EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 152 No. 2
February 2023

Research in Computing Science

Research in Computing Science, Año 22, Volumen 152, No. 2, febrero de 2023, 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 febrero de 2023.

Research in Computing Science, year 22, Volume 152, No. 2, February 2023, is published monthly by the Center for Computing Research of IPN.

Volume 152(2)

Advances in Computing Science and Applications

Tania Alcántara
Hiram Calvo-Castro (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

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

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Sistema geográfico para la representación de opiniones asociadas a la vacunación COVID-19

César Eduardo Monroy Pérez, Belém Priego-Sánchez, Gabriela A. García-Robledo, José A. Reyes-Ortiz

Universidad Autónoma Metropolitana Unidad Azcapotzalco, Departamento de Sistemas, México

cesarmonroy77@gmail.com, {abps, gagr, jaro}@azc.uam.mx

Resumen. En México la distribución de vacunas COVID-19 se hizo de acuerdo al lugar de residencia y edad de cada habitante. En ocasiones una persona puede tener más de una opción de módulos de vacunación debido a la magnitud de personas pertenecientes a la misma localidad, esta información puede ser de utilidad junto con la opinión de las personas que ya acudieron previamente para desarrollar una idea del nivel de atención del personal, el tiempo estimado de aplicación o la opinión de los usuarios de acuerdo a cada módulo de vacunación. Este trabajo recupera información de Twitter para analizarla y mostrar la opinión general de las personas acerca de cada módulo de vacunación. El resultado es representado en un mapa interactivo de México, donde se muestra la ubicación de cada módulo, el tipo de opiniones que tiene (positivas, negativas o neutras) y el nombre oficial del sitio de aplicación. Las principales aportaciones de este artículo son: un mapa interactivo de México, información de los distintos módulos de vacunación, un módulo de recuperación de opiniones de la red social Twitter, un analizador de opiniones automático y la representación de información entendible para el usuario final.

Palabras clave: Análisis de sentimientos, representación de información, recuperación de información.

Geographic Representation System of Opinions Associated with Vaccination COVID-19

Abstract. In Mexico, the distribution of vaccines COVID-19 was made according to the place of residence and age of each inhabitant. Sometimes a person may have more than one choice of vaccination modules due to the magnitude of people belonging to the same locality, This information can be useful, together with the opinion of people who have already visited previously, to develop an idea of the staff's level of attention, the

estimated time of application or the opinion of the users according to each vaccination module. This work retrieves information from Twitter to analyze it and show the general opinion of people about each vaccination module. The result is represented in an interactive map of Mexico, where the location of each module is shown, the type of opinions it has (positive, negative or neutral) and the official name of the application site. The main contributions of this article are: an interactive map of Mexico, information on the different vaccination modules, a module for retrieving opinions from the social network Twitter, an automatic opinion analyzer and the representation of understandable information for the end user.

Keywords: Sentiment analysis, information representation, information retrieval.

1. Introducción

A partir de que surgieron las vacunas para combatir el virus COVID-19, en México se decidió comenzar con múltiples módulos de vacunación [1] para que cada habitante decida donde realizarse la aplicación de su vacuna a partir de su lugar residencia, primer apellido y edad; por ejemplo, si una persona vive en la alcaldía Gustavo A. Madero de la CDMX podría acudir el día que se le asigne a dos distintas sedes: "Centro cultural Jaime Torres Bodet" y "Escuela Nacional Preparatoria #9", debido a esta múltiple opción el usuario necesita conocer el tipo de atención del personal a través de las opiniones de las personas, con el objetivo de conocer el tiempo de espera, atención al ciudadano o mejor organización, sin dejar de lado la necesidad de conocer la dirección exacta del módulo al que se desea acudir, para que su experiencia sea lo mejor posible. Por lo tanto, se realizó un sistema que representa la calidad de servicio de cada módulo de vacunación y dirección utilizando las opiniones recuperadas de la red social Twitter.

Twitter [2] es una red social que permite recuperar los comentarios de los usuarios a través de distintos métodos. Para este trabajo la biblioteca de python snscrape [3] funciona como metodo de recuperación, obteniendo información que permite realizar un análisis de polaridad con ayuda de la biblioteca de python `sentiment_analysis_spanish` [4], que utiliza la clasificación Naïve Bayes [5] para predecir el sentimiento de las oraciones en español, clasificando las opiniones de los usuarios en tres categorías: positiva, negativa o neutra, contando con un presición de validacion del 90 %.

Naïve Bayes es un algoritmo simple y poderoso para la clasificación, con una suposición de independencia entre los predictores. Naïve Bayes es fácil de construir y útil para conjuntos de datos muy grandes [5].

La connotación social proporciona una perspectiva de la calidad de servicio que se está brindando en cada módulo de vacunación. Por ello, el objetivo principal de este trabajo es realizar un sistema capaz de representar en un mapa

interactivo la calidad del servicio de los módulos de vacunación en México, a partir de un análisis de polaridad en opiniones de usuarios, para servir de auxiliar en la decisión de una sede para la aplicación de la vacuna contra el COVID-19, Además, obtener una perspectiva de la calidad de servicio percibida en los centros de vacunación dentro del territorio de México, debido a que un análisis de esta información aun no ha sido publicado. utilizando áreas de investigación, tales como, recuperación y extracción de información, análisis de sentimientos, procesamiento de lenguaje natural y representación de información.

El resto del artículo se organiza de la siguiente manera. En la sección 2 se presentan los avances más importantes y recientes respecto a las áreas de investigación de este trabajo, tales como: a) análisis de sentimientos, b) representación de información en mapas interactivos y c) extracción y recuperación de información a partir de datos de una red social. Por su parte en la Sección 3 se expone metodología propuesta, en la Sección 4 se muestra la evaluación de este mediante una experimentación del método de recuperación de información general y del contenido de los tweets recuperados, la sección 5 documenta el análisis y resultados obtenidos. Finalmente, en la sección 6 se presentan las conclusiones de este artículo y los trabajos a futuro.

2. Estado del arte

En esta sección se presenta una revisión de los avances dentro del área de investigación de este trabajo. En [6] se implementa una herramienta que clasifica automáticamente la información contenida en Twitter haciendo uso del procesamiento del lenguaje natural, con redes neuronales artificiales, identificación de patrones y clasificación según la polaridad de las emociones (positivo, negativo y neutro). En el caso de [7] se desarrolla una herramienta que permite analizar y clasificar textos encontrados en los comentarios de Twitter, generados por los usuarios de un determinado producto. Por otro lado, en [8] se monitorea en tiempo real la actividad en Twitter siguiendo los tweets en inglés, con las palabras “corona”, “covid”, “covid-19”, “coronavirus” y las variantes de “sars-cov-2”, para proveer un conjunto de datos en CSV con los ID de los tweets, además de un sitio que visualiza el análisis de los sentimientos del feed de Twitter.

En [9] se describe y evalúa una aplicación que realiza un análisis supervisado de sentimientos a través de un clasificador en tiempo real de opiniones políticas de tweets, su metodología permite la realización de análisis longitudinales, para detectar cambios en las tendencias asociadas a los partidos políticos y sus candidatos, así como comparar los cambios con los acontecimientos cotidianos. Utilizar modelos predictivos y la aplicación de un clasificador permite conectarse al flujo de datos de Twitter en tiempo real, para predecir y visualizar el sentimiento de cada tweet, y de conjuntos agregados de mensajes. También, en [10] se brinda una alternativa móvil para la visualización de noticias del Ecuador, se presenta un sencillo análisis de sentimientos del contenido de la noticia. En [11] se construye una herramienta que utiliza extracción y ciencia,

para analizar información general sobre los tweets y usuarios, así como los sentimientos especialmente enfocados al COVID-19 de forma que se puedan establecer tendencias e incluso posteriores análisis más complejos.

En el trabajo de [12] se plantea ejecutar un análisis de sentimientos en tweets escritos en Español Mexicano utilizando técnicas de aprendizaje profundo y neuroevolución. La propuesta incluye un algoritmo genético de diseño propio, el cual usa la similitud de las redes neuronales convolucionales para ejecutar el operador de cruce. Asimismo, este algoritmo genético también busca los mejores pesos de los filtros convolucionales y de la capa clasificadora. Los tweets son representados a través de dos modelos: Word2Vec y BERT.

Finalmente, en [13] se utiliza la plataforma Intuiface para un desarrolló de un mapa interactivo y multitáctil, con el objetivo de llevar un inventario de los árboles e informar sobre la supervivencia de estos, proporcionando información enriquecida sobre su ubicación y su clasificación, identificando las especies protegidas y eliminando la necesidad de generar un registro en papel. Se propone un inventario de los árboles en la Sede, además del uso de tecnología multitáctil con la finalidad de hacerlo atractivo para la comunidad estudiantil, proporcionando un entorno en el que los elementos de la aplicación se pueden manipular libremente. Los datos necesarios para crear el mapa se recolectaron mediante búsqueda de información en línea, y fotografías de los árboles.

Para el desarrolló de este trabajo el análisis de información se realizó a través de una biblioteca que utiliza la clasificación Naïve Bayes, para predecir el sentimiento de las oraciones en español, con una precisión de validación del 90 %, como se especifica en su página [4] así proporcionando por parte del desarrollador una clasificación de las opiniones de los usuarios en tres categorías: positiva, negativa o neutra.

3. Metodología propuesta

En esta sección se presenta el método propuesto con el que se desarrolló el sistema para realizar una representación de los módulos de vacunación del COVID-19 en un mapa de México, a partir de las opiniones de los usuarios de Twitter. La metodología consta de cinco fases, donde más adelante se muestra la parte del sistema que se generó para cada una ellas y su interaccion dentro del sistema.

3.1. Creación de base de datos

La base de datos desarrollada para este trabajo se muestra en la Figura 1 y fue creada con MariaDB en su versión 10.5, se encarga de almacenar y relacionar la información encontrada de cada unidad de vacunación en Twitter, con su posible connotación sentimental (positiva, negativa o neutra). Desarrollada de esta manera debido a que los objetos con información son: tweets, sentimientos, módulos. por lo que cada uno de ellos es una tabla, relacionándose por su información en común y cuya finalidad se observa en la Tabla 1.

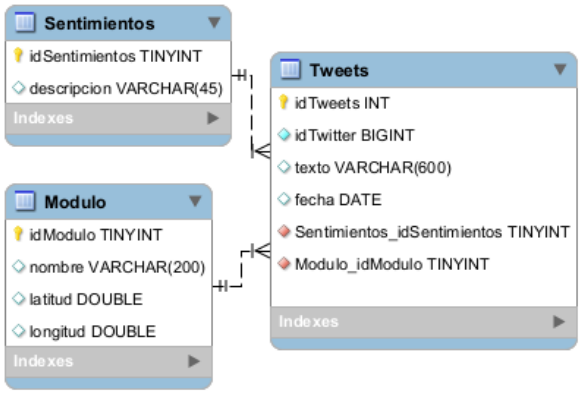

Fig. 1. Modelo relacional de la base de datos.

Tabla 1. Tablas que componen la base datos.

Nombre de la tabla en la base de datos	Descripción
Sentimientos	Catálogo con posibles valores sentimentales
Módulo	Catálogo con módulos registrados
Tweets	Datos recuperados de Twitter

Tabla 2. Valores de sentimientos

Sentimiento	Descripción
Positivo	El usuario se expresó denotando agrado
Neutro	No se expresaron palabras relacionadas al agrado o el desagrado
Negativo	El usuario expreso cierto desagrado

La tabla Sentimientos es un catálogo con tres posibles valores sentimentales que puede tener un tweet, los cuales se muestran en la Tabla 2. La información para la tabla Módulo fue obtenida de manera manual, buscando información en sitios de internet proporcionados por el estado correspondiente o portales de noticias y posteriormente obtener los valores de latitud y longitud de google maps [16], teniendo la cantidad de módulos por estado mostrada en la Tabla 3, siendo un total de 457.

3.2. Recuperación de Tweets

En esta sección se realiza la recuperación de información de la red social Twitter relacionada a los módulos de vacunación en México. Dicho proceso busca rescatar opiniones de usuarios que acudieron a un centro de vacunación, después generar una evaluación aproximada a la calidad brindada.

Para la recuperación se utilizó la biblioteca de python3 snscrape [3] en su versión 0.4.3, que realiza una búsqueda de información que contenga palabras

Tabla 3. Número de módulos por Estado de la Republica Méxicana.

Estado	Número de Módulos	Estado	Número de Módulos
Ciudad de México	113	Tlaxcala	5
Aguascalientes	8	Veracruz	16
Baja California	8	Yucatán	9
Baja California Sur	16	Guanajuato	17
Campeche	13	Guerrero	9
Chiapas	6	Jalisco	26
Chihuahua	4	Quintana Roo	9
Colima	1	San Luis Potosí	48
Durango	4	Sinaloa	11
Hidalgo	10	Sonora	5
Michoacán	9	Morelos	12
Nuevo León	9	Nayarit	6
Puebla	14	Oaxaca	8
Estado de México	20	Querétaro	15
Tabasco	6	Zacatecas	6
Tamaulipas	9	Coahuila	5

clave por cada unidad de vacunación, recuperando aquellos tweets que fueron publicados después de la fecha en la que dio inicio el proceso de vacunación, hasta la fecha en que se ejecuta el programa. Es decir, del 15 de febrero del 2020 hasta el 15 de junio del 2022. Se manejó un radio de 200 metros alrededor de la geolocalización de cada módulo, especificada en la base de información de las unidades, este espacio es considerado el suficiente para cubrir la mayoría de las instalaciones donde se realizó el proceso de vacunación.

Adicionalmente, se rastreó a los que tuvieran al menos una de las siguientes palabras: Vacuna, Vacunándome, Dosis, Covid, AstraZeneca, Pfizer, Sputnik. La composición de mayúsculas o minúsculas en las palabras no influyen en la búsqueda y estas fueron elegidas, ya que se consideran con una mayor probabilidad de aparecer en alguna publicación relacionada con una unidad de vacunación. Entre las palabras clave solo se buscaron tweets para tres vacunas: AstraZeneca, Pfizer y Sputnik, debido a que son las más conocidas y las principales suministradas en México, éstas son buscadas para cada módulo de vacunación y la cantidad de tweets buscadas por cada una es especificada al momento de la ejecución. Los retuits no entran como la información recuperada, recuperando solo el texto de cada tweet.

Es importante mencionar que la base de datos junto con el método de almacenamiento no permite duplicidad de información, en caso de ser así no se almacena.

3.3. Pre-procesado de Tweets

Como primer paso se realiza una depuración de información, haciendo uso de expresiones regulares dentro de Python; para limpiar los datos irrelevantes que

Tabla 4. Categoría por número asignado en el módulo de análisis.

Categoría asociada	Número obtenido del análisis
Positivo	Mayor que 0.67
Neutro	Entre 0.67 y 0.33
Negativo	Menor que 0.33

contiene un tweet. Es decir, aquellas partes donde se involucran menciones o los enlaces a páginas externas, debido a que no brindan algún tipo de contexto. Se eliminan los símbolos numerales (#), las direcciones de páginas web, el texto que se encuentra entre paréntesis o precedido por una mención (símbolo arroba- @).

Un ejemplo de tweet recuperado de twitter es: "Los viejitos contentos de haberse vacunado. Buena organización y ambiente. #VacunasCOVID19 #Iztacalco cx: @Claudiashein https://t.co/nP5j2VPIUa". El texto después del pre-procesado es: "Los viejitos contentos de haberse vacunado Buena organización y ambiente VacunasCOVID19 Iztacalco cx"

3.4. Análisis de polaridad

El tweet pre-procesado es la entrada del módulo de análisis, el cual trabaja con la biblioteca `sentiment_analysis_spanish` en su versión 0.0.25, encargada de devolver un número entre 0 y 1. Los que se encuentran cerca del cero se asocian a sentimientos negativos, mientras que los cercanos a uno se consideran positivos. Para el efecto del sistema se crearon tres categorías en base a la representación numérica que puede ser asignada a un tweet, con el fin de mostrar en un lenguaje simple de entender la emoción que describe a la calidad del servicio. La categoría asociada a cada número asignado al tweet en el módulo de análisis se muestra en la Tabla 4.

Por ejemplo, si se tiene el tweet pre-procesado:"Los viejitos contentos de haberse vacunado Buena organización y ambiente VacunasCOVID19 Iztacalco cx". El número obtenido del módulo de análisis de polaridad es 0.877554016340069, que corresponde a un sentimiento positivo.

Los emojis no están registrados como aportación a la connotación sentimental, por lo que estos son ignorados al momento del pre-procesamiento

3.5. Representación de polaridad en mapa interactivo

La interfaz encargada de mostrar la información recolectada al usuario final es una página web, capaz de mediar con la base de datos descrita en la Sección 3.1, teniendo como principal característica una alta compatibilidad al momento de compartir información, capaz de visualizarse en cualquier dispositivo que disponga de un navegador web.

El portal utiliza la biblioteca Leaflet [15] para integrar de un mapa urbanístico, similar al usado por el servicio de google maps [16]. Se eligió este tipo

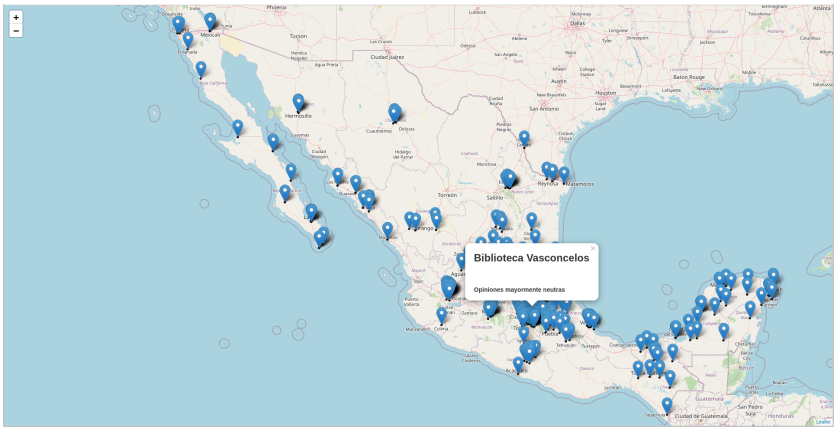

Fig. 2. Mapa web urbanístico con marcadores de los módulos de vacunación en la República Mexicana.

de interfaces ya que facilitan la localización de información por zona geográfica, aunado a esto, los módulos de vacunación muestran marcadores de color azul dentro de una circunferencia, representando el área de búsqueda en el mapa, lo que se puede observar en la Figura 2.

Este módulo consulta la información de la base de datos para obtener el promedio de las opiniones de cada centro de vacunación y enlazarlo a su marcador. Además, de despliega una tarjeta que contiene el nombre del módulo y un mensaje dependiendo del valor promedio referente al sentimiento asociado, mostrando uno de tres posibles mensajes: Opiniones mayormente positivas, Opiniones mayormente neutras y Opiniones mayormente negativas. En caso de no contar con información suficiente su principal se asignará una connotación neutra a la unidad de vacunación.

4. Resultados experimentales

Para la evaluación del sistema propuesto se realizaron tres ejecuciones del sistema encargado de la recuperación, análisis y almacenamiento de tweets, usando los módulos de vacunación pertenecientes a la Ciudad de México. En la Tabla 5 se observa el número de tweets obtenidos en cada una de ellas, además de aquellos almacenados en la base de datos y los que no. Debido a una restricción de repetición, significando que cada tweet solo podía aparecer una vez en el almacén de información.

Se considera duplicidad cuando existe una colisión de los espacios geográficos de búsqueda, la aparición de distintas palabras claves en el mismo tweet o el mismo inicio de búsqueda de información por parte del módulo de recuperación.

Al final de las ejecuciones correspondientes a 113 módulos de vacunación registrados, se obtuvieron con 938 tweets.

Tabla 5. Ejecuciones del módulo de recuperación de opiniones.

Ejecución	Tweets antes de la ejecución	Tweets recuperados	Tweets almacenados	Tweets no guardados
1	0	494	483	11
2	483	772	274	498
3	757	954	181	773

Tabla 6. Cantidad de módulos por tweets solicitados.

Tweets solicitados	Módulos con la cantidad de tweets solicitada	Módulos con menor cantidad de tweets de la solicitada	Módulos sin tweets asociados
20	13	50	50
50	0	14	99
80	0	7	106

La primera ejecución se realizó buscando 20 tweets por cada palabra clave relacionada a un módulo de vacunación para dar inicio a la base de información, se enriqueció con 483 nuevos tweets provenientes de 63 centros de vacunación como se puede ver en la Tabla 5 y 6. Se restringe el almacenamiento a 11 tweets debido a la restricción de duplicidad.

La segunda y tercera ejecución se realizaron con la finalidad de recuperar más información y medirla para comprobar el punto de partida de la búsqueda. Se diagnostica la cantidad de módulos de vacunación que generaban la mayor cantidad de datos.

La segunda ejecución buscó 50 tweets por cada palabra clave, partiendo de 483 tweets almacenados, recuperando 772 y almacenando 274 originarios de 14 módulos de vacunación, como se observa en la Tabla 5 y 6, esto significa que 498 fueron ignorados debido a las coincidencias.

Por último, la tercera ejecución buscó 80 tweets por cada palabra clave, iniciando con 757 tweets en la base de información, recuperando 954, de los cuales solo agregó 181 tweets provenientes de 7 centros de vacunación, ignorando 773 debido a la duplicidad, terminando con 938 tweets.

Posteriormente se realizó una nueva ejecución con la base de información completamente vacia y buscando mil tweets por cada módulo de vacunación registrado, abarcando toda la República Mexicana. Obteniendo la clasificación de opiniones plasmada en la Tabla 7

Al obtener el porcentaje de las opiniones positivas en comparación con la cantidad total se obtiene la Tabla 8, siendo un referente al momento de mostrar al estado con una mejor percepción de calidad de servicios.

5. Análisis y discusión de resultados

Contemplando las evaluaciones realizadas se determina que conforme crece la base de conocimiento la parte de sistema que recupera, analiza y almacena tweets

Tabla 7. Número de opiniones por estado.

Estado	Positivas	Neutras	Negativas	Estado	Positivas	Neutras	Negativas
Ciudad de México	137	446	1434	Tlaxcala	0	0	0
Aguascalientes	1	1	20	Veracruz	12	17	33
Baja California	0	0	8	Yucatán	0	1	1
Baja California Sur	0	0	3	Guanajuato	0	1	6
Campeche	0	3	2	Guerrero	0	2	1
Chiapas	3	1	9	Jalisco	7	39	67
Chihuahua	0	0	1	Quintana Roo	3	4	7
Colima	0	0	0	San Luis Potosí	1	1	5
Durango	0	0	0	Sinaloa	0	0	2
Hidalgo	2	12	13	Sonora	1	0	0
Michoacán	2	4	6	Morelos	7	4	36
Nuevo León	8	1	36	Nayarit	0	0	0
Puebla	0	0	1	Oaxaca	0	0	0
Estado de México	4	3	12	Querétaro	8	33	73
Tabasco	0	0	0	Zacatecas	2	3	9
Tamaulipas	3	3	17	Coahuila	0	0	0

disminuye el reconocimiento de información encontrada como nueva, indicando así que los puntos de inicio de las búsquedas se encontraban cercanos, se abre la posibilidad de que la recuperación empiece a partir de la fecha especificada o inicie desde la fecha de ejecución del módulo de recuperación afectando a los resultados debido a ejecuciones consecutivas.

Por otro lado, se observa que existe un empalme entre algunos módulos de vacunación lo que hace que los datos que pertenecen a un módulo puedan ser filtrados erróneamente a otro.

La principal causa de la duplicidad es por los tweets que tuvieron diferentes palabras clave, de esta manera se obtiene la misma información al buscar diferentes palabras clave. Esto sugiere que la mejora en un método de recuperación es realizar un menor número de ejecuciones con una base mayor para la búsqueda de palabras clave por centro de vacunación.

El sistema muestra que los estados cuentan generalmente con el doble de opiniones negativas que de positivas y neutras como se observa en la Tabla 7. La Ciudad de México cuenta con demasiadas opiniones en comparacion con los demás estados, esto debido a su mayor base de información, y aun así la relación de los sentimientos de la información es muy similar a la de los otros estados.

Tomando la Tabla 8 se llega a la conclusión que el estado con mejor percepcion de calidad del servicio es Sonora; sin embargo, la cantidad base de información de dicho lugar es muy poco como para tener este resultado como cierto. Se intuye que el estado con mayor número de unidades de vacunación con perspectivas positivas en la Ciudad de México debido a su gran banco de datos, tanto de comentarios como de sitios que sumistran vacunas.

Tabla 8. Porcentaje de opiniones positivas por estado.

Estado	% Opiniones positivas	Estado	% Opiniones positivas
Ciudad de México	6.79	Tlaxcala	0
Aguascalientes	4.54	Veracruz	19.35
Baja California	0	Yucatán	0
Baja California Sur	0	Guanajuato	0
Campeche	0	Guerrero	0
Chiapas	23.07	Jalisco	6.19
Chihuahua	0	Quintana Roo	21.42
Colima	0	San Luis Potosí	14.28
Durango	0	Sinaloa	0
Hidalgo	7.4	Sonora	100
Michoacán	16.6	Morelos	14.89
Nuevo León	17.7	Nayarit	0
Puebla	0	Oaxaca	0
Estado de México	21.05	Querétaro	7.01
Tabasco	0	Zacatecas	14.28
Tamaulipas	13.04	Coahuila	0

6. Conclusiones y perspectivas

En este artículo se ha realizado la implementación de un sistema para la consulta de módulos de vacunación contra COVID-19, a partir del razonamiento de opiniones recuperadas de la red social Twitter. Se realizó un análisis de polaridad de sentimientos en cada opinión clasificándose en tres categorías: positiva, negativa o neutra, lo que permitió calcular el promedio de la perspectiva de calidad del servicio de cada módulo de vacunación en México.

Así mismo, se realizó la representación de información en un mapa de la república mexicana, que brinda a los usuarios la oportunidad de comparar la información entre los módulos de vacunación de acuerdo a su ubicación.

Aunque se cuenta con una reducida base de información se logra ver que la relación entre opiniones positivas, negativas y neutras es similar para todos los estados, dejando ver que la opinión general es negativa con respecto a los servicios de vacunación.

Como trabajo a futuro se espera mejorar los resultados del método de recuperación de Tweets, para no encontrar un empalme en los espacios de búsqueda, agregando un método que identifique la relación entre módulos de vacunación, contemplando la reducción de información que esto puede provocar y agregando muchos más módulos de vacunación sobre los cuales buscar información.

Referencias

1. Lidia Arista, Linaloe R. Flores, Vacunación en México arranca el 24 de diciembre con personal de hospitales COVID, Expansión política, 2020 [En linea]. Disponible:

https://politica.expansion.mx/presidencia/ 2020/12/23/vacunacion-en-mexico-arranca-el-24-de-diciembre-con-personal-de-hospitales-covid [Accedido: 24-Apr-2021].
2. Twitter, [En línea]. Disponible: https://www.twitter.com/ [Accedido: 27-Sep-2021]
3. snscrape, [En línea]. Disponible: https://github.com/JustAnotherArchivist/snscrape [Accedido: 4-Feb-2022]
4. sentiment-spanish, [En línea]. Disponible: https://github.com/ sentiment-analysis-spanish/sentiment-spanish [Accedido: 4-Feb-2022].
5. L. Gonzalez, Naïve Bayes – Teoría, [En línea]. Disponible: https://aprendeia.com/naive-bayes-teoria-machine-learning/ [Accedido: 2-Abr-2022].
6. V.Rojas Luis, Sistema para la clasificación de opiniones generadas en Twitter usando redes neuronales artificiales, Licenciatura, Universidad Autónoma Metropolitana Unidad Azcapotzalco, 2017.
7. L. Poot Terán, Minería de opiniones sobre textos en Twitter, Licenciatura, Universidad Autónoma Metropolitana Unidad Azcapotzalco, 2018.
8. Corona Virus (COVID-19) Tweets Dataset, IEEE Datasets, 2019 [En línea]. Disponible: https://ieee-dataport.org/open-access/coronavirus -covid-19-tweets -dataset [Accedido: 24- Apr-2021]
9. C. Arcila, F. Ortega, J. Jiménez y S. Trullenque, Análisis supervisado de sentimientos políticos en español: clasificación en tiempo real de tweets basada en aprendizaje automático, El profesional de la información, Vol. 26, pp.973-982, Sep 2017.
10. C. Solis B. Cagua, Prototipo móvil para el análisis de sentimientos a través de tweets de noticias del ecuador, Licenciatura, Facultad de ciencias matemáticas y físicas, 2021.
11. E. López, Extracción y análisis de sentimientos y tendencias sobre covid-19 en redes sociales, Licenciatura, Universidad de Málaga, 2021.
12. José Clemente Hernández Hernández, Aprendizaje Profundo y Neuroevolución para el Análisis de Sentimientos en Tweets Escritos en Español Mexicano, Maestría, Instituto de Investigaciones en Inteligencia Artificial, Universidad Veracruzana, 2021.
13. B. Quintino, M. Ávila, F. Ávila, M. Bianchetti, E. Franco, Diseño de mapa interactivo y multitáctil de supervivencia de árboles, Pistas Educativas, Vol. 39, pp. 426-436, Dic 2017.
14. PyMySQL, [En línea]. Disponible: https://pypi.org/project/PyMySQL/ [Accedido: 4-Feb-2022]
15. Leaflet, [En linea]. Disponible: https://leafletjs.com/ [Accedido: 28-Jul-2021]
16. Google Maps, [En línea]. Disponible: https://maps.google.com.mx/ [Accedido: 4-Feb-2022]

ISSN 1870-4069

My two Bitcoins? Implementation of Double-Spending on Fast Bitcoin Payments

Juan David Peña Melo[2], Saúl Eduardo Pomares Hernández[1,2], Lil María Xibai Rodríguez Henríquez[3], Julio César Pérez Sansalvador[3]

[1] LAAS-CNRS, Université de Toulouse, France

[2] INAOE, Santa María Tonantzintla, Mexico

[3] INAOE-Cátedra CONACyT, Santa María Tonantzintla, Mexico

jdom1824@inaoep.mx

Abstract. Bitcoin is a payment system that eliminates trusted intermediaries in the exchange of digital currencies. To process transactions, Bitcoin uses a set of nodes with different specialized roles that function as a trusted third party. The Bitcoin confirmation transaction is a slow process that can take up to 72 hours. However, in fast payment scenarios, products are delivered immediately. These scenarios in Bitcoin are vulnerable to double-spending attacks. Different strategies have been proposed to mitigate the double-spending attack on Bitcoin, such as allowing transactions to propagate freely in the network, inserting a new node role to detect attacks, and penalizing malicious users for revealing their identities. To the best of our knowledge, there are no works to avoid double-spending attacks on fast Bitcoin payments. This article is a guide that shows how easy it is to perform double-spending attacks on fast Bitcoin payments and highlights the vulnerability of Bitcoin when the transaction is unconfirmed. The experiments run on the Bitcoin Testnet, an environment where coins are worthless, and developers can experiment on a distributed network infrastructure. The experiments show an analysis of the speed of Bitcoin to process transactions, the low investment that a malicious user needs to make to carry out the attack, and the high probability of success of a double-spending attack in fast Bitcoin payment scenarios.

Keywords: Bitcoin, cryptocurrencies, double-spending, peer-to-peer, fast-payments.

1 Introduction

Bitcoin is a payment system created by Satoshi Nakamoto that uses distributed systems and cryptography to eliminate trusted intermediaries in the value

exchange [13]. Bitcoin does not have a central server that serves as a control point to process transactions and uses a set of nodes with different specialized roles that work as a trusted third party [18]. The confirmation of a transaction in Bitcoin is a slow process that can take up to 72 hours and is a process irreversible once the transaction is processed and confirmed in the Blockchain [12]. Bitcoin developers designed the payment system for internet sales where it is possible to wait for confirmation before delivering the product.

However, in fast payment scenarios, products are delivered immediately (in seconds order), for example, in ATMs [17] or takeaway restaurants [3]. These scenarios are vulnerable to Bitcoin double-spending attacks because the payment must be confirmed when the product or service is delivered, and the Bitcoin confirmation process is not fast enough, which increases the probability of successful double-spending attacks. In a successful double-spending attack, a malicious Bitcoin user pays twice with the same currencies, i.e., pays a seller and reverses the transaction so that the currencies go back to an address of his own [8].

Recently, have been proposals in the literature to mitigate double-spending attacks on fast Bitcoin payments. The first strategy is to propagate all transactions in the network without restrictions so that the network nodes can identify the double-spending attack [10] [9]. Another advance is to introduce observers to alert attack nodes [15]. A third approach avoids network isolation to ensure a higher probability of seeing inconsistencies related to double-spending attacks on the system [2]. Finally, another strategy is to reveal the identity of malicious users attempting double-spending attacks [16]. Currently, Bitcoin does not guarantee a complete solution for double-spending attacks on fast Bitcoin payments.

This article is a guide to a double-spending attack on fast Bitcoin payments. This guide is based on Karame's requirements for a successful double-spending attack [10] and shows in detail how a malicious user can take advantage of the distributed nature of the system to purchase products and services without spending their coins. The experiments run on the Bitcoin Testnet [4], an environment where coins are worthless, and developers can experiment on a distributed network infrastructure. The experiments show an analysis of the speed of Bitcoin to process transactions, the low investment that a malicious user needs to make to carry out the attack, and the high probability of success of a double-spending attack in fast Bitcoin payment scenarios.

2 Background

2.1 Bitcoin Overview

Bitcoin is a peer-to-peer payment system based on cryptography and distributed systems. The network's peers have various roles, such as mining nodes, full

blockchain nodes, full nodes and lightweight nodes [1, p. 172]. The purpose of peers is to propagate, verify, and confirm transactions that transfer value between network users without needing a trusted entity.

Transactions are data structures cryptographically signed by the owner that can exchange value on Bitcoin. The data structure is composed of an identifier, a pointer to the previous transition called input, and outputs that define the new owners of the coins. To launch a transaction in Bitcoin it is necessary to connect through a node. Every time a node receives a transaction, it verifies that:

1. The transaction has enough Bitcoins to consume, i.e., the output must not exceed the input.
2. The input is spent once.
3. The digital signature is authentic.

Once the transaction is verified, the nodes store it in a memory space called a Mempool [5], where it waits for confirmation. The confirmation process refers to inserting a transaction into the Blockchain through "mining" a new block. The miners are a set of nodes that reach a consensus through a non-deterministic process to insert the new blocks in the Blockchain. The mining process is beyond the scope of this article, as a successful double-spending attack on fast Bitcoin payments does not need computational power.

2.2 Karame's Model

The Karame model consists of a malicious user and a seller connected to the Bitcoin network. The malicious user wants to buy a product from the seller without spending his coins. To achieve this, the malicious user carries out a double-spending attack. The double-spending attack concerns spending the same currency twice [10]. In Karame's model, the malicious user controls multiple nodes that help execute the attack since he cannot sign two transactions that spend the same currency on a single node. The malicious user does not have enough computational power to create a block, and a transaction belonging to a block is considered irreversible [9].

Necessary Conditions for Successful Double-Spending A successful double-spending attack is performed as follow: Alice creates two transactions that spending the same currency $Tr(A)$ and $Tr(B)$, the transaction $Tr(B)$ pays Bob for the product Alice wants to buy, while transaction $Tr(A)$ returns the coin to Alice. Notice that the double-spending attack is a competition in the propagation of the two transactions to belong to the Blockchain. To achieve a successful double-spending attack, Alice must ensure that $Tr(A)$ belongs to the Blockchain before transaction $Tr(B)$ by meeting the following requirements:

– Requirement 1 - $Tr(B)$ is added to Bob's Mempool. If $Tr(B)$ is not added to Bob's Mempool, then there is no product delivery because there is no evidence that Alice wants to pay Bob.

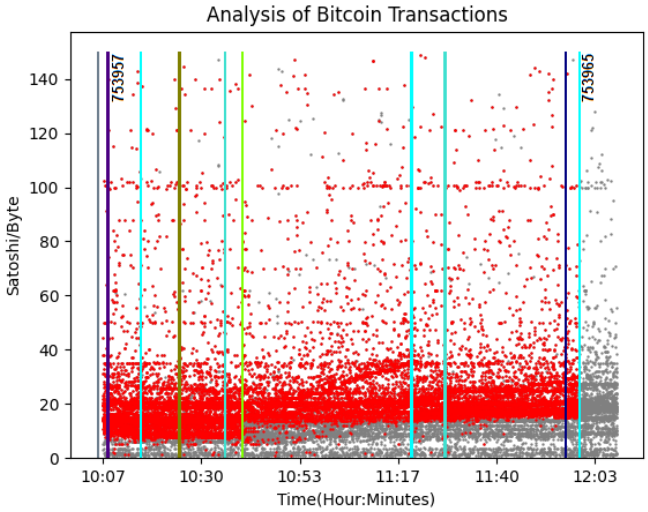

Fig. 1. Confirmation of transactions in the blockchain and stagnation of transactions in the Mempool. This graph can be generated with the repository by publishing in [11].

– Requirement 2 - $Tr(A)$ is confirmed on the Blockchain. If transaction $Tr(B)$ is confirmed before transaction $Tr(A)$, Alice will not be able to get her coins back and, Bob will receive payment for the product.

– Requirement 3 - Bob's product delivery time is less than Alice's misbehavior detection. Bob needs to be in a fast-payment scenario to ensure the product is delivered before the Blockchain confirmation.

3 Analysis of Bitcoin Transactional Processing

We analyze the Bitcoin Mempool with 30,000 transactions between blocks 753957 and 753965. We show this data in Figure 1, where gray dots are confirmed transactions and red dots are unconfirmed transactions. On the x-axis, we show the observation and data capture time, while on the y-axis, we see the amount of fee a transaction pays per byte. Finally, the colored bars represent the instant in time where a block is added to the blockchain.

Analyzing Figure 1 we find that there are two scenarios where a malicious user can perform successful double-spending attacks:

First Scenario: in Bitcoin, each transaction pays a fee for the number of bytes added to the Blockchain. Fees are rewarding for miners to receive for the computational power invested in the network. Most nodes adjust the Satoshi/Byte fee (Satoshi is the smallest unit of a Bitcoin that can be sent, that is, hundredth of a millionth Bitcoin) based on the number of transactions in the

Mempool [7]. Figure 1 shows from 10:30 to 10:53 GMT-5 the increase in fees for the number of transactions in the Mempool. Note that Bitcoin's transaction processing capacity is a bottleneck for the system, causing many transactions to remain on hold without confirmation for up to 72 hours or eventually be discarded [12]. During the propagation and confirmation time of a transaction, a malicious user has the advantage of performing a double-spending attack on Bitcoin, propagating two transactions with low-fee that spend the same currency and obtaining a product or service from a merchant who does not wait for the confirmation.

Second Scenario: Bitcoin creates a block every 10 minutes on average. However, Figure 1 shows that there are long time intervals in the creation of new blocks, i.e., the standard deviation is high, which means that sometimes there are time intervals of up to 40 minutes between one block and another [6]. This standard deviation also increases the risk of a successful double-spending attack in fast payment scenarios. Because if a malicious user spreads two transactions that spend the same currency with a high fee, he would still have the long confirmation times, which are sufficient in scenarios where the exchange of products and services is immediate.

4 Related Work

The double-spending attack on Bitcoin fast payments is a vulnerability detected by Karame et. al in [10] [9]. The author identifies the necessary requirements for a successful double-spending attack based on his Bitcoin fast-payment model. They propose that all transactions, including inconsistent ones, propagate without restrictions in the network so that every node can detect double-spending attacks. However, the mechanism proposed by Karame could generate inconsistencies in the network for a prolonged period, affecting the consensus between the nodes and causing a denial of service attacks.

Based on the fast payments of Bitcoin and the vulnerability proposed in the Karame model, T. Bamert et. al in [2] propose a strategy to identify double-spending attacks by connecting to random nodes and listening for transactions in the network that have inconsistencies. This solution can identify double spending attacks with a high percentage as long as the set of nodes is permissioned and limited. However, the Bitcoin network is permissionless, and its pool of nodes tends to grow over time.

C. Pérez-Solà et. al in [16] proposes to mitigate the double spending attack by penalizing malicious users. For this, it uses a vulnerability of the digital signature scheme based on the elliptic curve to reveal the private key of the attackers. However, revealing the users' private key in an asymmetric encryption scheme is undermining the security foundation of Bitcoin, and this would cause a risk to possibly not malicious users.

The works mentioned before are the most relevant in the literature on double-spending attacks on fast Bitcoin payments. However, the proposed

Table 1. Bitcoin Testnet vs Bitcoin Mainnet

Bitcoin Testnet	Bitcoin Mainnet
No monetary value	Real value
The difficulty restarts	The difficulty is variable
Port 18333	Port 8333
Transaction Frequency Low	Transaction Frequency High
No economic benefit for mining	Economic incentive for mining
The data is periodically deleted	Traceability from the Genesis Block
Connection Port RPC 18332	Connection Port RPC 8332

strategies do not guarantee 100% attack detection, opening a topic for future research.

5 Double-Spending Attacks on Bitcoin Testnet

In Bitcoin, transactions have no temporal constraints, and the confirmation process starts when a transaction is added to a block. Bitcoin payments are slow and refer to a scenario where the merchant waits for up to 6 confirmations from the Blockchain to deliver the product. However, the double spending attacks shown in this section are based on Karame's model, they set up a scenario of fast payments without waiting for any confirmation to deliver the product, and this scenario runs our attacks.

The section is organized as follows: first, the differences between the Bitcoin Mainnet and the Testnet. Second, the hardware and software used in the attacks. Third, a description of the transactions that spend the same currency, then the attacks are described, and finally, a discussion about the results found.

5.1 Difference between Bitcoin Testnet and Bitcoin Mainnet

Bitcoin has two disjoint Blockchains: Bitcoin Mainnet and Bitcoin Testnet. Developers used Bitcoin Testnet as environment to test without spending money or causing inconsistencies in the Bitcoin Mainnet. Coins on Testnet are separate from real Bitcoins and never have value. The differences are shown below:

The transactional verification process, from creating a transaction to adding it to the blockchain, is similar on both Blockchains [1]. Therefore, a double-spending attack on fast Bitcoin payments runs the same on Testnet and Mainnet. However, since the frequency of transactions is higher on the Mainnet, the fee for each transaction is more expensive. The scenario posed in the Karame model [9] refers to a fruit seller who receives fast Bitcoin payments. This scenario is not profitable for a malicious user since executing a double-spending attack on the Bitcoin Mainnet would be more expensive than the product. However, not all fast payment scenarios handle low amounts. For example, an ATM [17] exchanging Bitcoin for cash can be a high-money-loss

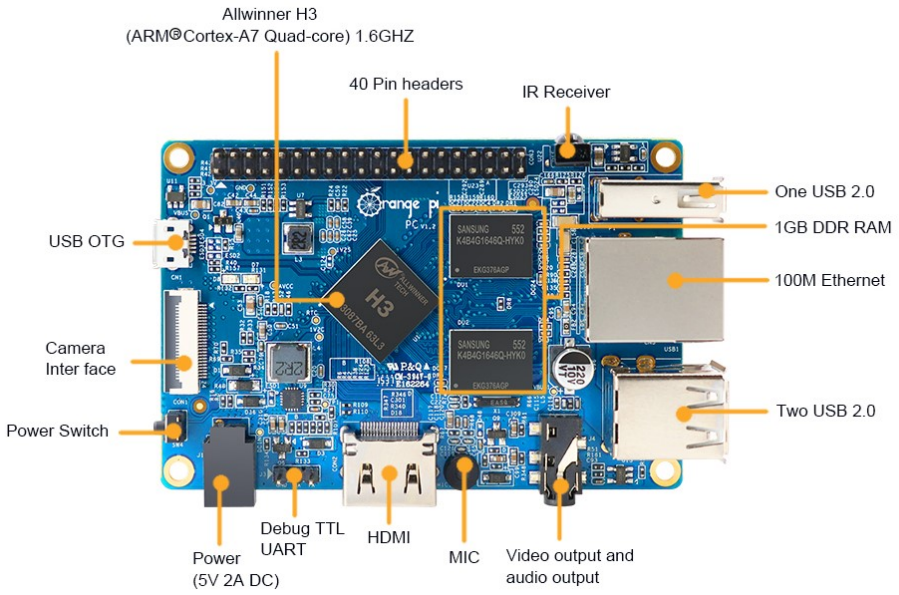

Fig. 2. Orange Pi PC minicomputer with hardware specifications [19].

scenario if a malicious user achieve a successful double-spending attack, as shown in section 5.4 Figure 8.

5.2 Hardware and Software

We perform double-spending attacks on Bitcoin Testnet [4] following Karame's model with the two scenarios seen before. The hardware used for these experiments is an Orange pi PC [19] minicomputer show in Figure 2, under the Armbian Buster [14] operating system. This minicomputer is chosen for its low cost, and power consumption. It also meets technical specifications that require a Bitcoin node to store the Blockchain.

For the implementation of the attacks, the Bitcoin Core software is installed on 3 Orange pi PC in the role of lightweight nodes. The nodes simulate the behavior of a malicious entity named Alice, an entity merchant named Bob, and an Alice's Helper Node. The process of synchronizing the nodes with the network takes up to 3 hours. Subsequent, each node must be able to send and receive Bitcoin transactions via a generated public address and private key. The addresses are generated in the Bitcoin Core console with the `getnewaddress` command. The public key is similar to a bank account number and is used to receive transactions. The private key signs transactions to be propagated on the Bitcoin network.

```
createrawtransaction
'[{"txid":"894c8df02b155135b058ed9434b98ba36b062415eeb58cb4588adc58e888add6","vou
t":0,"scriptPubKey":"a914027f39211ddd0e234908d0cedb85fcfa8514019c87"}]'
'{"tb1qdpje0kwwez9kd22cmxkd3xdvt9ttyhpnm3w5hf":
0.00009,"2MsURkHeaFnhEVnFbAfLVXUGj8HmfDeRKPS":0.000003}'
0200000001d6ad88e858dc8a58b48cb5ee1524066ba38bb93494ed58b03551152bf08d4c890000000
000ffffffff0228230000000000001600146867d9cec88b66a958d9acd899ac5956b25c332c0100
0000000000017a914027f39211ddd0e234908d0cedb85fcfa8514019c8700000000
```

Fig. 3. Creation of the malicious transaction Alice $Tr(A)$.

```
createrawtransaction
'[{"txid":"894c8df02b155135b058ed9434b98ba36b062415eeb58cb4588adc58e888add6","vou
t":0,"scriptPubKey":"a914027f39211ddd0e234908d0cedb85fcfa8514019c87"}]'
'{"2MwnrzMFDYiVVdLwmgaAh2zuiv3ibd6SqAo":
0.00009,"2MsURkHeaFnhEVnFbAfLVXUGj8HmfDeRKPS":0.000003}'
0200000001d6ad88e858dc8a58b48cb5ee1524066ba38bb93494ed58b03551152bf08d4c890000000
000ffffffff022823000000000000017a91431dc48f21516212b42d8c9218f7d321b97948fb8872c01
00000000000017a914027f39211ddd0e234908d0cedb85fcfa8514019c8700000000
```

Fig. 4. Creation of the transaction that trying to pay Bob $Tr(B)$.

```
sendrawtransaction
02000000000101d6ad88e858dc8a58b48cb5ee1524066ba38bb93494ed58b03551152bf08d4c890
0000000171600145a9555c890f6e0d5753a90c2903ede70e45757c7ffffffff0228230000000000
0017a91431dc48f21516212b42d8c9218f7d321b97948fb8872c0100000000000017a914027f392
11ddd0e234908d0cedb85fcfa8514019c870247304402204c973e2138dc2847d0f69e2a3d49de94
56993a9111cc0e97e2d825a8c4b5e7460220269cd0b8c08b84f7109332f717443a776a03f41007e
89094f67dc5364c733be30121024bc6ff8ef128e307b3e3a06acf7f57418df4c1555ffa11110084
7a12ff70e2a000000000

ce18f033e2c0cb0032194ca93979773d53d17528864cdf4d7c05aab9f9b1f21e
```

Fig. 5. Propagation of $Tr(B)$ transaction to the Bitcoin Testnet network.

5.3 Create and Propagate Bitcoin Transactions that Spend the Same Currency

Bitcoin transactions are created by pointing to the identifier txid of the previous transaction and the output to be consumed. We assume that Alice makes two transactions pointing to the same identifier txid and the same output. $Tr(A)$ is a malicious transaction that returns the coins to Alice, and $Tr(B)$ is a transaction that tries to pay Bob for the product. Figure 3 and Figure 4 show the creation of two transactions $Tr(A)$ and $Tr(B)$.

Note that in Figures 3 and 4, transactions point to the same identifier txid and have a different recipient, i.e., the two transactions spend the same currency. After the transactions are created and signed, be propagated on the network. The propagation of the transactions to the network is done through the sendrawtransaction command. Figure 5 shows the propagation of the transaction $Tr(B)$, the result is a new identifier for the created transaction.

Bitcoin software does not allow signing two transactions that spend the same currency [12]. Therefore, Alice uses the Helper Node to sign the $Tr(A)$ transaction. In the next section, multiple transactions spending the same currency will be created and propagated according to the Karame model to observe the probability of success of double-spending attacks on fast Bitcoin payments.

5.4 Experiments

To satisfy Karame's requirements, we connect Alice's node directly to Bob's node, which satisfies requirement 1. We assume that Bob delivers the product to Alice once he sees transaction $Tr(B)$ in his Mempool, which satisfies requirement 3. Finally, to observe the probability of confirmation of transaction $Tr(A)$ to the blockchain, we will modify Alice's Helper Node connections and the propagation time of transaction $Tr(A)$ versus transaction $Tr(B)$.

The propagation of transactions $Tr(A)$ and $Tr(B)$ are made with a time difference shown on the x axis of Figure 6. For example, if the time is equal to 1 second, it means that $Tr(A)$ propagated 1 second after $Tr(B)$, and if time is equal to -4 seconds means that $Tr(A)$ propagated 4 seconds before $Tr(B)$. Therefore, when the time difference is 0, it means that the two transactions were propagated at the same time. For every time difference, 10 attacks on Bitcoin Testnet [4], the probability that $Tr(B)$ belongs to the blockchain is equal to $Tr(A) - 1$.

In the first set of attacks, Bob is connected to eight nodes, including Alice, and the Alice's Help Node connect to eight nodes without relationship to Bob's connections. The experiment aims to observe the probability of success of a double-spending attack in an uncontrolled environment with the Karame's model using the default configuration of the Bitcoin Core Testnet. Figure 6 shows that although Bob and Alice's Helper Node have the same eight connections, they are under an uncontrolled environment. Because when the difference in propagation time of transactions is 0, the malicious transaction $Tr(A)$ propagated by Alice's Helper Node has only a 20% chance of confirmation in the Blockchain. Also, when the propagation time of transaction $Tr(B)$ is delayed, the probability of adding transaction $Tr(A)$ to the blockchain increases.

In the second set of attacks, Alice's Helper Node connects to 50 nodes. We emphasize that increasing the number of connections increases the waiting time between the connections of each node. The connection timeout parameter in this experiment is 15 seconds per node. This experiment aims to observe the probability of confirmation of Alice's $Tr(A)$ malicious transaction when connections to Alice's Helper Node are increased. Although the environment is not controlled, the probability of confirmation of the transaction $Tr(A)$ should increase compared to the graph before.

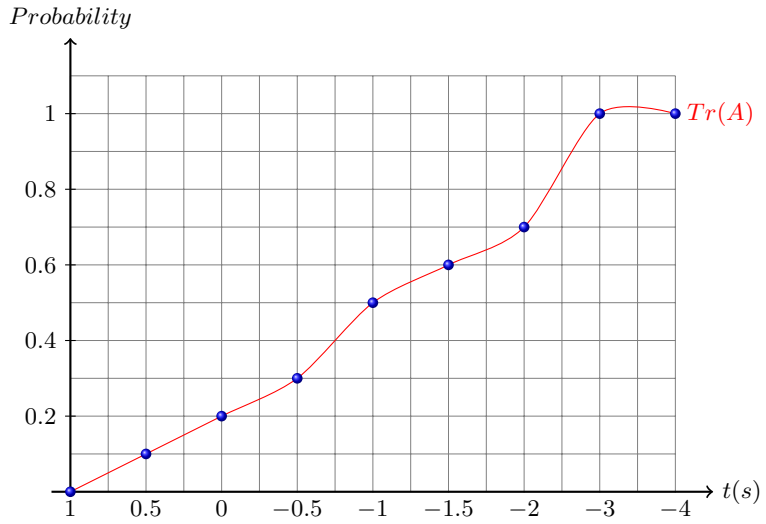

Fig. 6. Probability of success of the malicious transaction, when the attacking node and Bob's node connect to the same number of nodes.

Figure 7 shows a relevant increase in the probability of success of the malicious transaction. For attacks with one second of difference, the confirmation increased by 10% compared to attacks in the previous experiment. Also, the time the malicious transaction $Tr(A)$ reaches 100% of the probability of confirmation is reduced. However, the advantage of transaction $Tr(B)$ continues with a high percentage of success when the two transactions propagate at the same time. This modification of Alice's Helper Node connections shows how easy it is to give the malicious transaction $Tr(A)$ an advantage to satisfy requirement 2.

In the third set of experiments, Alice's Help Node connections increase to 100 nodes. The wait parameter between node connections is the same as in the previous experiment. We find that Orange pi's resources are limited to run this experiment, and a more powerful computer is used (Laptop Core I7 9th Generation with 16 Gb Ram and 1 Gigabit Ethernet Port). The experiment aims to increase the connections of Alice's helper node to 100 nodes and to observe if the probability of confirmation of the malicious transaction $Tr(A)$ increases proportionally. Figure 8 shows the result of increasing the connections of Alice's Help Node, increasing the probability of confirmation of the malicious transaction $Tr(A)$. However, the advantage obtained is low compared to the previous experiment, and the change in the hardware and software is costly.

5.5 Discussion

Finally, the experiments are an implementation following Karame's model for double-spending attacks successfully and supported by the analysis of the

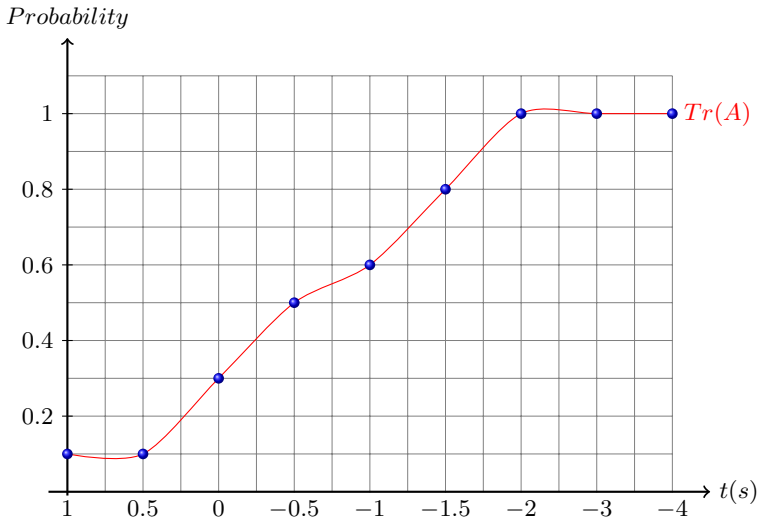

Fig. 7. Probability of success of the malicious transaction, when the attacking node connects to fifty nodes and Bob connects to eight nodes.

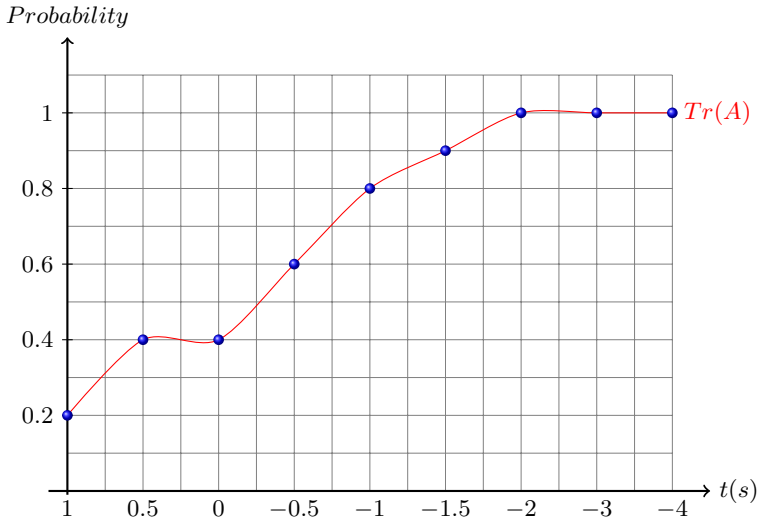

Fig. 8. Probability of success of the malicious transaction, when the attacking node connects to hundreds of nodes and Bob connects to eight nodes.

Bitcoin Mempool. We modify the variables to satisfy requirement 2, such as the propagation time difference between the malicious transaction $Tr(A)$ and

the transaction trying to pay Bob $Tr(B)$, also the number of connections. We note that there is a high probability of success of the double-spending attacks on fast Bitcoin payments under an uncontrolled environment such as the Bitcoin Testnet. The probability of confirmation $Tr(A)$ can increase with more complex attacks. We highlight that the experiments used hardware with limited resources Orange pi PC, and only necessary to modify the hardware in the last experiment. However, the probability did not increase as expected. Finally, it is possible to increase the probability of success of these attacks if the network connections are analyzed.

6 Conclusions

In this article, Bitcoin's vulnerability to double-spending attacks in fast payment scenarios was shown at a low level. The attacks were implemented on Karame's model and the analysis of transactional processing. The number of connections of Alice's Helper Node and the transactions' propagation time were modified to measure the attack success probability. The experiments showed a 70% success probability using a low-cost device such as Orange pi PC. This vulnerability stops the massive adoption of Bitcoin and leaves an open issue to develop mechanisms that avoid double-spending attacks, as future work remains to analyze the Karame model from a time-based logical distributed view and find the necessary and sufficient requirements for a double spending attack.

References

1. Antonopoulos, A. M.: Mastering Bitcoin: Unlocking Digital Crypto-Currencies. O'Reilly Media, Inc., 1st edn. (2014)
2. Bamert, T., Decker, C., Elsen, L., Wattenhofer, R., Welten, S.: Have a snack, pay with bitcoins. In: IEEE P2P 2013 Proceedings. pp. 1–5 (2013) doi: 10.1109/P2P.2013.6688717
3. Bastardo, J.: Usé bitcoin para pagar una hamburguesa en burger king y te lo cuento. https://es.cointelegraph.com/news/i-used-bitcoin-to-pay-for-a-burger-at-burger-king (2020)
4. Bitcoin, W.: Testnet. https://en.bitcoin.it/wiki/Testnet (18-06-2020)
5. Blockchain: Bitcoin developer reference. https://bitcoin.org/en/developer-reference (2020)
6. Blockchain.com: Bitcoin explorer mempool size. url = https://www.blockchain.com/es/explorer (2022)
7. Blockchain.com: Bitcoin explorer price. url = https://www.blockchain.com/es/explorer (2022)
8. Herrmann, M.: Implementation, evaluation and detection of a doublespend-attack on bitcoin (2012)
9. Karame, G. O., Androulaki, E., Capkun, S.: Double-spending fast payments in bitcoin. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. pp. 906–917. CCS '12, Association for Computing Machinery, New York, NY, USA (2012)

10. Karame, G. O., Androulaki, E., Roeschlin, M., Gervais, A., Čapkun, S.: Misbehavior in bitcoin: A study of double-spending and accountability. ACM Trans. Inf. Syst. Secur., vol. 18 (2015)
11. Melo, D.: Github - jdom1824/graph_mempool (2022), https://github.com/jdom1824/Graph_Mempool
12. Nakamoto, S.: Bitcoin web. https://developer.bitcoin.org/ (01-06-2020)
13. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2009)
14. Pečovnik, I.: Armbian. https://www.armbian.com/ (19-06-2020)
15. Podolanko, J. P., Ming, J.: Countering double-spend attacks on bitcoin fast-pay transactions. In: ieee-security. pp. 1–5 (2017)
16. Pérez-Solà, C.: Double-spending prevention for bitcoin zero-confirmation transactions. UCL Discovery, (2019)
17. Quirós, F.: Cajeros automáticos de bitcoin en colombia: ¿dónde están y cuántos hay? https://es.cointelegraph.com/news/bitcoin-atms-in-colombia-where-are-they-and-how-many-are-there (2020)
18. Rosenfeld, M.: Analysis of hashrate-based double spending (2014)
19. Xunlong, S.: Orange pi pc. http://www.orangepi.org/orangepipc/ (2015)

ISSN 1870-4069

Análisis de texto tóxico en redes sociales

Miguel Soto, Hiram Calvo, Alexander Gelbukh

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciencias Cognitivas Computacionales

msotoh2021@cic.ipn.mx,
hcalvo@cic.ipn.mx, gelbukh@cic.ipn.mx

Resumen. Las redes sociales se han convertido en una parte importante de nuestra sociedad y, en particular, de la vida de los jóvenes. En muchos casos, las redes sociales se convierten en un lugar para que la gente exprese sus opiniones y comparta sus pensamientos con el mundo. Mientras que los usuarios puedan expresarse libremente, esto también puede conducir a la propagación del lenguaje tóxico; ya que por desgracia algunos usuarios utilizan lenguaje intencionalmente hiriente, despectivo o dañino, y en algunos casos como forma de ciberacoso. Este tipo de comportamiento tóxico es un problema al que hay que hacer frente. Es por esto que en este trabajo proponemos la revisión de distintos modelos computacionales tanto de aprendizaje automático como de aprendizaje profundo, los cuales trabajarán bajo *K-Folds Cross-validation* con las mismas condiciones, con la finalidad de identificar cuál de estos modelos se desempeña de mejor manera para resolver el problema de identificar el texto considerado como tóxico; y en caso de ser identificado de esta manera, definir a qué etiquetas corresponde como parte de la explicación del por qué es tóxico.

Palabras clave: Lenguaje tóxico, redes sociales, ciberacoso, aprendizaje automático, aprendizaje profundo, validación cruzada, clasificación de texto, procesamiento de lenguaje natural.

Toxic Language Analysis on Social Media

Abstract. Social media has become an essential part of modern life, especially among young people, by enabling the free expression of ideas and opinions. However, this openness also facilitates the spread of toxic language, characterized by hurtful, derogatory, or harmful expressions that, in many cases, manifest as cyberbullying. This work proposes the review and comparison of different machine learning and deep learning models, evaluated under a K-Folds cross-validation scheme with homogeneous conditions. The objective is to determine which model achieves the best performance in toxic text detection and, additionally, to classify the text into different toxicity categories (e.g., insults, threats,

discrimination), thus providing a more complete explanation of why a text is considered toxic.

Keywords: Toxic language, social media, cyberbullying, machine learning, deep learning, cross-validation, text classification, natural language processing (NLP).

1. Introducción

En los últimos años se ha visto un crecimiento exponencial respecto al número de personas que utilizan redes sociales, y a pesar de que pueden ser una herramienta muy poderosa para las interacciones humanas virtuales, como conectar personas a largas distancias o inclusive para hacer crecer un negocio, estas traen consigo algunos problemas. Uno de ellos es la mala comunicación con los demás usuarios, esto sucede debido a que las personas que interactúan con los contenidos que ofrecen estos sitios no necesariamente tienen las mismas opiniones cuando algún tema surge en una discusión. Sumándole a esto, el poco control de estas plataformas para erradicar el mal uso, incita a las personas a utilizar lenguaje tóxico. Para este trabajo definiremos el lenguaje tóxico como: "Aquel lenguaje que tiene como intención insultar, amenazar o mostrar desprecio hacia una persona o grupo de personas". El uso de este lenguaje ha demostrado tener un impacto negativo en las personas convirtiéndolo en una de las razones que corrompe la salud mental de los adolescentes [21] y una de las principales razones de suicidio [13]. Por consiguiente, el detectar esta clase de lenguaje es un problema relevante que necesita ser tomado en cuenta.

Existen diferentes enfoques y herramientas disponibles para identificar el lenguaje tóxico u ofensivo, pero se han diseñado con un propósito muy general: estos enfoques proponen la extracción de palabras clave de los textos con contenidos tóxicos con criterios como la frecuencia. En este sentido, esto puede representar un problema debido a que una palabra relevante para un texto que no es ofensivo no debe catalogarse como una palabra clave. Por tal motivo es necesario identificar las palabras que son relevantes en los textos ofensivos y a su vez poco relevantes en los textos no ofensivos.

La razón principal por la que debemos erradicar el lenguaje tóxico de las redes sociales es para evitar que los usuarios que hacen uso de estas plataformas sigan sufriendo, ya que a menudo las personas que utilizan este tipo de lenguaje juzgan a los demás por sus publicaciones en redes sociales, y a partir de dichas publicaciones más usuarios pueden realizar comentarios ofensivos. De esta manera, las personas que ven comentarios ofensivos hacia otra persona pueden sentirse tentados a tomar represalias sobre la gente que esta haciendo daño. Esto genera un círculo vicioso, en el que la gente publica comentarios hirientes sobre otros, lo que provoca mas comentarios y palabras hirientes. Para contrarrestar el lenguaje ofensivo vive en la Internet, algunas redes sociales como Facebook han implementado medidas para detectar texto en imágenes con ayuda de algoritmos tipo OCR (Optical Character Recognition) [4].

Mientras que otros sitios suelen revisar manualmente o tener moderadores que revisen publicaciones y comentarios con la finalidad de eliminar este tipo de comentarios. Sin embargo, este tipo de revisiones manuales requieren de mucho trabajo, por lo que no son sostenibles ni escalables en el tiempo.

Es por esto que en este trabajo proponemos la creación de un modelo que sea capaz de implementar una mejor clasificación entre comentarios tóxicos a través de las múltiples subcategorías que este pueda tener y los comentarios limpios o libres de toxicidad.

2. Estado del arte

Debido a que la detección del lenguaje tóxico u ofensivo es una de las grandes problemáticas que se han buscado erradicar en el campo del procesamiento del lenguaje natural, se han realizado numerosas investigaciones al respecto, comenzando desde lo más básico del lenguaje utilizando simplemente las características léxicas como diccionarios [17] o la bolsa de palabras (BoW) [5]. Sin embargo, y a pesar de que el uso de las características léxicas funcionan, es necesario entender el contexto o la estructura sintáctica del texto que queremos analizar. Es por esto que más adelante se utilizaron enfoques basados en N-gramas donde se presenta un conjunto de datos con las etiquetas de *odio*, *lenguaje ofensivo* o *ninguna*, y se construye un clasificador basado en características con transformación TF-IDF (*Term frequency – Inverse document frequency*) sobre n-gramas, *part of speech*, análisis de sentimientos, entre otras características, donde el mejor modelo presentado por los autores fue entrenado utilizando regresión logística [8].

En 2017 [2], los autores experimentan con múltiples arquitecturas de aprendizaje profundo para la tarea de detección de discursos de odio en Twitter, obteniendo sus mejores puntuaciones F1 utilizando redes de memoria a largo plazo (LSTM) [14] y refuerzo de gradiente [2]. Esto daría pie a más desarrollos de aprendizaje con arquitecturas similares, como Georgakopoulos y colegas optan por comparar las Redes Neuronales Convolucionales (CNN) con el enfoque tradicional de bolsa de palabras [11], combinándolo con una selección de algoritmos como k-Vecinos más Cercanos (kNN), Máquinas de Vectores de Soporte (SVM), Naive Bayes (NB) o Asignación Latente de Dirichlet (LDA) [19] que utilizan una CNN híbrida con la intuición de que la entrada a nivel de caracteres contrarrestaría las palabras mal escritas a propósito o por error y los vocabularios inventados, demostrando que las CNN superan los enfoques tradicionales de la minería de textos en la clasificación de comentarios tóxicos, lo que supondría un gran potencial para el desarrollo de la clasificación de identificación de comentarios tóxicos.

Desde la introducción de algoritmos basado en *transformers*, el uso de los modelos lingüísticos preentrenados en la clasificación de texto se ha convertido en la corriente principal de investigación, ya que el proceso básico consiste en añadir capas específicas para la tarea a realizar, posteriormente añadir el modelo preentrenado y, a continuación, entrenar el nuevo modelo en el

que solo se entrenan las capas específicas de la tarea desde cero. Algunos de los modelos preentrenados más utilizados por la comunidad científica son *Bidirectional Encoder Representations from Transformers* (BERT) [9], *A Robustly Optimized BERT Pretraining Approach* RoBERTa [18], y *Cross-lingual Language Model Pretraining* XLM [16]. El uso de estos modelos se debe a que fueron preentrenados con corpus extremadamente grandes, como en el caso de BERT [9] que fue entrenado con mas de tres mil millones de palabras, y para este tipo de modelos por lo general las capas de salida se sustituyen por las capas específicas de la tarea a realizar y este nuevo modelo se entrena de forma supervisada.

El uso de los modelos preentrenados ha dado pie a la mejora de estos, ya sea añadiendo capas de atención (LSTM) entre la capa de clasificación lineal y el modelo preentrenado [3,7] o explorando el prentrenamiento continuo utilizando corpus del dominio correspondiente y luego transfiriendo el modelo recién afinado a las tareas de clasificación objetivo [12]. De esta manera la literatura nos expande el horizonte para ajustar los modelos preeentrenados a nuestra conveniencia para poder lograr una mejor exactitud a la hora de probar nuestros modelos.

En cuanto a los conjuntos de datos que se han utilizado a lo largo del tiempo, hay que tomar en consideración que la mayoría se han realizado para el idioma inglés. Por mencionar algunos, en 2017 Davidson y sus colegas presentan un conjunto de datos con alrededor de 25,000 tuits anotados con las etiquetas: odio, lenguaje ofensivo o ninguno de los dos [8]. Posteriormente en 2018, Founta y sus colegas anotan un conjunto de datos de 80,000 tuits y los dividen en ocho categorías: ofensivo, abusivo, discurso de odio, agresivo, ciberacoso, spam y normal; su anotación para su uso a gran escala se divide en cuatro categorías que son: abusivo, odioso, normal o spam [10]. OLID (*Offensive Language Identification Dataset*) [22], presentado en 2019, contiene únicamente 14,000 tuits anotados manualmente como ofensivos y no ofensivos. Para 2020, Kurrek y colegas presentan un conjunto de datos el cual fue dividido en múltiples etiquetas, las cuales son: despectivo, apropiado, no despectivo/no apropiado, homónimos y ruido [15].

A pesar de que para el idioma español no exista la misma cantidad conjuntos de datos para la detección de texto ofensivo o tóxico como para el inglés, existen algunos que vale la pena mencionar. El primero de ellos fue presentado en 2018, donde se anotaron 11,000 tuits en español mexicano en 2 etiquetas: agresivo y no agresivo [6]. En 2021, [1] presentan un conjunto de datos con la finalidad de promover la detección del lenguaje ofensivo en las variantes del español, y se encuentra dividido en las siguientes categorías: ofensivo, el objetivo es una persona (OFP), ofensivo, el objetivo es un grupo de personas o un colectivo (OFG), ofensivo, el objetivo es diferente de una persona o un grupo (OFO), no ofensivo, pero con lenguaje impropio (NOE), no ofensivo (NO).

2.1. Diferencia de esta propuesta con respecto al estado del arte

Tras la investigación del estado del arte, proponemos una estrategia distinta para el preprocesamiento del texto, el tratamiento de los datos, así como para el uso de los modelos propuestos que se encuentran descritos en la sección 3. Esto con la intención de observar si hay alguna mejora significativa o inclusive si tienen un peor desempeño los resultados al compararlos contra el estado del arte, y de igual manera para realizar una comparación contra modelos basados en *transformers* dónde se busca obtener resultados similares con mayor rapidez.

3. Desarrollo de la solución

El proceso que se llevará a cabo para implementar la solución al problema que busca resolver este trabajo constará de distintos pasos, los cuales se muestran en el esquema presentado en la figura 1. Cada una de las etapas será descrita a detalle más adelante en esta sección.

3.1. Recolección de datos

Para comenzar la implementación de la solución de este trabajo nos enfocaremos en trabajar el conjunto de datos obtenido por [8], al cual nos referiremos a partir de este momento como HSOL (por sus siglas en Inglés, *Hate Speech and Offensive Language*), con la finalidad de realizar una comparativa contra el estado del arte. HSOL es una recopilación de tuits, el cual fue dividido en 3 categorías por trabajadores de CrowdFlower (después llamado Figure Eight y posteriormente Appen)[1], dichas categorías son: *hate speech*, *offensive language*, *neither*. Posteriormente, con la finalidad de observar que el procedimiento realizado en este trabajo no solo funciona en HSOL, se hará uso del conjunto [15] presentado en el estado del arte, el cual será llamado SLUR para futuras referencias.

3.2. Preprocesamiento

El preprocesamiento del texto para los conjuntos de datos es muy similar ya que el objetivo es que la limpieza del texto sea lo más parecida posible.

Para lograrlo se realizaron los siguientes pasos:

- En caso de existir etiquetas de entidades HTML, serán eliminadas.
- Si el texto contiene *hashtags*, se separará por palabras en mayúsculas, en caso de ser algún acrónimo, este quedará como acrónimo.
- Se eliminarán los nombres de usuario y enlaces a sitios web.
- Para el idioma inglés, se expandirán las contracciones que se encuentren de manera tal que se pueda visualizar de una mejor forma el texto. Ejemplo: *didn't* pasaría a ser *did not*.

[1] https://appen.com/figure-eight-is-now-appen/

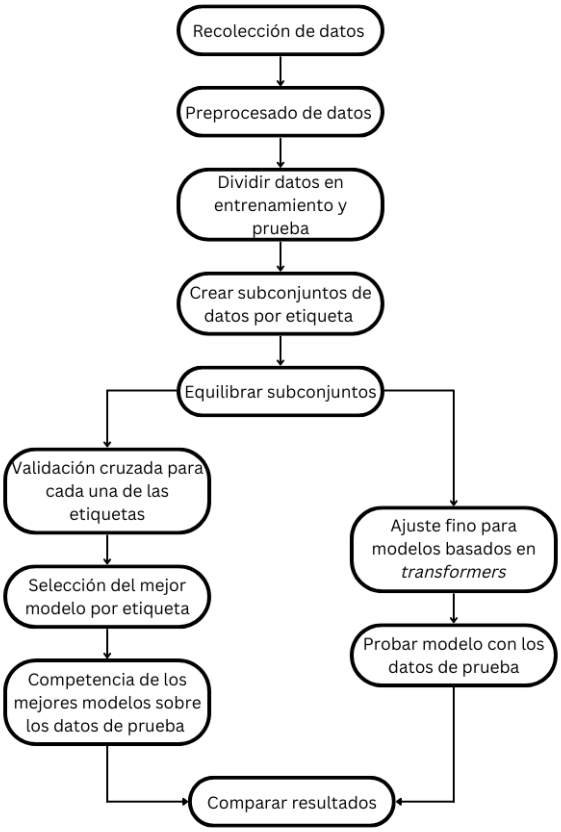

Fig. 1. Esquema planteado para resolver el problema.

- Los signos de puntuación serán removidos.
- Los emojis serán convertidos a texto.
- Se eliminarán las palabras que contengan números. Ejemplo: *y0u*.
- Todo el texto se transformó a minúsculas.

3.3. Partición de datos en entrenamiento y prueba

Como se menciona en la sección 3.1 se busca realizar una comparación contra el estado del arte, por lo cual tomaremos las características que se usaron en el trabajo de [8] para todos los conjuntos de datos. Por lo tanto, la partición para los datos quedaría de la siguiente manera: 90 % para datos de entrenamiento y 10 % para datos de prueba.

Tabla 1. Ejemplo de forma deseada del conjunto de datos.

Texto	Etiqueta 1	Etiqueta 2	...	Etiqueta n
colored contacts in your eyes	0	0	...	0
basic bitch starter pack	0	1	...	1
...	...	...	...	...

Tabla 2. Ejemplo de la forma final de los subconjuntos de datos.

Texto	Etiqueta 1
colored contacts in your eyes	0
basic bitch starter pack	1
...	...

3.4. Crear y equilibrar subconjuntos por etiqueta

En esta sección buscamos en primera instancia que los conjuntos de datos de entrenamiento tengan la misma forma, tal como se muestra en la tabla 1, donde cada etiqueta será marcada con un 1 si corresponde a una etiqueta y se marcará con un 0 en caso contrario. Este proceso se realizará para cada conjunto de datos, ya que estos en su forma original se encuentran con una estructura diferente a lo que buscamos.

Una vez que los conjuntos de datos tienen la misma forma, es hora de dividirlos por etiqueta y los llamaremos subconjuntos de tal manera que nos queden tal y como se muestra en la tabla 2, donde cada una de estas tablas contendrá todos los documentos del conjunto de datos. Posteriormente, identificaremos cuál es la parte de texto limpio, ya sea si es proveniente de alguna etiqueta o si tenemos que realizar algún otro tipo de manipulación de los datos. Cuando tengamos identificados los documentos limpios de los documentos tóxicos, pasaremos a equilibrar cada uno de nuestros subconjuntos con la finalidad de que estos tengan 50 % de documentos limpios y 50 % de documentos pertenecientes a la etiqueta de texto tóxico del subconjunto.

En la tabla 3 se muestran los subconjuntos de datos creados con la cantidad respectiva de documentos limpios, tóxicos y el total de documentos. Como se puede observar ahora todos los subconjuntos se encuentran equilibrados.

3.5. Implementación de los modelos

Aprendizaje automático. Para evaluar los conjuntos de datos utilizaremos algunos de los modelos más utilizados para tareas de clasificación. En este trabajo en concreto se eligieron los siguientes modelos de aprendizaje automático que podemos encontrar en la librería de Python *sklearn* [20]:

Tabla 3. Cantidad de documentos de los subconjuntos equilibrados.

Conjunto	Subconjunto	Limpios	Tóxicos	Total
HSOL	*hate_speech*	1283	1283	2566
	offensive_language	3714	3714	7428
	neither	3714	3714	7428
SLUR	*derogatory*	9793	9793	19586
	homonym	1242	1242	2482
	appropiative	151	151	302
	noise	63	63	126
	non_dorogatory/non_homonym	9793	9793	19586

- *Logistic regression* (LR): este modelo se encuentra definido como *LogisticRegression*() y se basa en un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de variables independientes o predictoras.
- *K-Nearest Neighbors* (KNN): este modelo se encuentra definido como *KNeighborsClassifier*() y es un algoritmo usado como método de clasificación de objetos o elementos que se basa en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos, dónde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación.
- *Bernoulli Naive Bayes* (BNB): este modelo se encuentra definido como *BernoulliNB*() e implementa los algoritmos de entrenamiento y clasificación de Bayes ingenuo para los datos que se distribuyen de acuerdo con las distribuciones Bernoulli multivariadas; es decir, puede haber múltiples características pero se asume que cada una es una variable de valor binario. Por lo tanto, esta clase requiere que las muestras se representen como vectores de características de valor binario.
- *Multinomial Naive Bayes* (MNB): este modelo se encuentra definido como *MultinomialNB*() e implementa el algoritmo ingenuo de Bayes para datos distribuidos multinomialmente, y es una de las dos variantes clásicas de Bayes ingenuo utilizadas en la clasificación de texto (donde los datos se representan típicamente como recuentos de vectores de palabras, aunque también se sabe que los vectores tf-idf funcionan bien en la práctica).
- *Linear Support Vector Machines* (LSVM): las máquinas de vector soporte (SVM) son un conjunto de métodos de aprendizaje supervisado utilizados para la clasificación, la regresión y la detección de valores atípicos. LSVM se encuentra definido como *LinearSVC*() el cual admite entradas densas y escasas, y el soporte multi-clase se gestiona de acuerdo con un esquema de uno contra el reposo.
- *Random Forest* (RF): Un bosque aleatorio es un meta-estimador que ajusta varios clasificadores de árboles de decisión en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión

predictiva y controlar el sobreajuste. Este modelo se encuentra definido como *RandomForestClassifier*().

Cada uno de los modelos será entrenado con cada uno de los subconjuntos de cada conjunto de datos. Los parámetros que se tomarán en consideración serán: (1) Para convertir el texto a vectores se utilizará el vectorizador TF-IDF considerando unigramas y bi-gramas, y una frecuencia de aparición mínima de 3 veces. (2) Los modelos serán entrenados utilizando la validación cruzada también conocida como *K-Folds cross-validation* con $k = 7$. (3) Para la etapa del ensamble de los modelos, se elegirá el mejor clasificador binario para cada clase utilizando la métrica *precision* como el indicador de mejor desempeño.

Aprendizaje profundo. Por otro lado, realizaremos el ajuste fino de dos modelos pre-entrenados. El ajuste fino esta relacionado con el término de aprendizaje por transferencia, el cual se produce cuando utilizamos el conocimiento que se obtuvo al resolver un problema y lo aplicamos a un problema nuevo pero relacionado. Los modelos pre-entrenados que se utilizarán en este trabajo serán:

- BERT [9]: es un modelo que está diseñado para preentrenar representaciones bidireccionales profundas a partir de texto no etiquetado acondicionando conjuntamente el contexto izquierdo y derecho en todas las capas. Como resultado, se puede ajustar con solo una capa de salida adicional para crear modelos para una amplia gama de tareas, como la respuesta a preguntas y la inferencia del lenguaje, sin modificaciones sustanciales de la arquitectura específicas de la tarea.
- RoBERTa [18]: se basa en la estrategia de enmascaramiento lingüístico de BERT [9], en la que el sistema aprende a predecir secciones de texto ocultas intencionalmente dentro de ejemplos de lenguaje que de otro modo no serían anotados. Modifica los hiperparámetros clave en BERT [9], incluida la eliminación del objetivo de preentrenamiento de la próxima frase de BERT [9] y la formación con minilotes y tasas de aprendizaje mucho más grandes.

Cada uno de estos modelos será entrenado con los conjuntos de datos completos, y los parámetros que se tomarán en consideración serán los siguientes: (1) El tamaño de vocabulario de cada conjunto de datos. (2) El número de épocas serán 5, con una detención anticipada de 2. (3) Habrá 500 pasos de calentamiento o *warmup steps*.

3.6. Ensamble de modelos

El ensamble de modelos solo será aplicado para los modelos de aprendizaje automático y se realizará de la siguiente manera: una vez que todos los modelos hayan sido entrenados con cada uno de los subconjuntos, se elegirá el mejor modelo por cada subconjunto, dicha selección se realizará tomando

en cuenta el mejor desempeño sobre la métrica *precision* de los resultados de la validación cruzada. Una vez los mejores modelos sean elegidos, obtendremos una probabilidad de pertenencia de cada documento de los datos de prueba con cada uno de los modelos. Finalmente realizaremos una competencia de probabilidades de todos los modelos, y la mayor probabilidad será la que obtenga la pertenencia para cada documento.

3.7. Novedad científica

Como se ha descrito a través de la sección 3 se propone una distinta estrategia para el preprocesamiento y equilibrio de los datos, creando así un subconjunto de datos para cada una de las etiquetas. De igual manera, en la subsección 3.6 se describe de una manera más detallada la técnica de ensamble de modelos que se utilizará con los modelos de aprendizaje automático.

3.8. Evaluación

La hipótesis del trabajo se evaluará conforme las métricas que se manejan en la sección 2, siendo estas: $precision = TP/(TP+FP)$, $recall = TP/(TP+FN)$ y $F1\ Score = (2*precision*recall)/(precision+recall)$, donde TP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos para las predicciones de cada una de los subconjuntos. Una vez los resultados sean obtenidos, se comparará contra el estado del arte y se discutirá si los experimentos realizados en el presente trabajo logra un mejor desempeño, se mantiene en un margen similar o si empeora.

4. Resultados

El objetivo principal de la experimentación es crear un ensamble de modelos a partir de los mejores modelos de aprendizaje automático que se obtengan de la validación cruzada, con dicho ensamble se realizará una competencia con la finalidad de obtener una etiqueta predicha por documento. Posteriormente, los resultados obtenidos serán comparados contra los resultados de los los ajustes finos, y en el caso del conjunto de datos HSOL [8] se comparará contra el estado del arte.

Con la finalidad de que los experimentos sean replicables, las características de los modelos y de la experimentación en general se describen a continuación: (1) Para la partición de datos de entrenamiento y prueba, y los modelos que aceptan un estado aleatorio se asignó la semilla 42. (2) En el caso del modelo KNN, se asignó 2 para el número de vecinos. Para RF se asignó en número de estimadores en 100. Por último, para LR, *LinearSVC*, BNB y MNB no se asignó ningún parámetro adicional.

Esta sección se encuentra distribuida de la siguiente manera: se dividirá en subsecciones para cada conjunto de datos con el que se experimentó, en cada una de estas subsecciones se encuentran los mejores resultados de validación y los resultados sobre los conjuntos de prueba.

Tabla 4. Mejores resultados de validación para los subconjuntos de HSOL [8].

Subconjunto	Modelo	Resultados		
		F1-Score	*Precision*	*Recall*
hate_speech	RF	0.88	0.94	0.84
offensive_language	LR	0.94	0.97	0.91
neither	*LinearSVC*	0.95	0.93	0.96

Tabla 5. Resultados para el conjunto de pruebas de los modelos para HSOL [8].

Modelo	Subconjunto	Resultados		
		F1-Score	*Precision*	*Recall*
Modelo ensamblado	*hate_speech*	0.32	0.31	0.32
	offensive_language	0.92	0.95	0.90
	neither	0.83	0.75	0.92
BERT	*hate_speech*	0.45	0.59	0.37
	offensive_language	0.94	0.92	0.97
	neither	0.88	0.92	0.85
RoBERTa	*hate_speech*	0.39	0.60	0.29
	offensive_language	0.94	0.92	0.97
	neither	0.87	0.92	0.83

4.1. Conjunto de datos HSOL [8]

La tabla 4 muestra el mejor modelo de aprendizaje automático para cada uno de los subconjuntos en la etapa de validación cruzada. Con los resultados obtenidos, nuestro modelo a ensamblar estará compuesto por los modelos RF, LR y *LinearSVC*. Por otro lado, la tabla 5 muestra los resultados obtenidos sobre el conjunto de pruebas para cada una de las etiquetas, en este punto se puede observar que el modelo ensamblado obtiene mejores resultados en la validación.

4.2. Conjunto de datos SLUR [15]

Similar a los resultados presentados en la subsección 4.1, la tabla 6 muestra los resultados obtenidos con los datos de entrenamiento para todos los subconjuntos que nos ofrece este conjunto de datos. En esta ocasión, nuestro modelo a ensamblar consistirá de los siguientes modelos: MNB, BNB, KNN, MNB y LR. Mientras que en la tabla 7 se observan los resultados obtenidos con los conjuntos de pruebas de todos los modelos propuestos.

5. Análisis y discusión

De manera general hay que mencionar que a pesar de que se realizó un equilibrio de los datos al realizar los subconjuntos para el modelo propuesto

Tabla 6. Mejores resultados de validación para los subconjuntos de SLUR [15].

Subconjunto	Modelo	Resultados		
		F1-Score	*Precision*	*Recall*
derogatory	MNB	0.81	0.86	0.78
homonym	BNB	0.97	0.99	0.95
appropiative	KNN	0.63	0.82	0.53
noise	MNB	0.85	0.94	0.79
non_derogatory/non_homonym	LR	0.85	0.86	0.83

Tabla 7. Resultados para el conjunto de pruebas de los modelos para SLUR [15].

Modelo	Subconjunto	Resultados		
		F1-Score	*Precision*	*Recall*
Modelo ensamblado	*derogatory*	0.57	0.94	0.41
	homonyn	0.82	0.83	0.81
	appropiative	0.03	0.02	0.54
	noise	0.12	0.07	0.50
	non_derogatory/non_homonym	0.69	0.62	0.79
BERT	*derogatory*	0.91	0.90	0.91
	homonyn	0.92	0 .91	0.94
	appropiative	0.07	0	0.04
	noise	0.18	0	0.10
	non_derogatory/non_homonym	0.83	0.83	0.83
RoBERTa	*derogatory*	0.91	0.90	0.91
	homonyn	0.92	0.91	0.94
	appropiative	0.07	0	0.04
	noise	0.18	0	0.10
	non_derogatory/non_homonym	0.83	0.83	0.83

en este trabajo, los conjuntos de datos originales muestran un gran desbalance de documentos entre clases, lo cual supone un gran problema debido a que los clasificadores no tienen suficiente información para ajustar sus parámetros, y de esta manera realizar una clasificación adecuada. Dicho fenómeno se puede observar en ambos conjuntos de datos en las tablas 5 y 7.

En cuanto a los resultados obtenidos, para el conjunto de prueba de HSOL [8] de los ajustes finos de BERT [9] y RoBERTa [18] obtienen un resultado muy similar ya que ambos obtienen 0.81 en *precision* de manera global, 0.73 y 0.70 para *recall* y un *F1 Score* de 0.73 y 0.76 respectivamente. Mientras que el modelo propuesto en este trabajo obtuvo una precisión global de 0.67, para *recall* 0.71 y un *F1 Score* de 0.69. Con estos resultados, no se logró obtener una mejora

contra el estado del arte, los cuales demuestran resultados globales de 0.91 para la métrica *precision* y 0.90 para *recall* y *F1 Score*.

Por otra parte, los resultados sobre el conjunto de pruebas del conjunto de datos SLUR [15] obtenemos un resultado global en la métrica *precision* de 0.53 para ambos ajustes finos, 0.56 para *recall* y 0.58 en la métrica *F1 Score*. Y para el modelo propuesto se obtuvo un resultado global de 0.50 en *precision*, 0.61 para *recall* y 0.45 para *F1 Score*.

6. Conclusiones

A pesar de que no se obtuvieron los resultados esperados, el desempeño del ensamble de modelos propuesto no se aleja de los resultados obtenidos en los ajustes finos, ya que pudimos observar que para algunas etiquetas obtuvimos mejores resultados en métricas donde los ajustes finos no lo hicieron. Para esto hay que tener en consideración que los modelos de aprendizaje automático se tomaron en su forma más básica y que aún hay mucho que experimentar con distintos parámetros que dichos modelos puedan tomar. Por estas razones, se seguirá trabajando sobre este modelado mejorando algunos aspectos con la finalidad de competir con modelos basados en *transformers*.

7. Trabajo futuro

A corto plazo, se buscará mejorar los modelos ajustando sus parámetros, utilizando etiquetados gramaticales POST (*part-of-speech tagging*) e identificando las posibles pérdidas de información en los vocabularios. A la par se se buscará implementar una solución para aquellos conjuntos de datos para los que ciertas clasificaciones no se tengan datos lo suficientemente grandes como para realizar una buena clasificación. Más adelante, se probará la implementación del ensamble de modelos propuesto en este trabajo con conjuntos de datos en otros idiomas como el español, así como con conjuntos de datos más grandes.

Referencias

1. Plaza-del Arco, F. M., Montejo-Ráez, A., Ureña-López, L. A., Martín-Valdivia, M.-T.: OffendES: A new corpus in Spanish for offensive language research. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021). pp. 1096–1108. INCOMA Ltd., Held Online (Sep 2021), https://aclanthology.org/2021.ranlp-1.123
2. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760. WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2017) doi: 10.1145/3041021.3054223 , https://doi.org/10.1145/3041021.3054223

3. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text. CoRR, vol. abs/1903.10676 (2019)
4. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 71–79. KDD '18, Asssociation for Computing Machinery, New York, NY, USA (2018) doi: 10.1145/3219819.3219861 , https://doi.org/10.1145/3219819.3219861
5. Burnap, P., Williams, M. L.: Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, vol. 5, no. 1, pp. 11 (2016) doi: 10.1140/epjds/s13688-016-0072-6
6. Carmona, M. A., Guzmán-Falcón, E., Montes, M., Escalante, H. J., Villaseñor-Pineda, L., Reyes-Meza, V., Rico-Sulayes, A.: Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) (08 2018)
7. Chronopoulou, A., Baziotis, C., Potamianos, A.: An embarrassingly simple approach for transfer learning from pretrained language models. CoRR, vol. abs/1902.10547 (2019)
8. Davidson, T., Warmsley, D., Macy, M. W., Weber, I.: Automated hate speech detection and the problem of offensive language. In: ICWSM (2017)
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, vol. abs/1810.04805 (2018)
10. Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale crowdsourcing and characterization of twitter abusive behavior. CoRR, vol. abs/1802.00393 (2018)
11. Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., Plagianakos, V. P.: Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence. SETN '18, Association for Computing Machinery, New York, NY, USA (2018) doi: 10.1145/3200947.3208069 , https://doi.org/10.1145/3200947.3208069
12. Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N. A.: Don't stop pretraining: Adapt language models to domains and tasks. CoRR, vol. abs/2004.10964 (2020)
13. Hinduja, S., Patchin, J.: Bullying, cyberbullying, and suicide. Archives of suicide research : official journal of the International Academy for Suicide Research, vol. 14, pp. 206–21 (07 2010) doi: 10.1080/13811118.2010.494133
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, vol. 9, pp. 1735–80 (12 1997) doi: 10.1162/neco.1997.9.8.1735
15. Kurrek, J., Saleem, H. M., Ruths, D.: Towards a comprehensive taxonomy and large-scale annotated corpus for online slur usage. In: Proceedings of the Fourth Workshop on Online Abuse and Harms. pp. 138–149. Association for Computational Linguistics, Online (Nov 2020) doi: 10.18653/v1/2020.alw-1.17 , https://aclanthology.org/2020.alw-1.17
16. Lample, G., Conneau, A.: Cross-lingual language model pretraining. CoRR, vol. abs/1901.07291 (2019)
17. Liu, S., Forss, T.: New classification models for detecting hate and violence web content. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). vol. 01, pp. 487–495 (2015)

18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. CoRR, vol. abs/1907.11692 (2019)
19. Park, J. H., Fung, P.: One-step and two-step classification for abusive language detection on Twitter. CoRR, vol. abs/1706.01206 (2017)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Édouard Duchesnay: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830 (2011)
21. Suzuki, K., Asaga, R., Sourander, A., Hoven, C., Mandell, D.: Cyberbullying and adolescent mental health. International journal of adolescent medicine and health, vol. 24, pp. 27–35 (08 2012) doi: 10.1515/ijamh.2012.005
22. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1415–1420. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019) doi: 10.18653/v1/N19-1144 , https://aclanthology.org/N19-1144

ISSN 1870-4069

Design and Development of an IoT-based System for Truck Load Tracking and Monitoring

Omar Otoniel Flores-Cortez[1], Bruno Alberto Gonzales Crespin[2]

[1] Universidad Tecnologica de El Salvador, El Salvador

[2] SmartMetrics, San Salvador, El Salvador

emailomar.flores@utec.edu.sv, bagonzalez.sv@gmail.com

Abstract. Freight transport of goods and raw materials is a main part of the supply chain in the commercial exchange between nations or cities. The control and monitoring of this activity are vital for an efficient economic flow and more importantly without losing money. Most of the problems that generate economic losses occur in cargo freight by land (cargo trucks). Losses due to changes in the weight of the payload to be transported or fuel/time losses due to capricious changes by the driver on the scheduled route. This work aims to demonstrate the use of Internet of Things (IoT) techniques to propose a prototype of a telemetry system to monitor in real-time the payload weight and location of a cargo truck, and become a technological tool that supports the tasks of monitoring and control of the use of cargo trucks, and together with other logistics measures, leads to minimizing economic losses. The development of this project was based on the IoT architecture reference model: an ATmega32u4 microcontroller was used together with a SIM808 GSM and GPS module as the main component of the IoT Node. In addition, Amazon Web Services (AWS) tools were used as an IoT web platform and cloud data storage. The main result was a prototype of a telemetry system to track a cargo truck via the web, the weight and position data are accessible from any device with internet access through a website. Preliminary field tests have been successful and have shown the proposed system to be an efficient and low-cost option.

Keywords: IoT, sensors, GPS, GSM, microcontroller.

1 Introduction

In Latin America, the exchange of merchandise and raw materials is carried out mainly through land transport. Its main advantage is the use of universal road infrastructure and its relatively low cost, is widely used for distances less than 1,000 km and loads less than 44 tons. Thus, each region or country has a large fleet of cargo trucks and its own legislation for this important economic activity. However, despite its wide use, it is in the transport of goods where there

are some problems that lead to economic losses, these due to multiple factors [31, 37, 29]. One problem is the non-control of the weight of the payload in the trucks, from the time it is loaded at the dock of origin until it is delivered to the customer. This creates inconsistencies between what is shipped and what is delivered, resulting in a loss of money for the carrier. It is estimated that these weight discrepancies are due to factors such as robberies on the highways, theft by drivers or the lack of technical standards in the scales of the weighing points along the route. Overloading in road freight transport is a common problem that occurs all over the world. Of the negative effects generated by the overloading of trucks, perhaps the one with the greatest impact is the deterioration of the roads, reducing their useful life and, in turn, higher maintenance costs. Other negative effects are the increase in road accidents, the increase in emissions, fines, and longer transport times [9, 11, 14]. Another problem is the disproportionate expense of time and fuel during truck trips, due to the negligence of drivers in changing or modifying the established route, which generates delays in delivery times that produce monetary losses [42, 5, 29].

Therefore, control and monitoring of trucks along the distribution route has become a necessity, specifically, real-time monitoring of variables such as weight of the load and the geographical position of the truck, in the order to take action to avoid problems due to changes in route or changes in the weight of the truck [36, 17, 28].

Previous works have been related to the development of positional tracking systems for vehicles [2, 30, 16]. But these have focused on the use of high-cost technological tools [23, 43, 1, 22] or are not connected to a website in real-time. The use of GPS technologies together with development boards for microcontrollers such as Arduino is very common in previous works [41, 7, 27, 2]. Other works have focused on developing devices to monitor the weight of cargo trucks [35, 21, 32] however, they also focus on technologies with a high budget or that do not report in real-time [15, 25, 33]. Some of these previous works only focus on monitoring one variable, either weight or location of the cargo truck. In most of the developments, analog type weight sensors have been used, these are based on deflection of shock absorber supports of the truck suspension, converting this angle of deflection into an analog voltage [33, 44, 46] , however also digital weight sensors have been used less in similar systems [38, 33].

The architecture of an Internet of Things (IoT) system is defined by two main blocks: Sensor Nodes and the Internet Platform. The sensor nodes are the telemetry device equipped with sensor elements that take readings of different behavioral variables of the equipment to be monitored. The Internet platform, also called the cloud, is where the data collected by the Node will be stored, in addition to its visualization through web dashboards [3, 24]. The link between these Nodes and the Platform can be implemented using radio technologies such as WiFi, Bluetooth, GSM/GPRS, LoRa, among others [10, 18, 26, 19, 40]. The Atmega and ESP microcontrollers are the most widely used option in the sensor node processor implementation [8, 32, 6, 34, 45]. For the Internet platform,

most used options are Amazon Web Services, Google IoT Core, Tingspeak and Ubidots [45, 39, 20, 4].

This work proposes a low-cost real-time telematics system based on IoT Technologies. The IoT stations are equipped with sensors that can take a reading of the weight and location variables of a cargo truck and send this data over a GSM/GPRS cellular link to the Internet. IoT platform for storage and web deployment accessible through any device connected to the Internet so that personnel can monitor and control possible anomalous situations on the transport route. This paper is structured as follows: Section II summarizes the development of the IoT system prototype. Section III presents the experimental results and a discussion of the proposal, and Section IV concludes and presents some final comments and ideas to be addressed in future work.

2 IoT System to Monitor Cargo Truck Location and Weight

This work aims to demonstrate the use of IoT techniques to propose a prototype of a telemetry system to monitor in real-time the payload weight and location of a cargo truck. The methodological development of this proposed system was based on the IoT Architectural Reference Model [3].

2.1 Purpose & Requirements Specification

Purpose: Automated monitoring of the weight and position of a cargo truck with a GSM cellular link to report in real-time through a web dashboard. Behavior: an electronic station with sensors capable of taking measurements of weight (Tons) and position (GPS) of the truck, and a central digital controller programmed to perform periodic sensor readings and send the collected data via GSM/GPRS cellular link to an Internet platform. Management: the system can be monitored through the Internet and management of programming and configuration of the sensor station can be done locally through a USB port provided in the station itself. Data Analysis: The data collected by the sensor is processed at the station itself and then sent in formatted payload values to the "cloud" service platform. Implementation of applications: the station control software or firmware remains inside the flash memory of the microcontroller and is encoded in C programming language. An IoT platform with web visualization panels is used to monitor the data produced by the node. Security: the system must have user authentication and a JSON protocol with token authentication to receive data payloads from the station to the platform. Access to the web data dashboard will be publicly accessible.

2.2 Process Specification

A single case of operation in a repetitive cycle is defined through the firmware in the digital controller: when the system boots, it executes actions to configure

Algorithm 1: IoT Node Station Process Specification

Result: Periodically t read GPS and weight sensor inside cargo truck then send it to IoT platform via GSM/GPRS cellular link.

```
1  Setup microcontroller hardware;
2  Setup GSM/GRPS hardware;
3  Define t ;                                   // minutes between read/send
4  while True do
5  |  read digital GPS sensor ;                 // long, lat, timestamp
6  |  read analog weight sensor output;
7  |  calculate the weight of the truck ;       // w
8  |  format JSON payload with lon, lat, w, time data ;
9  |  connect to GSM network;
10 |  activate GPRS data use;
11 |  make a HTTP POST into web storage server;
12 |  wait to server reply;
13 |  if replay == 200 then
14 |  |  turn off GRPS;
15 |  |  wait for t;
16 |  else
17 |  |  try again without delay t ;             // retry POST
18 |  end
19 end
```

Fig. 1. Single case algorithm for electronic station process.

the internal and external hardware of the microcontroller, then reads the weight and position sensors, formats them in a loads JSON and finally sends them to the IoT platform through a GSM/GPRS cellular network, this whole process is periodic (see Fig. 1).

2.3 Domain Model Specification

Physical Entity: the cargo truck whose weight and current global position will be read. Virtual Entity: represents a physical entity in the digital world, so only one is defined for the cargo truck. Device: programmable digital controller with GPS position and weight sensors, with GSM cellular network transceiver. Resource: firmware running on the device and a configuration script running on the IoT cloud. Service: the station service runs native on the device.

2.4 Functional View Specification

A functional view defines functional groups (FG) for the different functions of the IoT system. Each functional group has functions to interact with instances

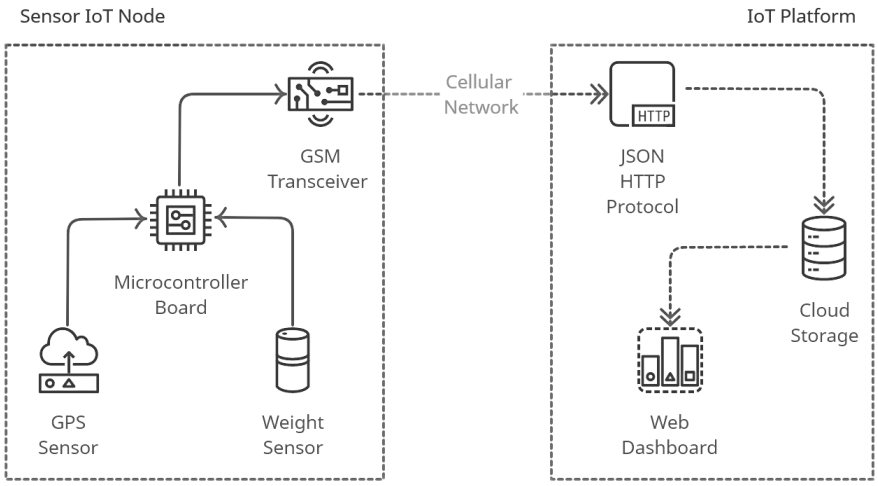

Fig. 2. Overview of functional blocks architecture of the proposed IoT system.

defined in the domain model or with information related to them. Device FG: includes the microcontroller, the GSM transceiver, weight and GPS position sensors. Communications FG: the protocols used are 802.11 link layer, IPv4 network layer, TCP transport layer, HTTP application layer and JSON protocol to send data payload to the IoT platform. Services FG: there is only one service running inside the IoT station control service. Management FG: performed by firmware resource inside the microcontroller. Security FG: the security mechanism is an authentication user credential for the IoT cloud configuration. Application FG: web interface to monitor values produced by the IoT node is in the "cloud" as an internet page.

2.5 Operational View Specification

Options for deployment and operation of the IoT system are defined. IoT node station: mains components are a microcontroller, a GSM network transceiver for internet access, a sensor for weight, a sensor for GPS position. Communication API: Amazon Web Services API. Communication protocols: 802.11, IPV4 / 6, TCP and HTTP. Services: controller service hosted on the device written on C programming language and running as a native service. Applications: Web and database Application – AWS web toolbox. Administration: device – Arduino IDE for electronics station and AWS for cloud applications (see Fig.2).

2.6 Device & Component Integration

Components for the IoT Node: an ATmega32u4 microcontroller is used as CPU, SIM808 chip is used as a transceiver for the GSM/GPRS cell network, which

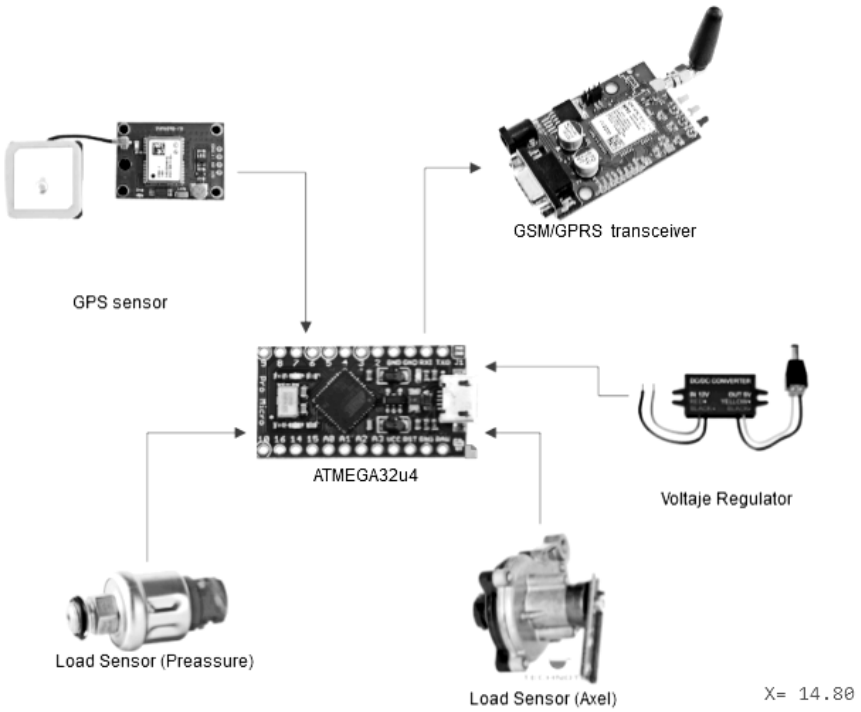

Fig. 3. Electronics components integration for sensor station of proposed IoT system.

also includes a GPS receiver sensor in the same encapsulation. As a weight sensor, GNOM DP sensor with analog output is used, which is placed on the suspension axle of the truck (see Fig. 3). Also in a second prototype, the weight sensor with analog output GNOM DDE is used, which detects changes in the pressure of the truck's damping hoses. As a development hardware platform SIM808 GSM/GPRS/GPS IoT Board from the manufacturer DFRobot is used, that includes the ATmega microcontroller along with the SIM808 in the same board [13].

2.7 Application Development

From the point of view of the software applications developed to run the IoT system 1) IoT node firmware: written in C programming language, the program follows a single loop structure and specific tasks that are repeated cyclically in a period of configurable time (see Fig.1). 2) IoT platform software configuration: "Cloud" service script: developed in JavaScript language hosted in the Amazon Web Services (AWS) cloud. The JavaScript telemetry protocol Object Notation (JSON) is used to send and receive data between the sensor's IoT node and the IoT platform. AWS services were selected for their low cost, high reliability, and

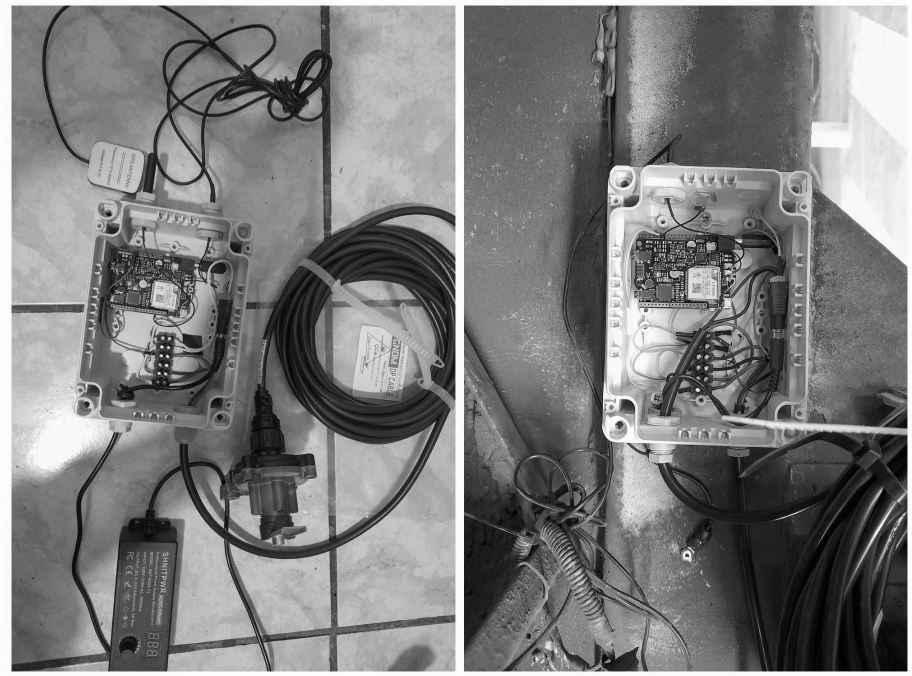

Fig. 4. Assembled electronic prototype for IoT Node Station.

availability versus other similar services. In addition to having a relatively short learning curve. 3) Web Dashboard – Developed using AWS hosting services, using a web toolbox to configure the website with data tables and a graphic dashboard to display the data generated by the sensors.

3 Results and Discussion

The main result of this work was a prototype of an IoT System to monitor the weight and location of a cargo truck in real-time.

3.1 IoT monitoring station

IoT Sensor Node: station with electronic sensors that allow taking measurements of weight and GPS position of a cargo truck and sending them to the IoT platform in the cloud (see Fig.4).

It is a design that takes into account the needs of the conditions of the Latin American region, based on state-of-the-art electronic components that are affordable, efficient and available in the local market. The hardware station design allows for more sensors to be added to the station to increase the variables

Fig. 5. Tests for IoT Node Station prototype attached to a cargo truck.

to be measured. Two detected magnitude values are reported to the website by the station every 10 minutes or can be configured in the microcontroller firmware.

The physical installation of the station is simple, it can be embedded in the structure of a truck. The technical requirements for the installation site are: a 12 VDC power supply near the truck battery and Cellular Network coverage, the station is configured for Internet access through a GPRS link and uses a 2G cellular network (see Fig.5).

The start-up only requires defining via firmware, the network access credentials and a SIM card with an active data plan. In the field tests, a waiting time between shipments to the IoT platform of 5 to 10 minutes was configured, this is changeable from the firmware of the station.

Among the electrical characteristics of the station prototype we have: Operating voltage: 12 VDC @ 0.4 W max. Operating temperature: +60 °C

max. Measurement operation: Weight range from 1 to 10 tons. GPS location Horizontal position accuracy: ¡ 2.5m CEP. Communication performance: Link: GSM/GPRS Quad-band 2G Network. GPRS connectivity: 85.6kbps max and standard SIM Card.

3.2 Web Platform and Field Test

As a field test, the designed system was implemented in the fleet of the company CORPORIN S.A de C.V, which focuses on cargo transportation services within the territory of El Salvador and to some cities in Guatemala and Honduras [12]. To keep track of the data collected, the user can access the website with the URL through any device with Internet access: https://bit.ly/3lcSe5X. This website includes tables and dashboards to view the history of weight, longitude and latitude values reported by the station installed inside the cargo truck (see Fig.6). System performance so far has been satisfactory. The telemetry link has not suffered losses and has remained stable. Several tests were carried out with different IoT Nodes located at different points within the structure of the truck, one of the best link results was obtained with the station located behind the truck cabin and with an open sky view. Looking at the data collected during the field test period, no out-of-the-average changes were observed in the data for the truck in question. The weight and GPS position data have been consistent with the preset route in the company's headquarters.

4 Conclusion and Future Work

Development of an IoT system to monitor in real-time the weight and positioning of a truck along its route is a fundamental step for the study of behavior, impacts and actions on possible data anomalies on the route and weight of the truck. This work demonstrates the use of Internet of Thing (IoT) techniques to design and build a prototype of a telemetry system to monitor in real time the weight of the payload and the location of a cargo truck, and become a tool technology that supports the monitoring and control tasks of the use of cargo trucks, and together with other logistics measures, leads to minimizing economic losses due to inconsistencies in the weight of the cargo. The proposed system was developed using state-of-the-art techniques in electronics, programming and the Internet of Things, which allowed the production of low-cost equipment that works according to the expected requirements. Tools like Atmega Microcontroller together with the C programming language enable efficient IoT development prototypes at low cost, with short development times and high performance. In addition, the use of AWS toolbox has enabled quick and easy monitoring of the web development platform and site to data from any device and in real-time.The contribution of this work was to show new and innovative techniques for the use of hardware and software components in the implementation of IoT Systems. In addition to being an ad-hoc application for need and context of cargo transportation in El Salvador, where aspects such as low cost and customization

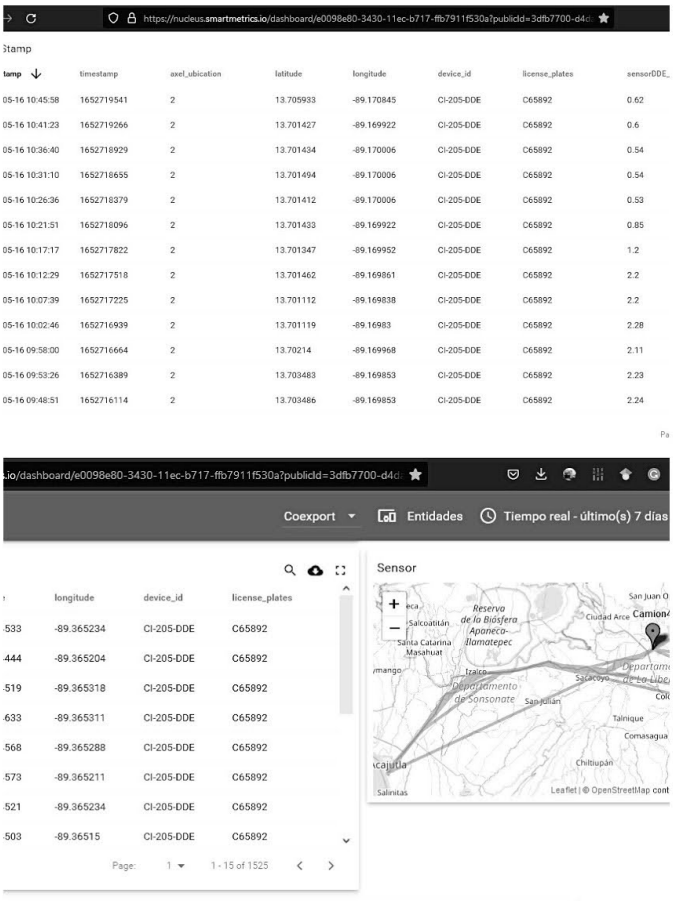

Fig. 6. Web site tables and dashboards for collected truck data coming from the sensor station.

are valuable aspects for innovative technological proposals. These can be applied in new developments, allowing rapid and efficient prototyping. To be done, this research is tasked with developing more stations by adding different sensors to capture more variables about truck performance and establish more field test validation. Implement a more robust cloud platform, with powerful dashboards and tables with ready-to-read data. Additionally, we seek to implement a monitoring network through radio frequency links and analyze big data or forecasts based on the data produced by the stations. The result of this work can be used in the development of new lines of applied research, in areas such as land or aquifer analysis, monitoring in agricultural and livestock fields, sports performance analysis, etc.

References

1. Alrifaie, M. F., Harum, N., Othman, M. F. I., Roslan, I., Shyaa, M. A.: Vehicle detection and tracking system iot based: a review. Int. Res. J. Eng. Technol, pp. 1237–1241 (2018)
2. Arguijo, J. E. M., León, E. G., Arellano, C. C., García, F. R. M.: Propuesta de sistema de gestión para optimización de redes de transporte público. Res. Comput. Sci., vol. 148, no. 10, pp. 235–245 (2019)
3. Bahga, A., Madisetti, V.: Internet of Things: A hands-on approach. Vpt (2014)
4. Balakrishna, S., Thirumaran, M.: Programming paradigms for iot applications: an exploratory study. Handbook of IoT and Big Data, pp. 23–57 (2019)
5. Beetrack: 6 problemas de distribución logística de productos [última milla] (Jul 2019), https://www.beetrack.com/es/blog/logistica-de-distribucion/
6. Bento, A. C.: Iot: Nodemcu 12e x arduino uno, results of an experimental and comparative survey. International Journal, vol. 6, no. 1 (2018)
7. Cadena, A. C. P., Matamoros, O. M., Pérez, D. A. P., Escobar, J. J. M.: Software de estación terrena para cohetes hidropropulsados. Res. Comput. Sci., vol. 148, no. 10, pp. 305–322 (2019)
8. Calixto-Rodriguez, M., Valdez Martínez, J. S., Meneses-Arcos, M., Ortega-Cruz, J., Sarmiento-Bustos, E., Reyes-Mayer, A., González-Castañeda, M., Domínguez García, R. O.: Design and development of software for the silar control process using a low-cost embedded system. Processes, vol. 9, no. 6, pp. 967 (2021)
9. Carbajal, J.: Vmt inspecciona peso de los camiones de carga y a la vez realiza exámenes antidoping a motoristas (Nov 2021), https://www.laprensagrafica.com/elsalvador/VMT-inspecciona-peso-de-los- camiones-de-carga-y-a-la-vez-realiza-examenes-antidoping-a-motoristas-20211124- 0062.html
10. Chanchí G, G. E., Ospina A, M. A., Campo M, W. Y., et al.: Iot architecture for monitoring variables of interest in indoor plants. Computación y Sistemas, vol. 25, no. 4 (2021)
11. Cortes, R.: Soluciones tecnológicas de pesaje inalámbrico en vehículos de carga (Feb 2019), https://blogs.iadb.org/ transporte/es/soluciones-tecnologicas-de-pesaje-inalambrico-en-vehiculos-de-carga/
12. de CV, C. S.: (2021), http://www.corporin.com
13. DFRobot: Sim808withleonardomainboard, https://wiki.dfrobot.com/SIM808-with-Leonardo-mainboard-SKU-DFR0355
14. Espinoza, J.: Inseguridad en el norte de centroamérica afecta al transporte de carga (Mar 2018), https://www.elnuevodiario.com.ni/economia/420764-inseguridad-norte-centroamerica-afecta-transporte/
15. Feng, M. Q., Leung, R. Y.: Application of computer vision for estimation of moving vehicle weight. IEEE Sensors Journal, vol. 21, no. 10, pp. 11588–11597 (2020)
16. Gallego Tercero, L. R., Menchaca Mendez, R., Rivero Angeles, M. E.: Spatio-temporal routing in episodically connected vehicular networks. Computación y Sistemas, vol. 24, no. 4 (2020)
17. Gohin Tay, C. A., Vera Bernuí, K. E.: Mejora del sistema de monitoreo y rastreo vehicular position logic-fermon perú sac (2015)
18. Golondrino, G. E. C., Alarcón, M. A. O., Muñoz, W. Y. C.: Sistema iot para el seguimiento y análisis de la intensidad de luz en plantas de interiores. Res. Comput. Sci., vol. 149, no. 11, pp. 317–327 (2020)

19. Gómez, A. P., Cahuich, A. C. S., Gómez, J. J.: Plataforma de gestión iot mediante técnicas de industria 4.0 para agricultura de precisión. Res. Comput. Sci., vol. 149, no. 11, pp. 303–315 (2020)
20. Hejazi, H., Rajab, H., Cinkler, T., Lengyel, L.: Survey of platforms for massive iot. In: 2018 IEEE International Conference on Future IoT Technologies (Future IoT). pp. 1–8. IEEE (2018)
21. Hernandez, S., Hyun, K.: Fusion of weigh-in-motion and global positioning system data to estimate truck weight distributions at traffic count sites. Journal of Intelligent Transportation Systems, vol. 24, no. 2, pp. 201–215 (2020)
22. Huang, C.-C., Lin, C.-L., Kao, J.-J., Chang, J.-J., Sheu, G.-J.: Vehicle parking guidance for wireless charge using gmr sensors. IEEE Transactions on Vehicular Technology, vol. 67, no. 8, pp. 6882–6894 (2018)
23. Jurado Murillo, F., Quintero Yoshioka, J. S., Varela López, A. D., Salazar-Cabrera, R., Pachón de la Cruz, Á., Madrid Molina, J. M.: Experimental evaluation of lora in transit vehicle tracking service based on intelligent transportation systems and iot. Electronics, vol. 9, no. 11, pp. 1950 (2020)
24. Lv, W., Meng, F., Zhang, C., Lv, Y., Cao, N., Jiang, J.: A general architecture of iot system. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). vol. 1, pp. 659–664. IEEE (2017)
25. Maia, J., Yudi, J.: An iot solution for load monitoring and tracking of garbage-truck fleets. In: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS). vol. 1, pp. 406–410. IEEE (2020)
26. Martínez, A., Onofre, H., Estrada, H., Torres, D., Maquinay, O.: Diseño y desarrollo de una arquitectura iot en contexto con la plataforma fiware. Res. Comput. Sci., vol. 147, no. 8, pp. 95–106 (2018)
27. Medel Juárez, J. d. J., Urbieta Parrazales, R., Garduño Mendieta, V.: Meteorological portable system consulted via wi-fi. Computación y Sistemas, vol. 23, no. 4 (2019)
28. Minero, E.: El uso de tecnología para aumentar la productividad de la carga (Dec 2015), https://www.equipo-minero.com/contenidos/el-uso-de-tecnologia-para-aumentar-la-productividad-de-la-carga/
29. Moldtrans: El transporte de mercancías ha mejorado, pero debe perfeccionarse (Sep 2019), https://www.moldtrans.com/cuales-son-los-problemas-mas-habituales-en-el-transporte-de-mercancias/
30. Molina, Y. A., Ramírez, S. S., Morales, J. G., Reyes, A. M., Sánchez, R. G., García, I. V.: Diseño y desarrollo de un sistema de monitoreo remoto implementando internet de las cosas. Research in Computing Science, vol. 149, pp. 235–247 (2020)
31. Moral, L. A.: Logistica del transporte y distribucion de carga. Ecoe Ediciones (2014)
32. Nayyar, A., Puri, V.: A review of arduino board's, lilypad's & arduino shields. In: 2016 3rd international conference on computing for sustainable global development (INDIACom). pp. 1485–1492. IEEE (2016)
33. Oskoui, E. A., Taylor, T., Ansari, F.: Method and sensor for monitoring weight of trucks in motion based on bridge girder end rotations. Structure and Infrastructure Engineering, vol. 16, no. 3, pp. 481–494 (2020)
34. Polianytsia, A., Starkova, O., Herasymenko, K.: Survey of hardware iot platforms. In: 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). pp. 152–153. IEEE (2016)

35. Putra, S. A., Trilaksono, B. R., Riyansyah, M., Laila, D. S., Harsoyo, A., Kistijantoro, A. I.: Intelligent sensing in multiagent-based wireless sensor network for bridge condition monitoring system. IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5397–5410 (2019)
36. Q: El control satelital de camiones y por qué debes considerarlo (Oct 2020), https://www.ubicalo.com.mx/blog/control-satelital-de-camiones/
37. Quijano, R.: Transporte de carga: Un trabajo de peso: Noticias de el salvador (Nov 2018), https://historico.elsalvador.com/historico/540323/transporte-de-carga-un-trabajo-de-peso.html
38. Radhakrishnan, K., Julien, C., Baranowski, T., O'Hair, M., Lee, G., De Main, A. S., Allen, C., Viswanathan, B., Thomaz, E., Kim, M., et al.: Feasibility of a sensor-controlled digital game for heart failure self-management: Randomized controlled trial. JMIR serious games, vol. 9, no. 4, pp. e29044 (2021)
39. Ray, P. P.: A survey of iot cloud platforms. Future Computing and Informatics Journal, vol. 1, no. 1-2, pp. 35–46 (2016)
40. Saleem, S. I., Zeebaree, S., Zeebaree, D. Q., Abdulazeez, A. M.: Building smart cities applications based on iot technologies: A review. Technology Reports of Kansai University, vol. 62, no. 3, pp. 1083–1092 (2020)
41. San Hlaing, N. N., Naing, M., San Naing, S.: Gps and gsm based vehicle tracking system. International Journal of Trend in Scientific Research and Development (IJTSRD), (2019)
42. Sánchez, R., Cipoletta Tomassian, G.: Identificación de obstáculos al transporte terrestre internacional de cargas en el MERCOSUR. CEPAL (2003)
43. Saritha, B., Bharadwaja, C., Nikhitha, M., Nethra Reddy, C., Arun, K., Ahmed, S. M.: An intelligent anti-theft vehicle locking system using iot. In: ICDSMLA 2020, pp. 1589–1595. Springer (2022)
44. Seo, M. K., Shin, H. Y., Lee, H. Y., Ko, J. I., Tumenjargal, E.: Development of onboard scales to measure the weight of trucks. Journal of Drive and Control, vol. 18, no. 1, pp. 9–16 (2021)
45. Singh, K. J., Kapoor, D. S.: Create your own internet of things: A survey of iot platforms. IEEE Consumer Electronics Magazine, vol. 6, no. 2, pp. 57–68 (2017)
46. Stawska, S., Chmielewski, J., Bacharz, M., Bacharz, K., Nowak, A.: Comparative accuracy analysis of truck weight measurement techniques. Applied Sciences, vol. 11, no. 2, pp. 745 (2021)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación