electrical Engineering, vol. 90, pp. 107005 (2021) doi: 10.1016/j.compeleceng.2021.107005
9. Variani, E., Lei, X., McDermott, E., Moreno, I. L., Gonzalez-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. In: IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 4052–4056 (2014) doi: 10.1109/ICASSP.2014.6854363
10. Plchot, O., Burget, L., Aronowitz, H., Matejka, P.: Audio enhancing with DNN autoencoder for speaker recognition. In: IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 5090–5094 (2016) doi: 10.1109/ICASSP.2016.7472647
11. Lukic, Y., Vogt, C., Dürr, O., Stadelmann, T.: Speaker identification and clustering using convolutional neural networks. In: IEEE 26th International Workshop On Machine Learning For Signal Processing (MLSP), pp. 1–6 (2016) doi: 10.1109/MLSP.2016.7738816
12. Torfi, A., Dawson, J., Nasrabadi, N. M.: Text-independent speaker verification using 3d convolutional neural networks. In: IEEE International Conference On Multimedia And Expo (ICME). IEEE, pp. 1–6 (2018) doi: 10.1109/ICME.2018.8486441
13. Dhakal, P., Damacharla, P., Javaid, A. Y., Devabhaktuni, V.: A near real-time automatic speaker recognition architecture for voice-based user interface. Machine learning and knowledge extraction, vol. 1, no. 1, pp. 504–520 (2019) doi: 10.3390/make1010031
14. Jahangir, R., Teh, Y. W., Memon, N. A., Mujtaba, G., Zareei, M., Ishtiaq, U., Akhtar, M., Ali, I.: Text-independent speaker identification through feature fusion and deep neural network. IEEE Access, vol. 8, pp. 32187–32202 (2020) doi: 10.1109/ACCESS.2020.2973541
15. Nagrani, A., Chung, J. S., Zisserman, A.: Voxceleb: a large-scale speaker identification dataset. ArXiv Preprint ArXiv:1706.08612 (2017) doi: 10.21437/Interspeech.2017-950

ISSN 1870-4069

Predicción de señales financieras usando una red neuronal convolucional de memoria de corto plazo

Jesus-Alejandro Lechuga-Ortega, Volodymyr Ponomaryov, Rogelio Reyes-Reyes

Instituto Politécnico Nacional,
Sección de Estudios de Posgrado e Investigación,
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán,
México

jlechugao1500@alumno.ipn.mx,
{vponomar, rreyesre}@ipn.mx

Resumen. Las señales financieras tienen como objetivo mantener un registro histórico del valor de las acciones en forma de series de tiempo. Para la predicción de valores en una serie de tiempo se han propuesto diferentes enfoques, centrados principalmente en el análisis estadístico clásico, sin embargo, cada vez existen más sistemas que emplean el aprendizaje automático, demostrando resultados optimistas para la predicción de señales financieras con un rango de error admisible. En el presente trabajo se emplea un método de aprendizaje automático basado en una red neuronal convolucional de memoria de corto plazo (ConvLSTM2D), donde a partir de la información obtenida de diferentes índices bursátiles de alta importancia a nivel mundial, permite predecir el valor de cierre para el día siguiente; dando especial énfasis al índice Standard & Poor's 500 (S&P500), debido a su gran longevidad e importancia a nivel mundial. Inicialmente se realiza un preprocesamiento de los datos obtenidos de los índices bursátiles, para interpolar aquellos que falten; posteriormente se realiza una extracción de características a partir de indicadores técnicos creando ventanas de tiempo, que servirán para el entrenamiento y validación del sistema propuesto. El sistema obtiene un MSE promedio de 1.22e-4, demostrando resultados favorables para la predicción de señales financieras.

Palabras clave: Series de tiempo, extracción de características, procesamiento de señales, LSTM, ConvLSTM2D, aprendizaje automático, machine learning.

Prediction of Financial Signals Using a Short-Term Memory Convolutional Neural Network

Abstract. Financial signals aim to keep a historical record of stock value in the form of time series. Different approaches have been proposed for the prediction

of values in a time series, mainly focused on classical statistical analysis, however, there are more and more systems that use machine learning, showing optimistic results for the prediction of financial signals with a range of permissible error. In the present work, an automatic learning method based on a short-term memory convolutional neural network (ConvLSTM2D) is used, where, based on the information obtained from different stock market indices of high importance worldwide, it allows predicting the closing value for the next day; giving special emphasis to the Standard & Poor's 500 (S&P500) index, due to its great longevity and importance worldwide. Initially, a preprocessing of the data obtained from the stock indices is carried out, to interpolate those that are missing; Later, an extraction of characteristics is carried out from technical indicators, creating time windows, which will serve for the training and validation of the proposed system. The system obtains an average MSE of 1.22e-4, showing favorable results for the prediction of financial signals.

Keywords: Time series, feature extraction, signal processing, LSTM, ConvLSTM2D, machine learning.

1. Introducción

Existe un gran registro histórico de los mercados financieros en forma de series de tiempo, que desde hace varios años se busca modelar y predecir; sin embargo, no es una tarea fácil, debido a su alta volatilidad y complejidad, sin olvidar las múltiples variables estocásticas y no estacionarias que comprenden las señales de este tipo [1]. Diariamente, se genera una gran cantidad de datos para las señales financieras, es por esto, que se ha despertado un gran interés en el área por parte de los investigadores para predecir su comportamiento [2]. El aprendizaje automático es una herramienta moderna para la predicción y extracción de características, debido a esto, demuestran una gran eficiencia para la predicción de las señales financieras [3].

Las redes neuronales artificiales (ANN), han demostrado un buen rendimiento para la predicción de señales financieras, debido a su capacidad de extraer características primordiales, además de reconocer patrones de información en espacios de alta dimensión [4-5].

Para predecir las señales financieras el modelo usado son las redes neuronales de memoria de corto plazo (LSTM), las cuales son un tipo de redes neuronales recurrentes [6]. Las LSTM tienen la capacidad de trabajar con intervalos de tiempo para predecir las señales financieras, para este fin, es necesario tomar en consideración no solamente los datos actuales, sino también los datos previos de la señal [7].

Un modelo más que demuestra tener resultados favorables para la predicción de señales financieras, son las redes convolucionales de memoria de corto plazo (ConvLSTM2D), siendo una variación de las redes LSTM [8]. Debido a la alta complejidad de las señales financieras, las redes ConvLSTM2D pueden encontrar patrones espaciotemporales a partir de la información recolectada [9].

En este artículo se presenta un sistema para predecir las señales financieras mediante el uso de ConvLSTM2D, para predecir el precio de cierre del índice Standard & Poor's 500 (S&P500) a partir de la información proporcionada por los índices mundiales más importantes. La principal contribución de este trabajo es la siguiente: una novedosa forma para el entrenamiento a partir de validaciones para las ConvLSTM2D usando ventanas de tiempo, con la intención de obtener una mejor precisión en la predicción de las señales financieras.

2. Trabajos relacionados

Hoseinzade et al. [10], proponen 2 sistemas basados en redes neuronales convolucionales (CNN), estos trabajan de forma bidimensional y tridimensional usando diferentes índices bursátiles americanos: Nasdaq Composite (NSQ), Dow Jones Industrial Average (DJI) y S&P500 para predecir el movimiento de los mercados. Inicialmente, extraen 80 indicadores técnicos de cada índice. Posteriormente el 60% de la información recopilada, es utilizada para el entrenamiento, 20% para validación y 20% para pruebas de la red.

Sin embargo, al utilizar pocas características, no generan suficiente información para el entrenamiento de las CNN, lo que no permite una correcta generalización del problema planteado. Adicionalmente, el uso de las técnicas de reducción de dimensiones en las características extraídas no demuestra un menor error de predicción en el sistema propuesto.

Ye et al. [11], presentan un algoritmo basado en CNN para predecir señales financieras. Para obtener información de los mercados, utilizan los índices estadounidenses: Johnson Outdoors Inc (JOUT), GlaxoSmithKline plc (GSK), DJI y S&P500. Primeramente, para la extracción de características relevantes usan la deformación dinámica del tiempo (DTW) y la divergencia de Jensen-Shannon (JS) con la intención de extraer características similares a las DTW. Seguidamente, efectúan una división de los datos para el entrenamiento y prueba del sistema.

Adicionalmente, aplican la técnica Transfer-Learning (TL), la cual a partir de la información obtenida de un índice se entrena otra red CNN con la intención de predecir el índice objetivo. No obstante, el aplicar la DTW a los índices, es un enfoque inapropiado, debido al alto costo computacional que conlleva el extraer la cantidad adecuada de características para un mismo índice.

Jerez et al [12], implementan varios sistemas basados en redes neuronales perceptron multicapa (MLP), redes neuronales recurrentes (RNN) y CNN para pronosticar el movimiento de diferentes mercados financieros a nivel global.

Utilizan los datos de Apertura, Máximo, Mínimo y Cierre (OHLC) de las señales financieras. Emplean los índices DJI, NSQ y S&P500. En primer lugar, realizan una extracción de características mediante el algoritmo K-means para generar de 2 a 5 clases, dando énfasis a la volatilidad de los mercados.

Seguidamente, seccionan los datos en 80% para entrenamiento y 20% para pruebas, además de generar un subconjunto de validación, conformado por el último 20% de los

datos del conjunto de entrenamiento; es importante mencionar, que el subconjunto de validación ayuda a mitigar el sobreajuste (*overfitting*), o el desajuste (*underfitting*) de las redes.

Pese al inconveniente de seleccionar la cantidad de clases adecuadas como características, es necesario conocer la clase esperada para una predicción adecuada. Lo cual es un proceso complicado en caso de aumentar las características.

Ghosh et al [13], proponen un esquema centrado en el uso de redes neuronales de memoria de corto plazo (LSTM) para la compra y venta de las acciones, usando todas las acciones que comprenden el índice S&P500. Inicialmente dividen los datos en un 25% para extracción de características, 50% para entrenamiento, 25% para validación y prueba del sistema.

Seguidamente, realizan una extracción automática de 3 características utilizando el precio de apertura, cierre y el retorno diario para todas las acciones que comprenden el índice. Posteriormente, seleccionan 93 características empleando bosques aleatorios (RF). Consecutivamente, realizan una clasificación binaria mediante el retorno diario para obtener el movimiento positivo o negativo del índice S&P500. Sucesivamente, crean un nuevo subconjunto a partir de los conjuntos de entrenamiento, validación y prueba obtenidos anteriormente, además de usar como etiqueta la clasificación binaria previamente realizada.

Finalmente, seccionan el nuevo conjunto en un 50% para entrenamiento, 25% para prueba y 25% para validación del esquema. Sin embargo, el esquema de partición de datos causa un sesgo de información a causa de la manipulación realizada con los conjuntos creados y dando como resultado un mal entrenamiento de la LSTM.

Rezaei et al [14], desarrollaron un sistema basado en CNN y LSTM para predecir el precio de cierre de los índices Nikkei 255 (NIK), German Stock 30 (DAX), S&P500, DJI. En primer lugar, realizan una descomposición de modo empírico (EMD) para obtener las componentes significativas.

Luego, obtienen la frecuencia de las señales usando funciones de modo intrínseco (IMF) y efectúan la extracción de características mediante CNN. Posteriormente, se crea un conjunto de ventanas de tiempo a partir de las características. Finalmente, seccionan los datos recopilados donde el 80% los datos son para el entrenamiento y el 20% son para prueba de la LSTM.

Los autores no mencionan cuantas características son necesarias para el entrenamiento de la CNN, otro inconveniente, es que por cada IMF se entrena una LSTM, por lo que se necesita una gran capacidad de cómputo.

Cao et al [15], implementan un algoritmo orientado a predecir el precio de divisas USD/CNY usando redes LSTM. Primeramente, emplean diversas características a nivel de mercado (Mercado de materias primas, mercado de acciones) y macroeconómico (Política monetaria, rango de inversión, inflación, balanza comercial e incertidumbre económica). Posteriormente, se dividen los datos en 2 conjuntos, donde el 70% es para entrenamiento y el 30% es para pruebas. Seguidamente, se entrena una LSTM para cada característica obtenida. Finalmente, se entrena una LSTM con la información obtenida para predecir el movimiento del par USD/CNY.

No obstante, para el entrenamiento de múltiples redes se necesita una gran cantidad de cómputo, otra desventaja, es que el enfocarse solo en un índice, no aporta la información suficiente para predecir otros mercados enfocados a las divisas.

Para hacer frente a los retos anteriores, se propone un sistema para predecir señales financieras basado en redes ConvLSTM2D, usando la información de los índices mundiales más importantes. Su objetivo consta de extraer características a partir de indicadores técnicos para cada índice seleccionado, además de utilizar las señales previamente recopiladas para predecir el precio de cierre del índice S&P500.

3. Sistema propuesto

El sistema consta de cuatro etapas principales: procesamiento de los datos, extracción de características, arquitectura y entrenamiento de la ConvLSTM2D, y postprocesamiento de los datos. La Fig. 1 muestra de forma detallada el diagrama a bloques del sistema propuesto para la predicción de señales financieras.

En primer lugar, se eligen los índices bursátiles más importantes a nivel global: Hang Seng (HSI), Financial Times Stock Exchange (FTSE), Nyse Composite (NYSE), German Stock Index (DAX), Nikkei 255 (NIK), Dow Jones Industries (DJI) y S&P500. Seguidamente, en la etapa de preprocesamiento, se usa la interpolación lineal, con la intención de que las fechas de los índices sean homogéneas; si no es suficiente, se opta por eliminar esos datos.

Posteriormente, en la etapa de procesamiento de datos, se extraen múltiples características, a través de diferentes indicadores técnicos, las cuales son normalizadas; a partir de estas señales, se crean múltiples ventanas de tiempo, que se utilizan para el entrenamiento, validación y prueba del modelo propuesto.

Finalmente, se obtienen las predicciones para cada índice de manera independiente, con la finalidad de obtener información suficiente para predecir el índice S&P500. Cada una de estas etapas se detallarán a continuación.

3.1. Preprocesamiento

Debido a la diferencia horaria de cada uno de los índices, puede existir una discordancia entre las fechas, por lo que se busca agregar información faltante, de tal manera que los índices contengan la misma cantidad de información en la mayoría de las fechas.

Esto se lleva a cabo mediante una interpolación lineal, a partir del intervalo de tiempo empleado para el S&P500 a fin de que los índices trabajen con fechas semejantes durante el lapso estudiado. Finalmente, se considera que en ocasiones los mercados no operan, razón por la que la interpolación no es suficiente y se opta por eliminar esos datos.

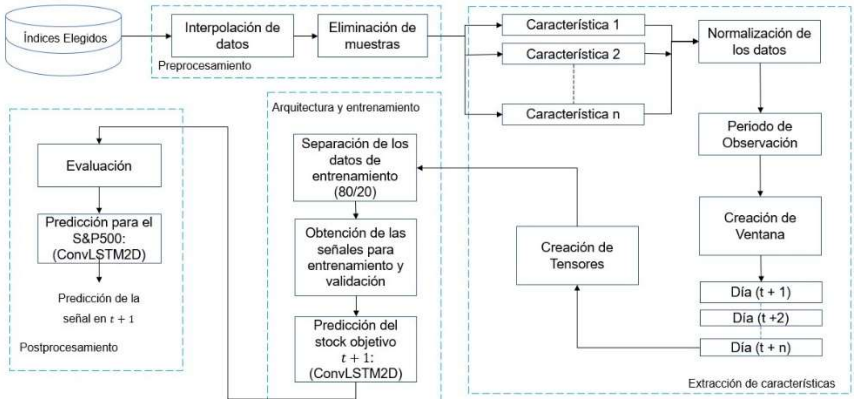


Fig. 1. Diagrama a bloques del sistema propuesto.

3.2. Extracción de características

Inicialmente, se realiza la extracción de características [16], a partir de las señales OHLC, seleccionando un conjunto de indicadores técnicos divididos en 6 subconjuntos: indicadores de tendencia (indican la dirección del movimiento del precio del índice), osciladores de tendencia (miden la desviación media del precio del índice), reconocimiento de patrones (buscan patrones similares en las señales OHLC), transformación de precio (denotan los movimientos abruptos de los índices), indicadores de volatilidad (calculan la rapidez con la que cambia el precio del índice) y estadísticas. Se tiene un total de 31 características extraídas las cuales se muestran en la Tabla 1.

Posteriormente, se realiza una normalización *MinMax* para cada característica extraída, empleando (1):

$$\overline{x_i} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \tag{1}$$

donde x_{min} y x_{max} son los valores mínimos y máximos de cada valor en la serie de tiempo, x_i es el valor actual y $\overline{x_i}$ es el valor normalizado.

Finalmente, se crean ventanas de tiempo de 1, 5, 10 y 30 días, las cuales corresponden a los datos de entrada para el entrenamiento del sistema propuesto usando una red ConvLSTM2D.

3.3. Arquitectura propuesta

Se hace uso de las redes ConvLSTM2D [17], debido a que estas arquitecturas trabajan con datos espacio temporales. La estructura interna de cada capa ConvLSTM se detalla en (2 - 6):

Tabla 1. Características extraídas.

Indicadores de tendencia	**Osciladores de tendencia**		**Reconocimiento de patrones**	**Transformación de precio**	**Indicadores de volatilidad**	**Estadísticas**	
SMA	STCK	STCD	TWO	AVG	ATR	AVG	BETA
EMA	RSI	WR	TREOUT	MED	NATR	LNR	LNRA
MOM	ADO	CCI	TREIN	TYP	TRANGE	LNRS	STD
MACD	RET		TRE DRG	WCL		TIME	VAR

SMA: Simple Moving Average, EMA: Exponential Moving Average, MOM: Momentum, MACD: Moving Average Convergence/Divergence, STCK: Stochastic Oscillator %K, STCD: Stochastic Oscillator %D, RSI: Relative Strength Index, WR: Williams' %R, ADO: Chaikin A/D Oscillator, CCI: Commodity Channel Index, RET: Return, TWO: Two Crows, TREOUT: Three Outside Up/Down, TREIN: Three Inside Up/Down, TRE: Three Black Crows, DRG: Dragonfly Doji, AVG: Average Price, MED: Median Price, TYP: Typical Price, WCL: Weighted Close Price, ATR: Average True Range, Normalized Average True Range, TRANGE: True Range, BETA: Beta, PRS: Pearson's Correlation Coefficient, LNR: Linear Regression, LNRA: Linear Regression Angle, LNRS: Linear Regression Slope, STD: Standard Deviation, TIME: Time Series Forecast, Var: Variance.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * X_{t-1} + W_{ci} \circ C_{t-1} + b_i), \quad (2)$$

$$f_t = \sigma\big(W_{xf} * X_t + W_{hf} * X_{t-1} + W_{cf} \circ C_{t-1} + b_f\big), \quad (3)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * X_{t-1} + W_{co} \circ C_{t-1} + b_o), \quad (4)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (5)$$

$$H_t = o_t \circ tanh(C_t), \quad (6)$$

donde: $X_1, \dots, X_n$ son las entradas de la red, $C_1, \dots, C_n$ son las salidas de cada celda, $H_1, \dots, H_n$ son los estados, H_n, i_t, f_t, o_t son las compuertas de entrada, olvido y salida respectivamente en un tiempo determinado *t*, cada capa cuenta con su respectiva matriz de pesos W y bias b.

La arquitectura propuesta se encuentra conformada por: una capa ConvLSTM2D, una función de activación *ReLU,* una capa dense: 1, un optimizador: descenso de gradiente estocástico (*SGD)*, una función de pérdida: valor medio cuadrático (*MSE)*, número de épocas: 300, parada sin cambios: 100 épocas, se usa el gradiente acelerado de *Nesterov,* estos parámetros fueron seleccionados de forma experimental, tomando en consideración [10, 16]. La Tabla 2 muestra los hiperparámetros de la red propuesta.

Tabla 2. Hiperparámetros para ConvLSTM2D.

Capas	Kernel	Recurrent Dropout	Taza de aprendizaje	Momento	Exponential Decay
1	3x3, 5x5	0.1	0.1, 0.001, 0.0001	0.8	0.1

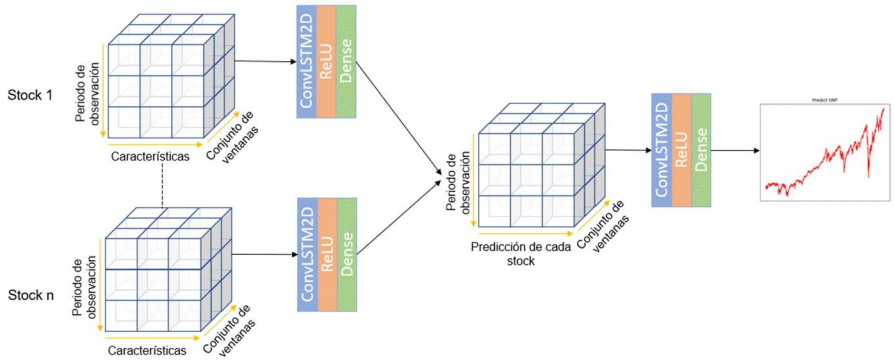


Fig. 2. Arquitectura ConvLSTM2D propuesta Fig. 2. Arquitectura ConvLSTM2D propuesta.

Seguidamente, para el entrenamiento, validación y prueba de las redes propuestas, se usan 5 años de los índices seleccionados, donde el periodo de tiempo analizado va del 01/01/2015 – 31/12/2020, obteniendo 1510 días. Los datos se dividen en 2 conjuntos donde el 80% de los datos son para el conjunto de entrenamiento y el 20% restante es para el conjunto de pruebas, adicionalmente, se crea un subconjunto de validación, el cual consta de un 20% de los datos de entrenamiento para el esquema de validación secuencial [13].

Por último, se entrena una ConvLSTM2D diferente para cada índice seleccionado, con la misma arquitectura, los datos de entrada están conformados por tensores que incluyen la siguiente información: periodo de observación, características seleccionadas y las ventanas de tiempo, como se muestra en la Fig. 2.

3.4. Postprocesamiento

Finalmente, se entrena una red ConvLSTM2D con la misma arquitectura mencionada en la sección 3.3, donde los datos de entrada serán las señales obtenidas por cada una de las redes entrenadas anteriormente, las cuales aportan la información necesaria para predecir el valor de cierre del índice S&P500; en la Fig. 3 se muestran las señales obtenidas de cada modelo entrenado.

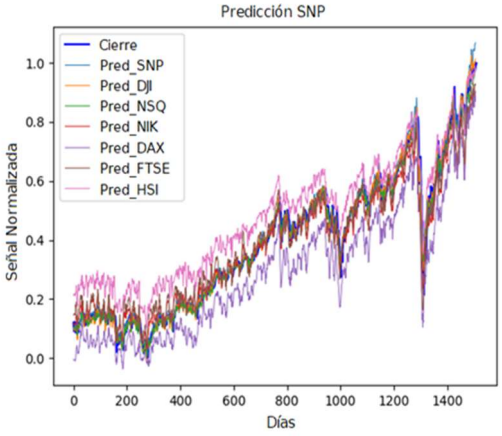


Fig. 3. Predicciones obtenidas para el S&P500 con los datos de los índices seleccionados.

Tabla 3. Resultados obtenidos para los modelos propuestos.

Algoritmo	**Características**	**MSE**	**RMSE**	**MAPE**	**Tiempo de procesamiento (Hrs)**
Ventana: 30 días					
LSTM	30	5.83e-3	5.42e-2	2.19e-1	18.54
Bidireccional LSTM	30	1.4e-3	3.73e-2	3.79e-1	16.77
ConvLSTM1D	30	2.3e-3	4.8e-2	6.96e-1	6.14
ConvLSTM2D	30	1.9e-3	4.38e-2	1.75e-1	5.92
Ventana: 10 días					
LSTM	30	7.15e-3	2.67e-2	2.50e-1	9.45
Bidireccional LSTM	30	7.79e-4	2.79e-2	2.15e-1	8.13
ConvLSTM1D	30	9.19e-4	3.03e-2	4.83e-1	6.26
ConvLSTM2D	30	9.3e-3	9.64e-2	1.75e-1	6.10
Ventana: 5 días					
LSTM	30	4.96e-4	2.23e-2	1.46e-1	6.52
Bidireccional LSTM	30	4.4e-4	2.1e-2	1.85e-1	4.28
ConvLSTM1D	30	6.40e-4	2.53e-2	3.05e-1	2.31
ConvLSTM2D	30	**1.22e-4**	**1.11e-2**	**1.24e-1**	1.71
Ventana: 1 día					
LSTM	30	1.95e-4	1.41e-2	2.15e-1	2.44
Bidireccional LSTM	30	1.50e-4	1.23e-2	2.61e-1	1.31
ConvLSTM1D	30	1.71e-4	1.31e-2	3.01e-1	0.68
ConvLSTM2D	30	2.45e-4	2.05e-2	1.46e-1	**0.55**

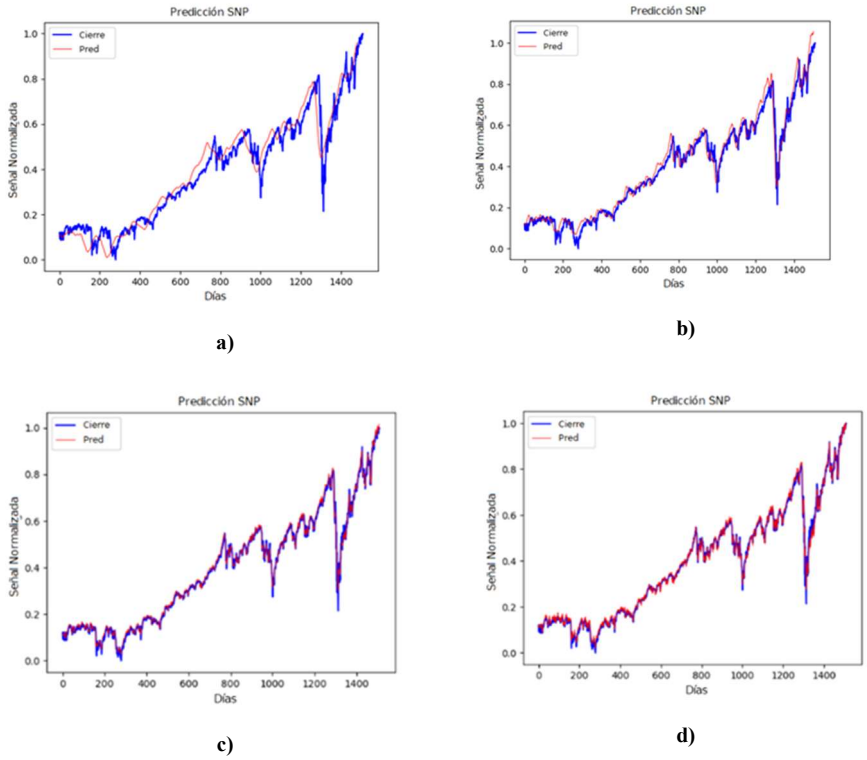


Fig. 4. Predicciones obtenidas por el modelo ConvLSTM2D: a) Ventana 30 días, b) Ventana 10 días, c) Ventana 5 días, d) Ventana 1 día.

Tabla 5. Comparación con [11].

Algoritmo	RMSE
Tr-GRU	2.31e-2
D-LSTM	1.08e-1
D-CNN	1.27e-2
Propuesta (ConvLSTM2D)	**1.20e-2**

4. Resultados experimentales

Se realizaron experimentos con los modelos LSTM, bidirectional-LSTM, ConvLSTM1D, los cuales usan hiperparámetros por defecto de la librería Tensorflow 2, así como el sistema propuesto basado en una red ConvLSTM2D. Se emplearon las 31 características mostradas en la Tabla 1 y se utilizaron las métricas *MSE, RMSE, MAPE (7-9)* así como el tiempo de ejecución para la evaluación y comparación de los

Tabla 6. Comparación con [12].

Algoritmo	MSE
MLP	1.64e-4
LSTM	1.6e-4
Conv	1.59e-4
Propuesta (ConvLSTM2D)	**1.45e-4**

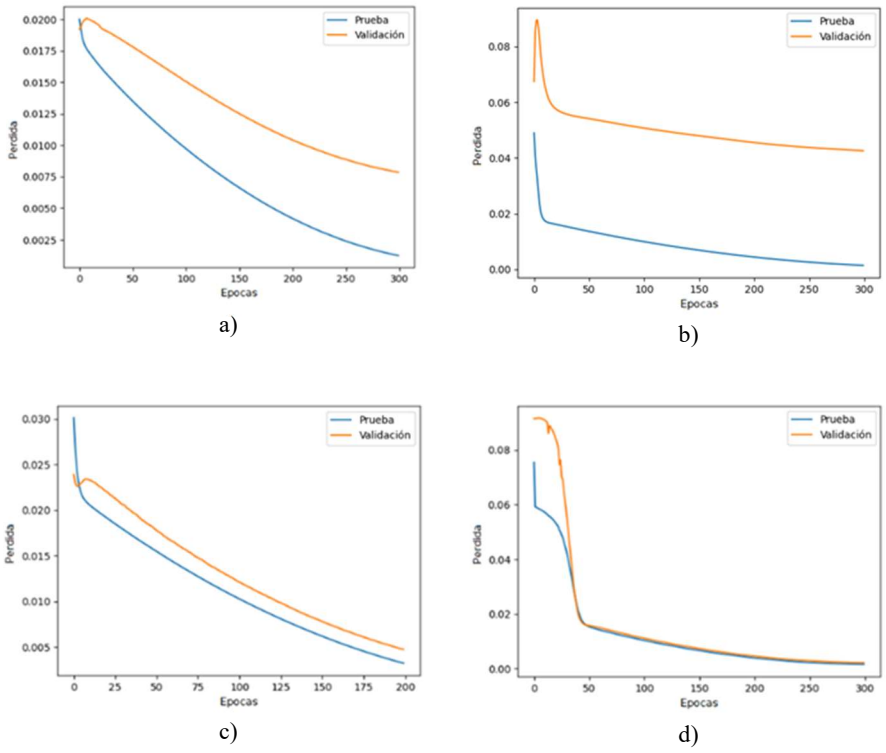


Fig. 5. Curvas de validación y prueba obtenidas para el modelo ConvLSTM2D: a) Ventana 30 días, b) Ventana 10 días, c) Ventana 5 días, d) Ventana 1 día.

diferentes sistemas. Se empleo hardware específico con las siguientes especificaciones: CPU Ryzen 5 5600X 3.7 GHZ, GPU Nvidia GeForce RTX 3060 y 16 Gb RAM:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{7}$$

$$RMSE = \left|\frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2, \right. \tag{8}$$

$$MAPE = \frac{100}{n}\sum^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{9}$$

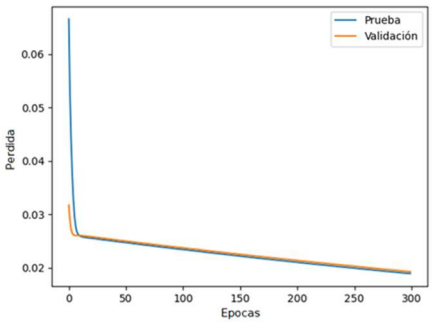


Fig. 6. Curvas de validación y prueba obtenidas del sistema propuesto, para el mismo intervalo de tiempo empleado en [11].

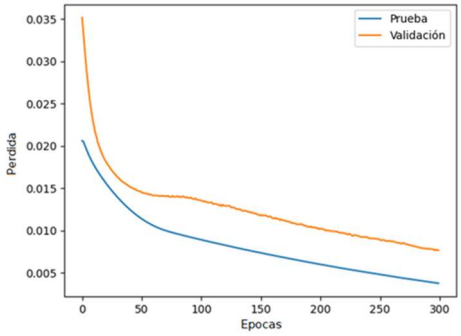


Fig. 7. Curvas de validación y prueba obtenidas del sistema propuesto, para el mismo intervalo de tiempo empleado en [12].

donde: y_i es el valor actual, $\hat{y}_i$ es la predicción y n es el número total de muestras.

Los resultados obtenidos se muestran en la Tabla 3, observándose que el sistema propuesto basado en la red ConvLSTM2D tiene mejores resultados de acuerdo con las métricas elegidas. Se observa que las predicciones obtenidas con ventanas de 10 y 30 días presentan un retraso, mientras que para las ventanas de 1 y 5 días la predicción se encuentra más cercana al precio real de cierre, en especial la ventana de 5 días tiene el error más bajo conforme al MSE (1.22e-4), véase la Fig. 4.

Por otra parte, la curva de validación es similar a la de pérdida, donde ambas decrecen suavemente de manera simultánea y proporcional, evitando el *underfitting* y el *overfitting* [18] como se muestra en la Fig. 5.

A continuación, se muestra la comparativa del sistema propuesto con otros autores [11, 12]. En [11], se utilizan 10 años de datos, los cuales van de 16/11/2009-14/11/2019.

El criterio empleado para esta comparación es el *RMSE*, y se tomaron como referencia los 3 modelos propuestos con los mejores resultados obtenidos por los

autores, los resultados comparativos se muestran en la Tabla 5. De acuerdo con los resultados experimentales, el sistema tiene un menor error en la predicción respecto a la métrica RMSE a comparación de [11], esto se debe a la extracción de características primordiales al usar una red ConvLSTM2D. En la Fig. 6 se puede observar que las curvas de validación y perdida decaen de forma constante a partir de la época 20, por lo cual, el sistema propuesto no presenta *overfitting* ni *underfitting* en comparación con [11] durante el mismo intervalo de tiempo analizado en ambas propuestas.

En relación con [12], el intervalo de tiempo empleado es de 48 años, el cual comprende de 02/01/1970-01/06/2018. El criterio considerado para esta comparación es el *MSE*, y se ocuparon como referencia los 3 algoritmos propuestos con los mejores resultados obtenidos por los autores, los resultados comparativos se muestran en la Tabla 6.

De acuerdo con los resultados experimentales, el sistema tiene un menor error respecto a la métrica MSE a comparación de [12], ya que al utilizar las señales obtenidas de diferentes índices como entrada de datos para la red ConvLSTM2D, se logra un buen entrenamiento para la misma. En la Fig. 7 se puede apreciar que las curvas de validación y perdida decrecen suavemente a partir de la época 50, debido a esto, el sistema propuesto no presenta *overfitting* ni *underfitting* en comparación con [12] durante el lapso analizado en ambas propuestas.

De acuerdo con los resultados experimentales obtenidos a partir de diferentes intervalos de tiempo, se demuestra que el sistema propuesto tiene mejor desempeño (menor error) en la predicción del precio de cierre para el índice S&P500 debido al uso de diferentes índices técnicos; ya que estos aportan información relevante para un correcto entrenamiento de las redes ConvLSTM2D independientes.

5. Conclusiones y trabajo a futuro

En el presente trabajo se diseñó un sistema para predecir señales financieras basado en redes ConvLSTM2D, en el cual se utiliza la información de los índices más representativos a nivel mundial, entre los que se encuentran: NIK, HSI, DAX, FTSE, NYSE, DJI y S&P500, de los cuales se extraen 31 características relevantes que consisten en: indicadores de tendencia, osciladores de tendencia, reconocimiento de patrones, transformación de precio, indicadores de volatilidad y estadísticas. Posteriormente, se crean ventanas de tiempo de 1, 5, 10 y 30 días para el entrenamiento y validación del sistema propuesto.

Uno de los puntos clave del sistema es el uso de validaciones secuenciales, donde el sistema busca patrones en un cierto intervalo de tiempo sin mostrar todo el periodo estudiado. Ayudando a la red a encontrar patrones relevantes para una predicción eficaz del precio de cierre del índice S&P500.

Los resultados obtenidos para las métricas MSE, RMSE y MAPE fueron 1.22e-4, 1.11e-2 y 1.24e-1 respectivamente, usando una ventana de tiempo de 5 días, demostrando así un desempeño superior a los métodos propuestos en el estado del arte.

El diseño de este tipo de sistemas puede ayudar a los financieros a tomar decisiones efectivas al momento de comprar o vender un activo. Sin embargo, se debe tener en

consideración el riesgo asociado al mercado de acciones, debido a múltiples factores externos que influyen directamente a los mercados. Como trabajo futuro se pretende implementar módulos de atención para mejorar el performance del sistema propuesto, además de crear portafolios de inversión con la intención de tener un retorno aceptable con el menor riesgo posible.

Referencias

1. Marszałek, A., Burczyński, T.: Modeling and forecasting financial time series with ordered fuzzy candlesticks. Information sciences, vol. 273, pp. 144–155 (2014)
2. Lau, K. W., Wu, Q. H.: Local prediction of non-linear time series using support vector regression. Pattern recognition, vol. 41, pp. 1539–1547 (2008)
3. Arévalo, A., Niño, J., Hernández, G., Sandoval, J.: High-frequency trading strategy based on deep neural networks. Intelligent computing methodologies, vol. 9773, pp. 424–436 (2016)
4. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, vol. 65 (1958)
5. Moghaddam, A. H., Moghaddam, M. H., Esfandyari, M.: Stock market index prediction using artificial neural network. Journal of Economics, Finance and Administrative Science, vol. 21, pp. 89–93 (2016)
6. Hochreiter, S., Jürgen S.: Long short-term memory. Neural computation, vol. 9, pp. 1735–1780 (1997)
7. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on CEEMDAN and LSTM. Physica A: Statistical Mechanics and its Applications, vol. 519, pp. 127–139 (2019)
8. Shi, X., Chen, Z., Wang, H., Yeung, D. Y.: Wong, W. K., Woo, W. C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, vol. 28 (2015)
9. Gunduz, H., Yaslan, Y., Cataltepe, Z.: Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. Knowledge-Based Systems, vol. 137, pp. 138–148 (2017)
10. Hoseinzade, E., Haratizadeh, S.: CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications, vol. 129, pp. 273–285 (2019)
11. Ye, R., Dai, Q.: Implementing transfer learning across different datasets for time series forecasting. Pattern Recognition, vol. 109 (2021)
12. Jerez T., Kristjanpoller W.: Effects of the validation set on stock returns forecasting. Expert Systems with Applications, vol. 150, pp 113271 (2020)
13. Ghosh, P., Nufeld, A., Keshari Sahoo, J.: Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. Finance Research Letters, pp 102280 (2021)
14. Rezaei, H., Faaljou, H., Mansourfar, G.: Stock price prediction using deep learning and frequency decomposition in Expert Systems with Applications, vol. 169 (2021)
15. Cao, W., Zhu, W., Wang, W., Demazeau, Y., Zhang, C.: A deep coupled LSTM approach for USD/CNY exchange rate forecasting. IEEE Intelligent Systems, vol. 35, pp. 43–53 (2020)

16. Henrique, B. M., Sobreiro, V. A., Kimura, H.: Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications, vol. 124, pp. 226–251 (2019)
17. Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., Woo, W. C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, vol. 28 (2015)
18. Li, F., Wu, J., Gao, R.: Neural Networks for Visual Recognition in class notes for CS231. (2021)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación