

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS



CARRERA:

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PREVIO AL GRADO ACADÉMICO DE INGENIERÍA
DE SISTEMAS Y COMPUTACIÓN

TEMA DE INVESTIGACION:

MAPEO SISTEMÁTICO SOBRE LOS MÉTODOS, TÉCNICAS Y TECNOLOGÍAS
ORIENTADAS A LA DETENCIÓN DE NOTICIAS FALSAS

LINEA DE INVESTIGACIÓN:

AUTOMATISMOS Y SISTEMAS INTELIGENTES

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO DE SISTEMA Y COMPUTACIÓN

AUTOR:

ALVAREZ DAZA NATHALY VALERIA

ASESOR:

PICO VALENCIA PABLO (PhD)

ESMERALDAS, 2021

TRIBUNAL DE GRADUACIÓN

Trabajo de tesis aprobado luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de Grado de la PUCESE previo a la obtención del título de INGENIERO DE SISTEMAS Y COMPUTACIÓN.

.....
Ph.D. Pablo Pico Valencia

Tutor de Tesis

.....
Mgt. Gustavo Chango Sailema

Lector 1

.....
Mgt. Manuel Nevárez Toledo

Lector 2

.....
Mgt. Susana Patiño Rosado

Coordinadora de la Escuela de Sistemas y Computación

AUTORÍA

Yo, **NATHALY VALERIA ALVAREZ DAZA** portador de la cédula de identidad No. **0803019066** declaro que los resultados obtenidos en la investigación que presento como informe final, previo a la obtención del título de “Ingeniero de Sistemas y Computación” son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola, exclusiva responsabilidad legal y académica.

.....
Nathaly Valeria Álvarez Daza

C.I. 0803019066

DEDICATORIA

Este trabajo de investigación está dedicado a Dios por la vida, la sabiduría y las condiciones para llevar a cabo mis estudios y poder terminarlos con éxito.

A mi familia, las personas que más amo y adoro en este mundo, a mi madre por su amor, trabajo y sacrificio en todos estos años, gracias a ella he logrado llegar hasta aquí. A mi abuelito, mis tíos y demás familiares, por el cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias.

A mis amigos por apoyarme cuando más las necesito, por extender su mano en momentos difíciles y por el amor brindado cada día.

También a todas las personas que formaron parte de este proceso porque también gracias a ellos este esfuerzo está dando fruto.

AGRADECIMIENTOS

En primera instancia agradezco a Dios por tenernos con salud y permitirme terminar esta etapa tan importante en mi vida.

Gracias a mi madre por motivarme día a día para ser una mejor persona y estar pendiente de mi educación, por guiarme, alentarme y corregirme para que siga adelante con mis metas.

A mi abuelito por su apoyo y llevarme a la universidad y a mis familiares y amigos por demostrarme todo su cariño y por sus palabras de aliento en estos años.

A mi asesor por su apoyo y por cada una de sus correcciones y directrices para realizar correctamente el proyecto.

RESUMEN

Las noticias falsas llegan a ser virales, debido a las redes sociales y esto causa desinformación en los usuarios. La presente investigación fue desarrollada con el objetivo de investigar los métodos, técnicas y herramientas para la detección de noticia falsas que emplean técnicas de Inteligencia artificial (IA). Para llevar a cabo esta investigación se realizó un mapeo sistemático de literatura considerando criterios de inclusión y criterios de exclusión, con estudios publicados en el periodo 2010-2020. El estudio fue desarrollado a partir de una metodología formal usada en ciencias de la Computación. El estudio de tipo mapeo sistemático se realizó siguiendo las etapas propuestas por Peterson de las cuales se plantearon 6 preguntas de investigación y se recuperaron 314 estudios de donde se identificaron los algoritmos de aprendizaje supervisado usados, herramientas y técnicas de procesamiento. Los resultados mostraron que las técnicas de inteligencia artificial (IA) de aprendizaje automático, aprendizaje profundo y minería de datos, fueron ampliamente usadas y existen varios factores que involucra en la detección, esto es, el algoritmo empleado o las técnicas usadas para el procesamiento de texto, lo cual afecta en los porcentajes de exactitud de los modelos, tanto de aprendizaje automático y aprendizaje profundo. Para ambas técnicas, se identificaron modelos entrenados que alcanzaron un 99.99% de precisión para la detección de noticias falsas. Además, la puesta en marcha de sistemas de detección de noticias falsas disponibles en repositorios públicos permitió constatar desde una perspectiva experimental los resultados obtenidos con el mapeo sistemático.

Palabras claves: Inteligencia artificial, noticias falsas, mapeo sistemático.

ABSTRACT

Fake news becomes viral, due to social media and this causes misinformation in users. This research was developed with the aim of investigating the methods, techniques, and tools for the detection of false news that use artificial intelligence (IA) techniques. To carry out this research, a systematic mapping of the literature was carried out considering inclusion criteria and exclusion criteria, with studies published in the 2010-2020 period. The study was developed from a formal methodology used in Computer Science. The systematic mapping type study was carried out following the stages proposed by Peterson, from which 6 research questions were posed and 314 studies were retrieved from which the supervised learning algorithms used, tools and processing techniques were identified. The results showed that machine learning, deep learning, and data mining artificial intelligence (IA) techniques were widely used and there are several factors involved in detection, that is, the algorithm used, or the techniques used to process text, which affects the accuracy percentages of the models, both machine learning and deep learning. For both techniques, trained models were identified that achieved 99.99% accuracy for the detection of fake news. In addition, the implementation of false news detection systems available in public repositories allowed the results obtained with systematic mapping to be verified from an experimental perspective.

Palabras claves: Artificial intelligence, fake news, systematic mapping.

ÍNDICE DE CONTENIDOS

AUTORÍA	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
RESUMEN	v
ABSTRACT	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
GLOSARIO	xi
INTRODUCCIÓN	1
Presentación del problema	1
Planteamiento del problema	1
Justificación	3
Objetivos	4
Objetivo general	4
Objetivos específicos	4
Contribuciones científicas relacionadas	4
2.1. Bases teóricas-científicas	5
2.1.1. Noticias falsas	5
2.1.1.1. Noticias falsas en redes sociales	6
2.1.1.2. Noticias falsas en Facebook	7
2.1.1.3. Noticias falsas en Twitter	8
2.1.1.4. Noticias falsas en otros medios sociales	9
2.1.2. Detección automática de noticia falsas con IA	10
2.1.2.1 Minería de texto	10
2.1.2.2. Aprendizaje automático	14
2.1.2.3. Aprendizaje profundo	22
2.1.2.4. Minería de datos	30
2.2 Antecedentes	32
2.3. Fundamentación legal	34
3. METODOLOGÍA	36
3.1. Delimitación de la investigación	36
3.2. Tipo de investigación	36
3.3. Método de investigación	37

3.4. Población y muestra.....	38
3.5. Variable e indicadores investigados.....	38
3.7. Técnicas de procesamiento y análisis de datos.....	41
3.8. Normas éticas.....	42
4. RESULTADOS	43
4.1 Descripción del estudio de mapeo sistemático	43
4.1.1. Preguntas de investigación.....	43
4.1.2. Definición de estrategia de búsqueda.....	44
4.2. Estudios analizados en el mapeo sistemático	46
4.3. Respuesta a las preguntas de investigación	53
4.4. Evaluación de modelos de clasificación de noticias falsas.....	76
5. DISCUSIÓN.....	84
6. CONCLUSIONES Y RECOMENDACIONES	86
6.1. Conclusiones.....	86
6.2. Recomendaciones	87
REFERENCIAS	88

ÍNDICE DE TABLAS

Tabla 1. Características de las noticias falsas y reales en redes sociales.....	6
Tabla 2. Uso de internet en América del Sur [14]	7
Tabla 3. Principales variables e indicadores sujeto a estudio	40
Tabla 4. Bases de datos científicas y librerías digitales empleadas.....	44
Tabla 5. Cadenas de búsqueda.....	45
Tabla 6. Cantidad de artículos brutos recuperados en las fuentes de información empleadas.	45
Tabla 7. Estudios orientados a detectar noticias falsas.....	48
Tabla 8. Algoritmo de aprendizaje automático para clasificación de noticias falsas. ...	56
Tabla 9. Clasificación de artículos con aprendizaje automático.....	61
Tabla 10. Categoría de pregunta RQ4	68
Tabla 11. Herramientas de software empleadas para desarrollar los clasificadores de noticias falsas.....	70
Tabla 12. Porcentaje de exactitud de los algoritmos de aprendizaje automáticos usados en la detección de noticias falsas.....	71
Tabla 13. Porcentaje de exactitud de los algoritmos de aprendizaje profundo usados en la detección de noticias falsas.....	72
Tabla 14. Comparación de resultados con respecto a exactitud	83

ÍNDICE DE FIGURAS

Figura 1. Proceso de minería de texto [28].....	11
Figura 2. Ejemplo de Regresión lineal simple. Fuente: Elaborado por la autora.	15
Figura 3. La función logística [48]	16
Figura 4. Ejemplo de Naïve Bayes [51]	17
Figura 5. Ejemplo de Jugar Tenis [53]	18
Figura 6. Ejemplo de bosque aleatorio.	18
Figura 7. Ejemplo sobre cómo analizar el algoritmo KNN [56].	19
Figura 8. Interpretación de geometría de SVM [57].	20
Figura 9. Ejemplo de k-means Fuente: Elaborado por la autora	21
Figura 10. Esquema general de una red neuronal [63].	23
Figura 11. Representación de una Neurona [57].	23
Figura 12. Proceso hacia adelante [125].	24
Figura 13. Propagación hacia atrás [125].	25
Figura 14 . Red neuronal recurrente [66].	26
Figura 15. Agrupamiento promedio en una CNN [68].	27
Figura 16. Arquitectura de una CNN [69].	27
Figura 17. Bloque de memoria con única celda [71].	28
Figura 18. Red LSTM [71].	29
Figura 19. Proceso en minería de datos [37].	30
Figura 20. Proceso para seleccionar los estudios sujetos a análisis.	46
Figura 21. Distribución de estudios primarios por año	53
Figura 22. Distribución de estudios primarios por país.....	54
Figura 23. Gráficos de las técnicas usadas.	55
Figura 24. Principales redes sociales desde las que se ha extraído datos para entrenar clasificadores para la detección de noticias falsas.	74
Figura 25. Tamaño de los conjuntos de datos utilizados en las investigaciones	75
Figura 26. Extracción de característica utilizados en los modelos	76
Figura 27. Conjuntos de datos trabajado para las pruebas	78

GLOSARIO

BOW	Bag Of Words Bolsa de palabras
CBOW	Continuos Bag Of Words Modelo de Bolsa de Palabras Continuo
CNN	Convolutional Neural Network Red Neuronal Convolutacional
COIP	Comprehensive Organic Criminal Code Código Orgánico Integral Penal
DT	Decision Trees Árboles de decisión
FNC	Fake News Challenge Desafío de Noticias Falsas
IA	Artificial Intelligence Inteligencia Articial
INEC	National Institute Of Statistics And Censuses Instituto Nacional de Estadística y Censos
KNN	K-Nearest-Neighbor K-vecinos más cercanos
LR	Logistic Regression Regresión Logística
LSTM	Long Short Term Memory Memoria Larga a Corto Plazo
ML	Machine Learning Aprendizaje Automático
NB	Naïve Bayes
NLTK	Natural Language Toolkit Kit de herramientas de lenguaje natural
PCA	Principal Component Analysis Análisis de componente principal
RF	Random Forest Bosques Aleatorios
RNA	Artificial Neural Networks Redes Neuronales Artificiales
RNR	Recurrent Neural Networks Redes Neuronales Recurrentes
RTRL	Real Time Recurrent Learning Aprendizaje Recurrente en Tiempo Real
SVM	Support Vector Machines Máquinas de vectores de soporte
TF	Term Frequency Término de Frecuencia
TF-IDF	Term Frequency – Inverse Document Frequency Frecuencia de Término – Frecuencia Inversa de Documento
XML	Extensible Markup Language Lenguaje de Mercado Extensible

INTRODUCCIÓN

Presentación del problema

En la actualidad las redes sociales se han vuelto muy populares por su fácil acceso y el intercambio de información. Además de permitir la comunicación entre amigos y familiares, éstas permiten informarse de las noticias que suceden en el mundo. Y con esto han proliferado las publicaciones de noticias que no están contrastadas o validadas y aparece lo que son las noticias falsas conocidas como *fake news*, definidas como artículos de noticias que son intencional y pueden engañar a los lectores [1].

Es necesario determinar la credibilidad de las publicaciones en medios sociales, donde hoy en día cualquier ciudadano puede compartir cualquier publicación. En este sentido, adquieren relevancia herramientas tecnológicas que permitan definir la confiabilidad de las noticias. Es así como la inteligencia artificial (IA) ha servido de apoyo con técnicas de aprendizaje automático, aprendizaje por refuerzo y la minería de datos para crear modelos que permiten detectar de manera automática las noticias falsas. Hay múltiples propuestas para la detección de noticias falsas en las redes sociales basado en la IA, algunas están enfocadas en la plataforma de Facebook y Twitter.

Planteamiento del problema

La información transmitida por medio de estas plataformas puede llegar a muchos países siendo una noticia falsa rápidamente difundida. A medida que más personas leen noticias por medio de sus redes sociales, el fenómeno de las noticias falsas es cada vez más común en estos medios de comunicación donde se publican contenidos de interés en varias áreas como: política, deportes, economía, ciencia, entretenimiento, entre otros. Un caso particular a nivel gubernamental fue el de Estados Unidos. En 2016, Allcott y Gentzkow [1] realizaron un análisis de las elecciones presidenciales en la cual se compartió en la red social Facebook 115 noticias falsas por parte del candidato Donald Trump y por la candidata Hillary Clinton 41. Otra área afectada es el mercado financiero donde Carvalho et al. [2] señalaron que un artículo publicado en Internet sobre la compañía matriz de United Airlines, causó que el precio de las acciones de la compañía cayera luego de identificar a la noticia como falsa, el precio de las acciones se recuperó.

Otro caso es de las noticias falsas compartidas a nivel mundial fue la relacionada al dióxido de cloro. Dichas publicaciones afirmaban dicho componente químico constituía

la cura contra el Covid-19 y a causa de esto muchas personas lo compraron y lo consumieron sin saber las consecuencias que podría causarle si se lo suministraba sin supervisión médica [3].

En el contexto de Ecuador, redes sociales como Twitter y Facebook se han convertido en un campo de batalla por parte de las figuras políticas del país [4]. La información falsa es rápidamente difundida y afecta a muchas áreas. Una de ellas es el ámbito de la política. Un caso ocurrido en Ecuador fue durante el mes de octubre de 2019, fecha en que se inició un paro nacional donde hubo protestas tras el anuncio de ciertas medidas económicas por parte del gobierno y con esto circularon noticias falsas por la red social Facebook. Benítez e Hidalgo [5] analizaron los comportamientos y el contenido compartido durante el paro nacional sobre noticias falsas que hace referencia a Jaime Nebot y la segunda a Rafael Correa.

Asimismo, Moreta [6], analizó la influencia de las fake news acontecidas los días del paro nacional en Ecuador, en los estudiantes de la carrera de comunicación social. Las publicaciones falsas que tuvieron impacto para los estudiantes fue que el presidente de la República del Ecuador Lenin Moreno ordenó matar a las personas que se encontraban en las manifestaciones, y el precio del gas se incrementaría debido a que se quitaría el subsidio. Otro caso ocurrido fue en Ecuador en el 2018 durante el proceso electoral de referéndum y consulta popular, se determinaron un total de 19 noticias falsas en la red social Twitter y juntas alcanzaron un total de 1.507 retweets, 1.146 me gusta, y 112 comentarios [4].

En las redes sociales hay muchos sitios web que tienen como objetivo producir noticias falsas siendo estas perjudiciales para toda la sociedad incluyendo empresas y gobiernos. Por ejemplo, las noticias falsas sobre las organizaciones, emitidas por *spam* o usuarios malintencionados, pueden causar daños considerables. Por lo tanto, la detección de noticias falsas se ha convertido en un área de investigación importante en los últimos años [7].

Como se ha descrito en los casos anteriores, la repercusión que una noticia falsa tiene puede ser grave o traer consecuencias negativas para la sociedad. Existe una diversidad de propuestas que aplican diferentes técnicas de IA, pero no se ha establecido una comparativa sobre cuáles de los métodos aplicados provee un mejor resultado en el reconocimiento de noticias falsas. Por ello, se necesita el direccionamiento de una revisión sistemática de la literatura que describa las principales técnicas de IA que han

sido propuestas para así determinar cuál método tuvo la mejor medida de exactitud (*accuracy*) para llevar a cabo la detección de noticias falsas. Lo que se propone es tomar todas las propuestas enfocadas en IA que han aplicado minería de texto para luego compararlas desde una perspectiva cualitativa y cuantitativa. Así, el estudio, documental propuesto ha intentado responder las siguientes interrogantes; (i) ¿cuáles han sido las técnicas de IA aplicadas en la detección de noticias falsas?, (ii) ¿cuáles redes sociales y países han sido más vulnerables en detectar una noticia falsa? y (iii) ¿en qué medida la IA ha coadyuvado para crear mecanismos de detección de noticias falsas en redes sociales?

Justificación

El hecho que existen muchas noticias falsas en las redes puede ocasionar pánico a la sociedad, por esto se busca incorporar mecanismos que permitan detectar las noticias erróneas que son difundidas por estos medios digitales. Lo que se pretende es desarrollar un estudio que compare las propuestas más relevantes que ya han sido elaboradas para implementar las técnicas más efectivas para el reconocimiento de noticias falsas en las redes sociales. Esta información es útil para el área académica o investigadores que quieran ver los diferentes métodos que existen hasta el momento para detectar información errónea en las redes sociales. De esta manera se dará confianza a los lectores y que no se confundan con noticias que no son válidas o contrastadas.

Es importante realizar este estudio debido al gran uso de las redes sociales que han permitido multiplicar las *fake news* y muchas de las personas no son capaces de diferenciar una información verdadera de una errónea y de esta manera con la determinación de las propuestas que detectan noticia falsas de manera más precisa, se ayuda a que los investigadores que deseen investigar en esta línea conozcan las técnicas de IA más efectivas para detectar noticias falsas en formato de texto que se publican en la red.

Es importante señalar que con los resultados obtenidos se podrá conocer los clasificadores automáticos que mayor precisión (*accuracy*) han tenido para la detección de noticias falsas y con esto implementarlo en sistemas de casos reales, de manera que el reconocimiento de noticia falsas sea lo más preciso posible, de acuerdo con los actuales algoritmos de aprendizajes existentes.

Objetivos

Objetivo general

Investigar los principales métodos, técnicas y tecnologías orientadas a detectar noticias falsas en redes sociales, que emplean técnicas de IA, mediante la ejecución de un proceso de revisión sistemática de la literatura.

Objetivos específicos

- a. Determinar las principales características que diferencian una noticia falsa de una real en redes sociales.
- b. Identificar los principales métodos, técnicas y tecnologías de Inteligencia Artificial propuestas para detectar noticias falsas en redes sociales.
- c. Describir los métodos, técnicas y tecnologías identificadas que se han aplicado en la detección de noticias falsas en medios sociales aplicando minería de texto.
- d. Comparar los métodos enfocados en la detección automática de noticias falsas desde un enfoque cualitativo y cuantitativo.

Contribuciones científicas relacionadas

La investigación desarrollada ha dado como fruto dos publicaciones en congresos internacionales. A continuación, se describe cada una de ellas:

- **Álvarez-Daza, Nathaly;** Pico-Valencia, Pablo. APRENDIZAJE AUTOMÁTICO Y APRENDIZAJE PROFUNDO ORIENTADO A LA DETECCIÓN NOTICIAS FALSAS EN LA RED SOCIAL TWITTER. En III Congreso Internacional de Ciencias Informáticas de la Universidad Técnica de Manabí, Fecha: 21 al 23 de octubre de 2020.
- **Álvarez-Daza, Nathaly;** Pico-Valencia, Pablo; Holgado-Terriza, Juan A. APRENDIZAJE MÁQUINA Y PROFUNDO PARA LA DETECCIÓN DE NOTICIAS FALSAS EN REDES SOCIALES. UNA REVISIÓN. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao. 41, pp. 632-645, 2021.

2. MARCO TEÓRICO

2.1. Bases teóricas-científicas

A continuación, se presentan las definiciones que permiten conocer y comprender de forma técnica, las bases teóricas-científicas relacionadas con la detección de noticias falsas en redes sociales, empleando las principales técnicas basadas en aprendizaje que son actualmente usadas en la Inteligencia Artificial.

2.1.1. Noticias falsas

Una definición ampliamente extendida en relación a las noticias falsas es aquella que las considera como artículos que son intencional y pueden engañar a los lectores [1], [8]. Este tipo de noticias, también conocidas como *fake news* por sus términos en inglés, son contenidos de desinformación que son difundidos por medios de las redes sociales con el fin de causar pánico o coraje a los internautas. En todo caso, las noticias falsas constituyen “información fabricada que imita el contenido de los medios de comunicación en forma; pero no en el proceso organizacional o intención” [9].

Los usuarios de Internet tienden a considerar como noticia real todas aquellas publicaciones que son compartidas por sus amigos en común, dándole cierta confianza al usuario que compartió un determinado contenido. Un mensaje solo debe ser estudiado como noticia falsa cuando tiene poca realidad [10]. Las noticias falsas son creadas con la intención de engañar, sin embargo, para determinar la credibilidad de las noticias que se publican en medios sociales, se puede tener en cuenta la fuente de donde ésta proviene.

Generalmente, la noticias falsas publicadas en medios sociales tienen la intención de distorsionar las emociones o pensamiento de la sociedad. Uno de los objetivos de este tipo de noticias es afectar el estado emocional de la persona que lo lee. De hecho, existe gente contratada para crear noticias falsas llamados *trolls* humanos quienes buscan causar una respuesta de ira, miedo y otros fines. Este tipo de noticias se incrementan principalmente por medio de las plataformas como el caso de jóvenes de Macedonia que generaron ingresos por crear noticias falsas [11].

La proliferación de noticias falsas puede afectar a los usuarios con intereses similares. Varios ejemplos reflejan cómo ciertas noticias falsas han llegado a multiplicarse en personas que siguen a grupos específicos [12]. También están las cámaras de eco en donde los usuarios siguen a personas o entran a grupos con ideas afines y éstos consumen y

comparten la misma información sin verificarla previamente, llegando a convertirse en impulsores de la difusión de información [13]. Pero no todo lo que se difunde en redes sociales son noticias falsas, mucha información es real y sirve para que las personas se informen de acontecimientos como: eventos naturales que acontecen, delincuentes que se buscan, corte de los servicios básicos, entre otros.

Existen diferencias entre las noticias reales y falsas. La propagación de noticias falsas tiende a seguir cierto patrón. En este caso están los bots sociales que son software que crean contenido falso de una manera fácil y son una de las razones por las cuales se propaga una noticia falsa teniendo la capacidad de hacerlo viral y confundir a los usuarios para que no sean capaces de determinar la confiabilidad de éstas. En este sentido, ciertas cuentas de Facebook y Twitter difunden noticias falsas a través de bots sociales, usuarios fantasmas o perfiles falsos. Estos bots son creados y desarrollados mediante algoritmos de Inteligencia Artificial y tienen como objetivo de hacer daño, amplificar una mentira, beneficiar a ciertos grupos, y manipular y difundir las noticias falsas. No obstante, este fenómeno ha traspasado las barreras de las redes sociales, llegando a otros medios digitales como blogs y la web en general.

Lo primero que se lee en una noticia es el título, las noticias falsas tratan de atraer el interés del público y por ello tienen como estrategia escribir con mayúsculas o resaltado, porque su objetivo es generar conmoción y sentimientos negativos en la población. Como se puede observar en la Tabla 1.

Tabla 1. Características de las noticias falsas y reales en redes sociales.

NOTICIAS FALSAS	NOTICIAS REALES
SENTIMIENTO NEGATIVO	Sentimiento informativo
PROPAGACIÓN ACELERADA	Propagación lenta
ENGAÑOSAS	Veracidad
TITULARES IMPACTANTES Y ENGAÑOSOS	Título simples

2.1.1.1. Noticias falsas en redes sociales

El mayor uso de Internet y de las redes sociales ha incrementado las noticias falsas. Como se observa en la Tabla 2, según datos de *Internet World Stats* [14], en el ranking del uso de Internet en América del Sur, Ecuador tuvo un alcance con una penetración de Internet

del 80%. En esa misma fuente se informa que las redes sociales como Facebook y WhatsApp alcanzaron en total 2500 millones de usuarios entre el 2017 y 2018 [15].

Tabla 2. Uso de internet en América del Sur [14]

País	Población 2018	% Población	Uso de internet	% Población (penetración)	% de usuarios	Facebook 2017
Argentina	44,688,864	10.4 %	41,586,960	93.1 %	13.6 %	30,000,000
Bolivia	11,215,674	2.6 %	7,570,580	67.5 %	2.5 %	6,100,000
Brasil	210,867,954	49.2 %	149,057,635	70.7 %	48.7 %	139,000,000
Chile	18,197,209	4.2 %	14,108,392	77.5 %	4.6 %	13,000,000
Colombia	49,464,683	11.6 %	31,275,567	63.2 %	10.2 %	29,000,000
Ecuador	16,863,425	3.9 %	13,476,687	79.9 %	4.4 %	10,000,000
Islas Malvinas	2,922	0.0 %	2,900	99.2 %	0.0 %	3,100
Guayana francesa	289,763	0.1 %	120,000	41.4 %	0.0 %	110,000
Guayana	782,225	0.2 %	395,007	50.5 %	0.1 %	360,000
Paraguay	6,896,908	1.6 %	6,177,748	89.6 %	2.0 %	3,300,000
Perú	32,551,815	7.6 %	22,000,000	67.6 %	7.2 %	20,000,000
Surinam	568,301	0.1 %	340,000	59.8 %	0.1 %	310,000
Uruguay	3,469,551	8.0 %	3,059,727	88.2 %	1.0 %	2,400,000
Venezuela	32,381,221	7.6 %	17,178,743	53.1 %	5.6 %	13,000,000
Total	428,240,515	100.0 %	306,349,946	71.5 %	100.0 %	266,583,100

Asimismo, el Instituto Nacional de Estadística y Censos (INEC) en el 2018 reportó que en los últimos 5 años más personas usaron el internet con un aumento de 20.7 puntos porcentuales a nivel nacional; así mismo con un 20.5% en el área urbana y el 20.1% en el área rural y que el 36% utiliza redes sociales desde sus teléfonos inteligentes [16]. Este sin duda alguna es un desencadenante para que la difusión de noticias no verificadas haya crecido considerablemente en los últimos años.

2.1.1.2. Noticias falsas en Facebook

La red social Facebook creada hace más de 16 años es una de las plataformas más populares de Internet. Además de poder comunicarnos, nos permiten estar informados con las noticias del mundo, pero sin saber si esta información es válida, por lo que para muchos usuarios es muy difícil reconocer si una noticia es verdadera o no lo es. Un claro ejemplo fue en el 2016 en Estados Unidos donde se compartió información falsa 3 meses antes de las elecciones presidenciales sobre los candidatos que buscaban llegar a la Casa Blanca [1]. En este mismo país, se midieron las tendencias en la difusión de contenido de

569 sitios web y de noticias falsas en Facebook entre enero de 2015 y julio de 2018. Esta evaluación reveló que 38 sitios de noticias obtuvieron alrededor de 200 a 250 millones de interacciones en Facebook por mes y que las interacciones cayeron de un pico de alrededor de 160 millones por mes a fines de 2016 a aproximadamente a 60 millones [17].

Otro ejemplo, también relacionado con Estados Unidos, fue en las elecciones presidenciales de 2016 en donde Hillary Clinton se vio sumergida y otras figuras políticas en la que según estaban coordinando una red de trata de niños de una Pizzería de Washington. Estas publicaciones fueron compartidas a través de Facebook generando miles de interacciones [18].

Ecuador también ha sido golpeado por las noticias falsas. Una de ella fue durante los días del Paro Nacional de octubre de 2019, que trataba de que el presidente de la República Lenin Moreno ordenó matar a las personas que se encontraban en las manifestaciones, y el precio del gas se incrementaría debido a que se quitaría el subsidio. Además en las publicaciones se trató de crear una especie de xenofobia contra los ciudadanos venezolanos, manifestando que el presidente les otorgaba un bono, luego determinaron que la noticia era falsa [6].

2.1.1.3. Noticias falsas en Twitter

Otra red social que cuenta con más de 328 millones usuarios activos y ha sufrido por las noticias falsas es Twitter. Muchos de los usuarios con el fin de desinformar se convierten en productores de noticias falsas creando cuentas con perfiles falsos, dándole la facilidad de difundir fácilmente y que otras personas le den *retweet*.

De esta plataforma se puede extraer datos para analizar las emociones e identificar opiniones. La información errónea también se puede transmitir de mala gana. Por ejemplo, en noviembre de 2010, la cuenta de Twitter del asesor presidencial para la gestión de desastres de Indonesia fue pirateada. El hacker luego usó la cuenta para publicar una falsa advertencia de tsunami [19].

Las noticias falsas también han estado involucradas con la manipulación ilegal del mercado de valores. Un caso relacionado con esta temática sucedió en 2013, el valor de las acciones de \$ 130 mil millones de dólares de valor de las acciones fueron eliminados

luego de un tweet falso sobre una "explosión" que hirió a Barack Obama, después de ser verificado como falso las acciones se recuperaron [20].

En la misma red social, también se ha publicado noticias falsas sobre desastres naturales, por ejemplo, el caso del huracán Sandy que causó destrucción en Estados Unidos en 2012. Las imágenes difundidas se volvieron virales en la red social Twitter lo cual provocó pánico y caos entre la sociedad [21].

Asimismo, en el caso de la realidad ecuatoriana, durante el proceso electoral de referéndum en Ecuador en 2018, se difundió noticias falsas en Twitter a favor del sí y en contra de la consulta popular que se planteaba realizar en dicho país. En su conjunto las publicaciones alcanzaron 1.507 *retweets*, 1.146 me gusta, y 112 comentarios [4].

Por los casos de noticias falsas originados en la red social Twitter, en Estados Unidos, entre enero de 2015 y julio de 2018, se realizó un monitoreo de la red social el cual reveló que en Twitter, las acciones de contenido falso estuvieron en el rango de 3 a 5 millones por mes desde fines de 2016, esto es, aproximadamente 20 millones por mes [17].

2.1.1.4. Noticias falsas en otros medios sociales

La difusión de contenidos falsos ha aumentado también en los últimos años en medios tradicionales de Internet. Un caso particular fue el de la empresa *United Airlines* donde un artículo publicado en Internet les causó grandes pérdidas económicas, sin embargo, luego de confirmar que la noticia era falsa el precio de las acciones se recuperaron.

Otro ejemplo de noticia falsa en medios tradicionales fue el caso en el que se compartió un artículo que incluía una fotografía de un ciudadano que según estaba decepcionado con Singapur, por lo que estaba pensando en renunciar a su ciudadanía. El artículo se compartió 44.000 veces y los comentarios fueron sentimientos xenófobos en la que fue afectado él y su familia [22].

En Brasil en el 2018 la aplicación WhatsApp se utilizó para difundir noticias falsas debido a que el 44% de la población usa WhatsApp para leer información sobre la política y esto ha permitido envenenar la política brasileña [23].

2.1.2. Detección automática de noticia falsas con IA

En la actualidad, la IA es una de las disciplinas ampliamente usadas para apoyar en el proceso de detección de noticias falsas. Las técnicas de IA que se identificaron como las más usadas son: el aprendizaje automático [24], aprendizaje profundo (redes neuronales) [25] y la minería de datos [26]. Estas técnicas son integradas en una técnica adicional, relacionada con el procesamiento del lenguaje natural, denominada minería de textos.

2.1.2.1 Minería de texto

El término minería de texto se usa comúnmente para denotar cualquier sistema que analice grandes cantidades de texto en lenguaje natural y detecte patrones de uso léxico o lingüístico en un intento de extraer información probablemente útil [27]. En términos generales, la minería de textos permite analizar una cantidad información almacenada en forma de texto escrito y con esto extraer nuevo conocimiento.

El proceso para realizar minería de textos consiste en dos etapas principales: una etapa de preprocesamiento y una de descubrimiento [28]. En la primera etapa permite tener la información ya con una estructura; luego en la segunda etapa se plantea la realización de un análisis que permite reconocer patrones y con esto obtener conocimiento [28]. Es en esta etapa en la que se integra el aprendizaje automático, aprendizaje profundo y la minería de datos, tres técnicas ampliamente empleadas para el reconocimiento de patrones y el descubrimiento de patrones sobre datos.

Los datos basados en texto compartidos por los usuarios en las redes sociales generalmente están en formas no estructurada [7]. Por ese motivo, éstos deben transformarse con una forma estructural con métodos de minería de texto, apoyado de la técnica de procesamiento del lenguaje natural [26]. Como se observa en la Figura 1, el proceso comienza con una colección de documentos, luego se realiza el procesamiento de datos, esto es, eliminar el ruido que los datos puedan tener y limpiarlos para que se puedan procesar por medio de la herramienta de PLN que se elija. Después, el documento pasa por una fase de análisis de texto que es un análisis semántico para derivar información de alta calidad del texto. El último proceso constituye un proceso de descubrimiento de conocimiento sobre el texto [28].

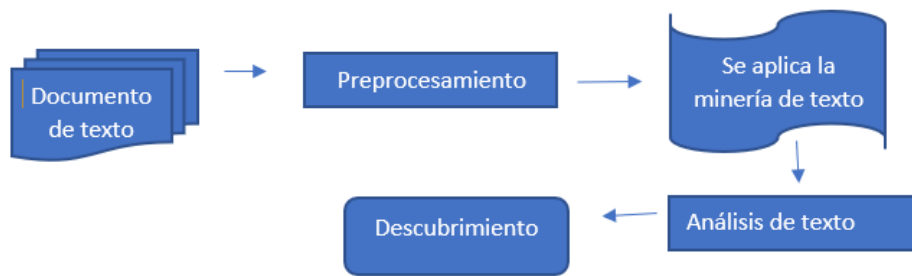


Figura 1. Proceso de minería de texto [28].

Los corpus tienen muchas oraciones y esto tienden a ser ruidosos disminuyendo el rendimiento de los clasificadores automáticos implementados mediante los algoritmos de aprendizaje automático, modelos de redes profundas, y algoritmos de minería de datos; por ese motivo, se debe realizar una limpieza de los datos de manera que se elimine signos de puntuación, palabras vacías, entre otros, y de esta manera, obtener información relevante usando algunas de las técnicas existen diferentes técnicas de PLN como son: Word2Vec, término frecuencia (TF), Frecuencia de Término – Frecuencia Inversa de Documento (TF-IDF), entre otras.

Las técnicas especificadas previamente permiten mejorar el rendimiento de los sistemas de minería de texto ya que se especializan en reducir el tamaño de las características del texto para que los clasificadores automáticos mejoren su rendimiento. A continuación, se describen brevemente las tres técnicas antes mencionadas:

- **Word2Vec.** Es una técnica que toma un corpus y construye un vocabulario a partir de los datos del texto de entrenamiento y luego aprende la representación vectorial de palabras, además, permite encontrar las palabras más cercanas a una palabra especificada por el usuario, se encuentra en dos formas modelo de bolsa continua de palabras (*CBOW*) o el modelo *Skip-Gram* [29], [30].
- **Bolsa de palabras (*Bag-of-words, BoW*).** Es un método que está basado en diccionarios en donde cada documento parece una bolsa que contiene una colección de palabras. También se puede definir como una matriz término-documento, en donde cada valor del vector representa una palabra, ya sea como frecuencia, peso o existencia en el documento [31].
- **N-Gramas.** Es una subcadena de n caracteres, dependiendo de n indica cuántos elementos debe tomarse en cuenta y se denominan unigrama, bigrama, trigramas,

quandgrama $n=4$ y así sucesivamente. Un ejemplo se puede hacer con la palabra TEXTO se la puede representar como un bigrama $_T, TE, EX, XT, TO, O_$ [32].

- **Frecuencia de término (TF).** Este método utiliza la técnica de la *BoW* para evaluar correspondencia de documentos, permite la construcción de modelos de recuperación de información avanzados de tipo vectorial. Cada documento contiene bolsa de palabras que se han representado como vector de longitud. La posibilidad que este modelo tiene para construir un espacio de múltiples dimensiones se debe a que es capaz de cuantificar el peso de cada palabra en una colección específica [33].
- **Frecuencia de término y frecuencia de transacción inversa (TF-IDF).** Esta técnica se ha utilizado para medir la importancia del término en un conjunto de datos. Mientras mayor es el número de palabras más alto es el peso del término. En [34] los autores han aplicado la minería de texto y dos técnicas de extracción de característica las cuales fueron *TF* y *TF-IDF* con el fin de mejorar el rendimiento de los clasificadores.
- **Análisis de componente principal (PCA).** Es un método muy conocido de reducción bidimensional que utiliza una transformación lineal para formar un conjunto de datos simplificado que conserve las características del conjunto de datos original [35]. El nuevo conjunto de datos puede tener una cantidad igual o menor de características que el conjunto de datos original [36].
- **Chi-cuadrado.** Es un algoritmo de selección de característica, hay dos tipos de pruebas para las que se utiliza chi-cuadrado es la prueba de independencia y de bondad de ajuste, este también es un enfoque de reducción bidimensional que investiga la correlación de las características, en donde las características que tengan correlación se mantienen y las restantes se descartan [36]

Luego de aplicar las técnicas de preprocesamiento de texto, se puede utilizar la nube de palabras (*Word Cloud*), también llamada nube de etiquetas, que muestra los miembros de una dimensión elegido como texto, pero en diferentes tamaños y colores. Para la detección de noticias falsas se la puede usar para analizar las palabras que más se repiten en el conjunto de datos [37]. Herramientas como esta se encuentran implementadas en las

principales herramientas especializadas en PLN existentes en la actualidad. Entre las principales herramientas de este tipo figuran las siguientes:

- **Natural Language Toolkit** (*NLTK* por sus siglas en inglés) es una herramienta desarrollada en el lenguaje Python para el procesamiento del lenguaje natural (*PLN*) usada para el análisis de texto, NLTK no sólo se centra en gente que trabaja en PLN viniendo desde la computación sino también en lingüistas haciendo trabajo de campo [37]. Esta herramienta se puede usar en Windows, Linux y Mac OS X [38]. Sus creadores fueron Steven Bird, Edward Loper, Ewan Klein.
- **TextAnalys**. Es una herramienta para la minería de texto donde utiliza técnicas lingüísticas, estadísticas y de aprendizaje automático que permite obtener información de datos no estructurados además, permite realizar tareas como clasificación de texto, análisis de sentimientos entre otros permitiendo extraer información importante y esto ayuda a las empresas a conocer tendencias en la industria y así tomar mejores decisiones comerciales [39].
- **OpenNLP8**. Es una biblioteca está desarrollada en Java por Apache OpenNLP y ofrece diferentes funcionalidades como tokenización, detección de oraciones, correferencias entre otras y se las puede encontrar en varios idiomas ya sea inglés, alemán español, holandés, portugués [38]. También OpenNLP ofrece varios modelos prediseñados para diferentes lenguajes utilizando técnicas de aprendizaje automático, OpenNLP no está completamente desarrollado como otras tareas, sin embargo, ya han realizado algunas avances en esta área [40].
- **PolyAnalist**. Esta herramienta permite realizar varios procesos entre ellas predicciones, un punto importante es que permite analizar grandes conjuntos de texto [63]. Para usar esta herramienta se requiere de una licencia de pago, además, permite ver resúmenes de textos, navegar de forma eficiente a través de grandes bases textuales [41].
- **GATE**. Es un software que permite resolver problema de procesamiento de texto, empezó en 1995 y fue desarrollada por la Universidad de Sheffield [37]. Está basada en Java puede verse como una arquitectura en donde se definen la organización y las responsabilidades, un framework proporcionan diferente componentes de procesamiento. y un ambiente de desarrollo ayuda a disminuir el

tiempo de desarrollo de una api y una interfaz. Además, que incluye recursos para algoritmos y estructuras de datos [42].

2.1.2.2. Aprendizaje automático

El Aprendizaje Automático, conociendo ampliamente en inglés como *Machine Learning*, es una rama de la Inteligencia Artificial que tiene como objetivo lograr que las computadoras aprendan. Para cumplir este cometido, varios tipos de aprendizaje de máquina se han propuesto, los dos principales corresponden al aprendizaje supervisado y no supervisado [43].

Los algoritmos de inteligencia artificial supervisados son una forma de aprendizaje basada en un conjunto de datos usados para llevar a cabo la etapa de entrenamiento. Este tipo de aprendizaje asume que las etiquetas de las instancias en los conjuntos de datos ya son conocidas a priori [7]. Para implementar este tipo de aprendizaje se han planteado dos tipos de algoritmos, esto es, los algoritmos de clasificación y los de regresión.

Los algoritmos de clasificación tienen como objetivo predecir una etiqueta de clase, creando modelos predictivos a partir de datos que tiene características y etiquetas de clases. La clasificación puede ser clasificación binaria (i.e., clasificador de correo *spam*) o multicategoría (i.e., clasificador de animales). Por otro lado, los algoritmos de regresión se usan para predecir valores de salida basados en características de entradas, siendo estos continuos [44] (i.e., predicción de las ventas de una tienda electrónica).

Por otro lado, los algoritmos de agrupar los datos en clústeres con el objetivo de lograr crear modelos basados en similitudes de éstos (i.e., segmentación de clientes de acuerdo con sus hábitos de consumo).

Es importante señalar que tanto para implementar el aprendizaje supervisado y no supervisado se han planteado varios algoritmos y modelos estadísticos/matemáticos. Entre los algoritmos más populares e importantes figuran los siguientes: regresiones (lineal simple, lineal múltiple, polinomial), clasificadores (árboles de decisión, bosques aleatorios, k-vecinos más cercanos, máquina de vectores de soporte, regresión logística, Naïve Bayes) y de agrupación (k-medias). A continuación, se describe brevemente en qué consiste cada uno de los algoritmos:

Regresión lineal simple. Es un algoritmo de aprendizaje supervisado que tiene un enfoque lineal para modelar la relación entre una variable dependiente (y) y otra denominada variable independiente (x). En la Figura 2 se muestra un ejemplo de datos que obedecen el criterio de linealidad requerido por el modelo de regresión lineal simple. Llevándolo al terreno de un análisis de variables reales, las variables pudieran ser el número de matrículas de educación primaria (y) en base a la población de una determinada zona geográfica (x). Este constituye el modelo más sencillo de las regresiones que permiten llevar a cabo tareas de predicción de valores nominales; sin embargo, es importante señalar que previo a su uso se debe de verificar que los datos cumplan con el criterio de linealidad; caso contrario, los resultados obtenidos por el modelo no serán confiables en la toma de decisiones.

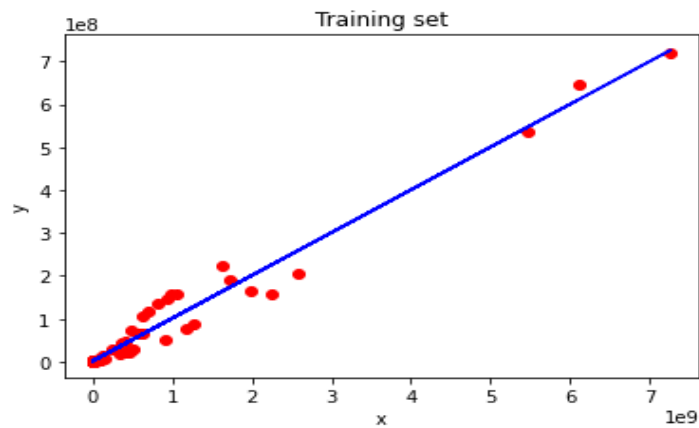


Figura 2. Ejemplo de Regresión lineal simple. Fuente: Elaborado por la autora.

Regresión lineal múltiple. Es un algoritmo que trata de ajustar modelos lineales entre una variable dependiente (y) y más de una variable independiente (x_1, x_2, \dots, x_n). En la ecuación 1 se puede observar la fórmula para el cálculo de la variable dependiente en base a k variables independientes (x) [21]. El parámetro $\beta_j, j=0,1$ se llama coeficiente de regresión parcial. Es importante señalar que para estimar el coeficiente de la regresión, que consiste en el elegir β_j , se usa el método de mínimos cuadrados el cual propone que la suma de los cuadrados de los errores ϵ_i se minimice [45].

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_K X_K + \epsilon \tag{1}$$

Regresión logística (LR). Es una técnica multivariable de dependencia en donde es un algoritmo utilizado para la clasificación de dos clases, que clasifica un punto dado según la probabilidad de que pertenezca a una clase (i.e., correo spam o no spam). En la ecuación 2 se muestra la función de regresión logística.

$$P(X) = \frac{e^x}{1+e^x} \quad (2)$$

Debido a que la probabilidad de que ocurra un evento debe estar acotada entre 0 y 1 [46], éste construye un modelo basado en una variable objetivo. Si existen 2 clases la LR reemplaza a la variable original. En $\Pr[1(a_1, a_2, \dots, a_k)]$ no se puede aproximar la función lineal entre 0 a 1, pero pueden estar entre infinito negativo y el infinito positivo llamándola función sigmoideal. El modelo resultante con pesos w es $\Pr[1(a_1, a_2, \dots, a_k)] = 1/(1 + \exp(-w_0 - w_1 a_1 - \dots - w_k a_k)$ [47]. En la Figura 3 se muestra la función logística tiene forma de “S”, y reduce cualquier cantidad a un valor entre los límites 0 y 1 [48].

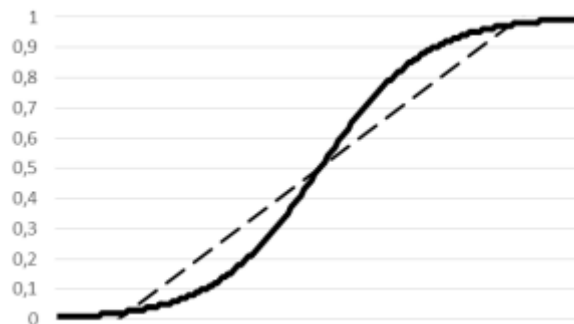


Figura 3. La función logística [48].

Naïve Bayes (NB). Es un algoritmo de clasificación de aprendizaje automático basado en el teorema de Bayes. Éste utiliza métodos de búsqueda y aplica que las características son independientes, es decir, la presencia de una palabra en un artículo no influye en la presencia de otra palabra en el mismo artículo [49]. En la ecuación 3 de teoremas de Bayes en la cual están los siguientes componentes $p(h | D)$ esta es la probabilidad de que se produzca una hipótesis, siempre que se tenga un conjunto de datos.

$$p(h|D) = \frac{p(D|h)p(h)}{p(D)} \quad (3)$$

Donde, $p(D | h)$ es la probabilidad de tener los datos, dada una hipótesis, $p(h)$ es la probabilidad de que una hipótesis tenga lugar, en general, y $p(D)$ esta es la probabilidad de tener los datos antes de conocer alguna hipótesis [50]. En la Figura 4 se muestra la estructura de Naïve Bayes en donde X_1, X_2, \dots, X_n corresponden a las variables o atributos que permiten predecir el valor de la clase [51] (i.e., día apto para hacer deportes en ambiente exterior o no).

Likelihood Table				
Outlook	Yes	No		
Sunny	3	2	$= (5/14)$	0.36
Rainy	2	3	$= (5/14)$	0.36
Overcast	4	0	$= (4/14)$	0.28
	$= (9/14)$	$= (5/14)$		
	0.64	0.36		

$P(\text{Sunny} | \text{Yes}) = 3/9 = 0.33$
 $P(\text{Sunny}) = 5/14 = 0.36$
 $P(\text{Yes}) = 9/14 = 0.64$

Figura 4. Ejemplo de Naïve Bayes [51].

Árboles de decisión (DT). Es un algoritmo de aprendizaje supervisado que emplea la estructura árbol para predecir una clase en base a reglas y esto obtener soluciones al problema [14]. Presenta ventajas competitivas ya que proporciona un alto grado de comprensión, facilitando la toma de decisiones y explicando el comportamiento de una tarea determinada. Los árboles de decisión pueden ser binarios o de más opciones. Su funcionamiento consiste en generar un modelo mediante divisiones en el árbol los cuales se representan como nodos, para agregar un nodo debe tener una correlación significativa con la columna de predicción y con esto va formando la estructura de un árbol [52]. En la Figura 5 se muestra un ejemplo de árbol de decisión que divide al árbol dependiendo de las condiciones humedad puede ser alta o normal y viento verdadero o falso refiriéndose a débil o fuerte [53].

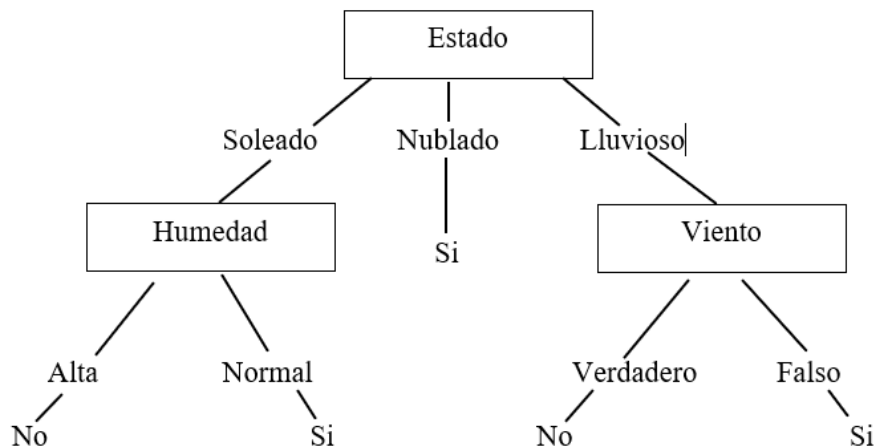


Figura 5. Ejemplo de Jugar Tenis [53].

Bosques aleatorios (*Random Forest, RF*). Es un algoritmo que tiene una combinación de árboles predictores en donde cada árbol de decisión da una clasificación y la que tenga más votos es la predicción del algoritmo. La diferencia entre un árbol de decisión y un bosque aleatorios radica en que la característica elegida en cada nodo es aleatoria en lugar de la mejor opción y se usa un subconjunto aleatorio del conjunto de entrenamiento para entrenar cada árbol [46]. En la Figura 6 se ilustra un bosque al azar con dos árboles.

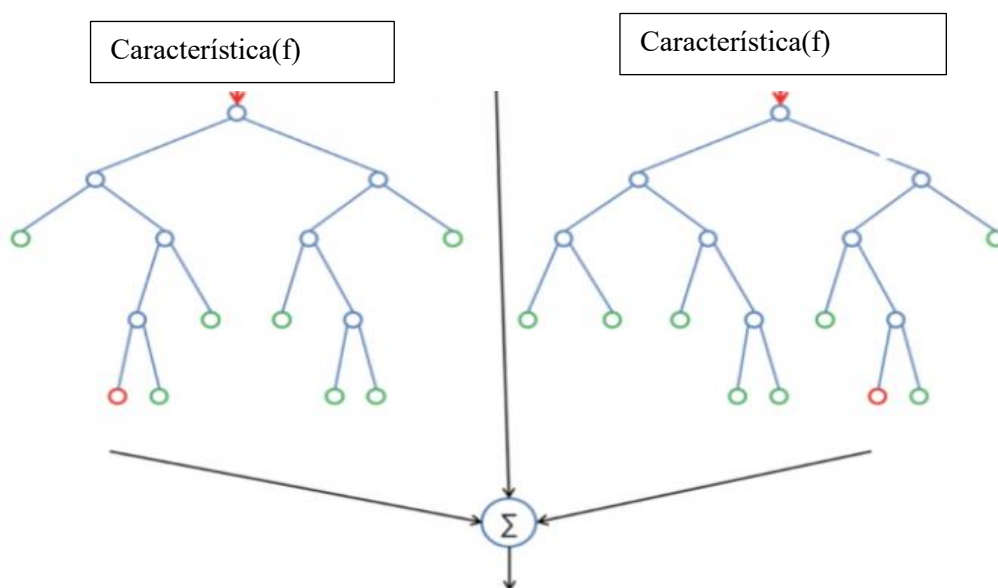


Figura 6. Ejemplo de bosque aleatorio.

K-vecinos más cercanos (*KNN*). Es un algoritmo de aprendizaje supervisado, que funciona como un clasificador y también para regresión [24]. La fórmula de *KNN* se la muestra en la ecuación 4, en donde $NK(x, D)$ son los (índices de) K puntos más cercanos a x en D y $\Pi(\epsilon)$ es la función indicadora definida como: 1 si he es verdadero y 0 si he es

falso [54]. Teniendo un conjunto de prueba este encuentra k-vecinos más cercanos entre los documentos de entrenamiento [55].

$$p(y = c|x, D, K) = \frac{1}{K} \sum_{1 \in N_K(x,D)} I(y_1 = c) \quad (4)$$

En la Figura 7 se muestra un ejemplo del algoritmos KNN donde se quiere clasificar un objeto de clase desconocida, en este caso el círculo; entonces si se toman los 3 vecinos más cercanos (k=3), la clase a la que pertenece el círculo es a la de los triángulos, mientras que si k=5, ya pertenece a la clase de los cuadrados [56].

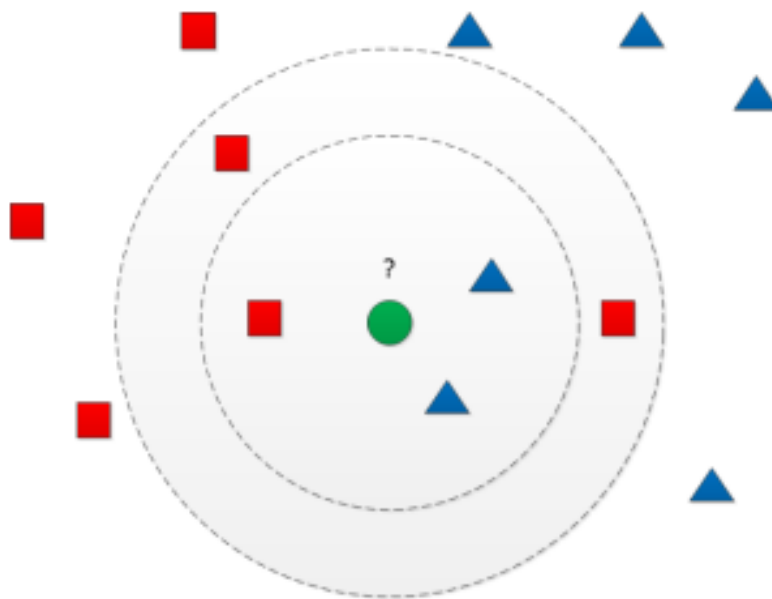


Figura 7. Ejemplo sobre cómo analizar el algoritmo KNN [56].

Máquinas de vectores de soporte (*Support Vector Machines, SVM*). Las máquinas de vector de soporte son implementadas mediante un algoritmo que corresponde a la categoría de métodos de *kernel* y puede ser usado para clasificación y regresión. Es un algoritmo de aprendizaje supervisado, que se encarga de separar los puntos de entrenamiento en dos o más clases. En la ecuación 5 se muestra la clasificación binaria en donde m es el número de instancias de entrenamiento de un set de datos x donde cada entrada x_i tiene un número determinado de atributos y clases definida por $y_i = -1, 1$.

$$\{x_i, y_i\} \text{ donde } i = 1 \dots m, y_i \in \{-1, 1\}, x \in R^n \quad (5)$$

En la Figura 8, aceptando que los datos son linealmente separables, por $w^T x + b = 0$ divide con una línea a las dos clases, denominando SVM a los puntos que están en $w^T x + b = 1$ y $w^T x - b = 1$ y los “+” que representan la clase positiva +1 y “-” clase negativa -1 [57].

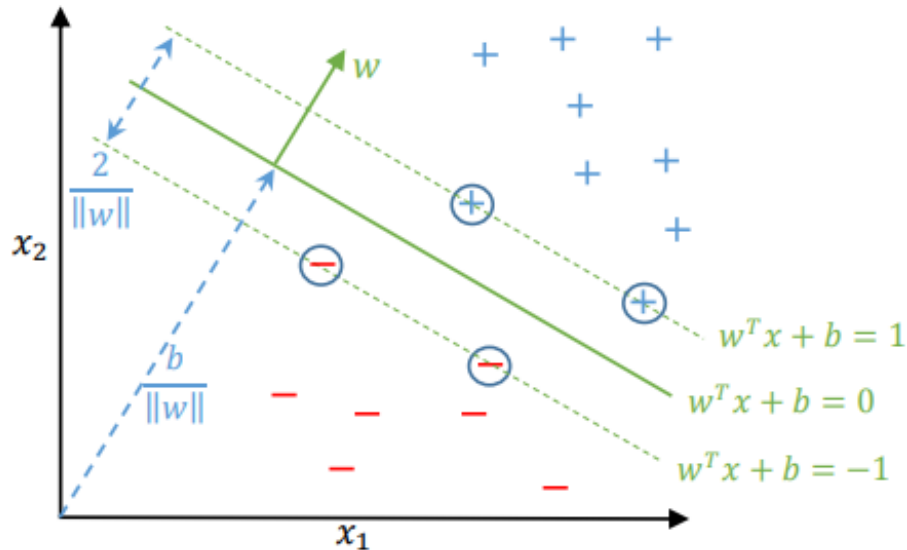


Figura 8. Interpretación de geometría de SVM [57].

K-medias (*k-means*). Es un método de agrupación o *clustering* que agrupa objetos en k grupos basándose en sus características. Este algoritmo divide en k grupos, dados por centroides, cada elemento de un grupo asignándole el centroide más cercano de manera que cada grupo se actualiza en función de los puntos asignados [47]. En la Figura 9 se muestra 4 grupos y se ha marcado cada centroide con un punto amarillo este ejemplo fue realizado con el conjunto de datos iris.

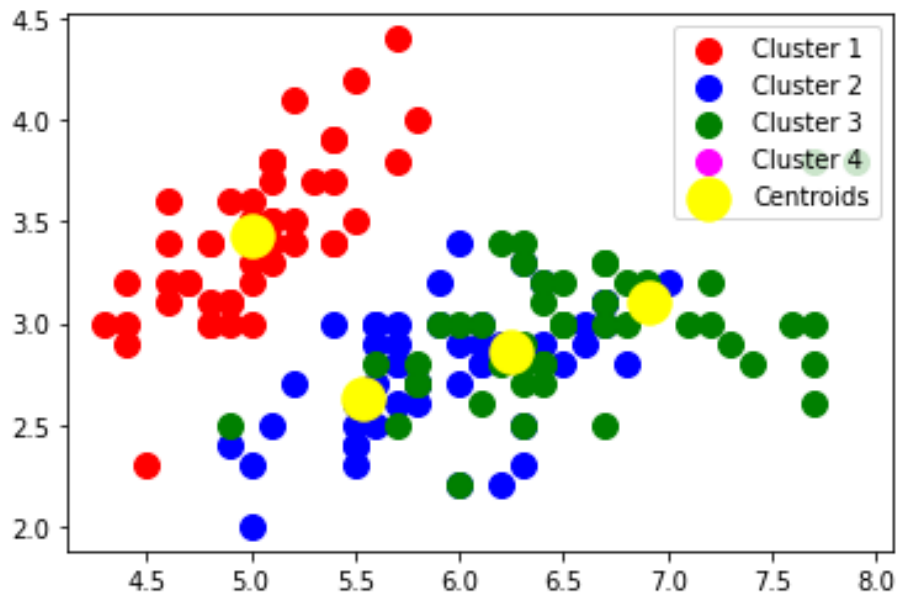


Figura 9. Ejemplo de *k-means* Fuente: Elaborado por la autora.

Herramientas para aprendizaje automático. Las principales herramientas utilizadas para realizar pruebas de aprendizaje automático están orientadas al lenguaje Python, Java y Matlab. Entre las más relevantes figuran las siguientes:

- **Scikit-learn.** Esta herramienta contiene una biblioteca completa de algoritmos de aprendizaje automático [54]. Es un biblioteca gratuito para el lenguaje de programación Python, y está diseñada para interoperar con las bibliotecas numéricas y científicas *NumPy* y *SciPy* [58]. El autor es David Cournapeau y surgió como un proyecto de *Google Summer of Code* en 2007.
- **Shogun.** Shogun es una biblioteca de software de aprendizaje automático de código abierto escrita en C ++. Ofrece numerosos algoritmos y estructuras de datos para problemas de aprendizaje automático. Ofrece interfaces para Octave, Python, R, Java, Lua, Ruby y C # utilizando SWIG [58].
- **PyTorch.** Es una librería de código abierto de Python, otorga una interfaz sencilla para la creación de redes neuronales. Es compatible con las principales plataformas en la nube, lo que proporciona un desarrollo sin fricciones y un escalado sencillo a través de imágenes prediseñadas y se suele usar para sustituir la librería numpy y que además suele trabajar con grafos dinámicos permitiendo modificar en tiempo de ejecución [59].
- **Apache Singa.** Iniciado en el 2014 por la Universidad Nacional de Singapur y Zhejiang, también tiene una gama amplia de modelos de aprendizaje profundo

entre sus características están que se admite la abstracción de dispositivos para su ejecución en dispositivos de hardware y tienen 3 componentes las cuales son IO, Core y Model [60].

- **Mahout.** Es una librería de aprendizaje automático escrita en Java incluye gran número de los algoritmos, no provee una interfaz de usuario o un instalador sino más bien es un conjunto de herramientas [61]. Apache Mahout es un proyecto que se encarga de producir implementaciones gratuitas de algoritmos de aprendizaje automático [58].

2.1.2.3. Aprendizaje profundo

Otra de las técnicas de aprendizaje automático, que se caracterizan por la extracción de características sin supervisión para su posterior procesamiento, y clasificación de patrones son las redes neuronales [62]. El concepto “profundo” del aprendizaje profundo, viene referido al número de capas. El aprendizaje profundo, incluye aquellas técnicas de aprendizaje automático que hacen uso de arquitecturas de redes neuronales y poseen técnicas que tratan de imitar el aprendizaje de los humanos para tener lograr aprender y adquirir cierto conocimiento [63].

Para la implementación de modelos de aprendizaje profundo se emplean modelos de redes neuronales. Las redes neuronales artificiales (*RNA*) se asemejan al funcionamiento del cerebro humano ya que en su estructura se cuenta con neuronas interconectadas que van formando la red, y a partir de ellos crear modelos artificiales para poder realizar tareas complejas y solucionar problemas complejos como: identificación de individuos, reconocimiento de objetos, identificación de tumores, entre otras.

Como se observa en la Figura 10, las redes neuronales artificiales constan en líneas generales de 3 capas que están interconectadas, esto es, la capa de entrada, la(s) capa(s) oculta(s) y la capa de salida. La primera capa recibe los valores de entrada, en la capa oculta se reciben los valores de la capa de entrada y realiza cálculos matemáticos con esas entradas. Y en la última capa se proporcionan la respuesta de la red. Esta no es la única estructura de una red neuronal, puede estar construida por un número específico de capas ocultas, todo depende del modelo en específico, que más adelante se van a describir.

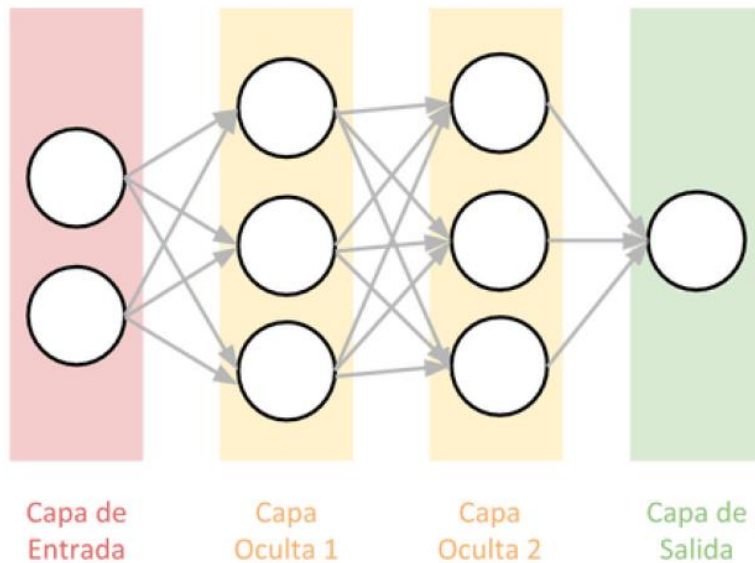


Figura 10. Esquema general de una red neuronal [63].

En la Figura 11 se muestra como una neurona consta de un conjunto de pesos los cuales suelen ser representados por w y otras ocasiones con la letra del alfabeto griego ϑ ; además, incluye el valor de bias, generalmente representada por b [57]. El parámetro pesos debe ser inicializado generados de forma aleatoria.

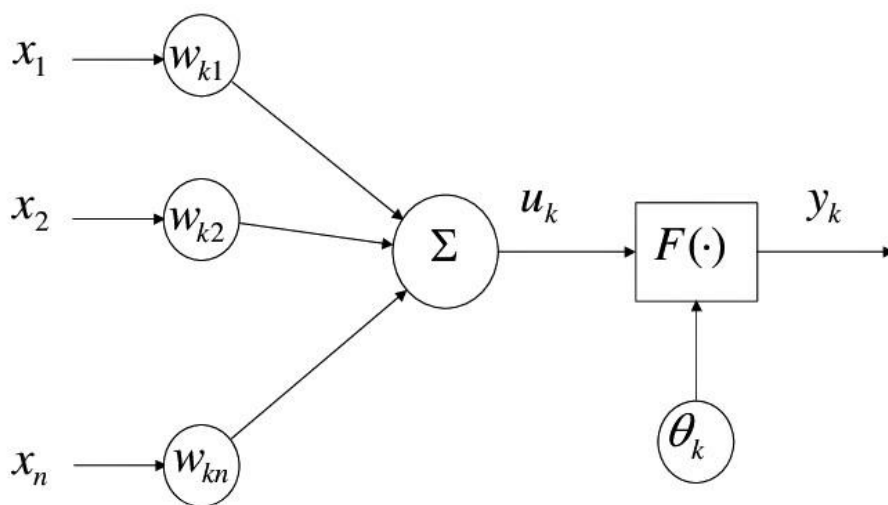


Figura 11. Representación de una Neurona [57].

Una vez inicializados los pesos, éstos son procesados mediante una función de agregación, que generalmente suma los productos de los pesos por las entradas, y éste a la vez es pasado a la función de activación a fin de determina la intensidad del impulso de una neurona. Algunas de las funciones de activación son sigmoidea, identidad,

tangente hiperbólica y arco – tangente. Las ecuaciones que la definen se muestran respectivamente en (6), (7), (8) y (9) [57].

$$g(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

$$g(z) = Z \quad (7)$$

$$g(z) = \frac{e^{aZ} - e^{-aZ}}{e^{aZ} + e^{-aZ}} \quad (8)$$

$$g(z) = \tan^{-1}(z) \quad (9)$$

Para implementar el modelo base de red neuronal artificial y de neurona se han propuesto varios modelos. Cada modelo está orientado a resolver problemas de clasificación, modificando el algoritmo de aprendizaje. Aunque existen varios modelos de redes neuronales, conocidos también como redes profundas, en este estudio se han descrito en detalle cuatro modelos, esto es, redes con propagación hacia adelante, redes recurrentes, redes convolucionales, y redes con memoria. Estos han sido los modelos más utilizados en la detección de noticias falsas.

Perceptrón multicapas. Constituye un modelo que lleva a cabo dos procesos: propagación hacia adelante y retro-propagación. La propagación hacia adelante (*forward propagation*) es el proceso de enviar las señales de activación desde la capa de entrada a las neuronas en las capas ocultas que existan hasta llegar a las neuronas en la capa de salida. En la Figura 12 se muestra la propagación en donde cada capa oculta acepta los datos de entrada los procesa según la función de activación y luego los pasas a la capa sucesiva [57].

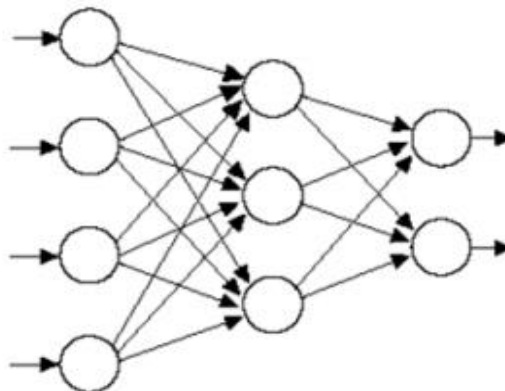


Figura 12. Proceso hacia adelante [125].

Una vez terminado la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. El proceso empieza en la última capa que es la de salida y pasa por todas las capas hasta llegar a la capa de entrada. Existen dos tipos de ajustes el de unidades procesadoras de la capa de salida y el otro grupo que es para la capa oculta; para la capa de salida se refiere a que cada capa produce un número real como salida y este se compara con el valor deseado; y el ajuste de pesos de la capa oculta que no tienen un vector de salida [64].

Como se muestra en la Figura 13 cuando las salidas pueden estar conectadas como entradas de neuronas de niveles previos o del mismo nivel, incluyéndose ellas mismas, la red es de propagación hacia atrás [125].

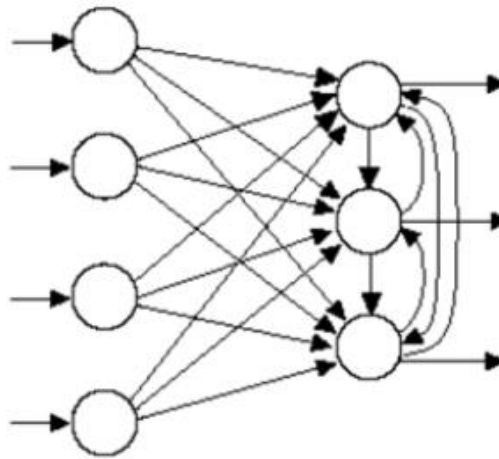


Figura 13. Propagación hacia atrás [125].

Redes neuronales recurrentes (RNR). La red neuronal recurrente o *RNN* por sus siglas en inglés, es una red neuronal artificial que forma un circuito de retroalimentación y constituye una red con conexión recurrente en la que la señal viaja de la capa de entrada hacia la de salida, y tiene una capacidad más grande de representación de la información. Existen diferentes modelos de redes recurrentes como son la de Jordan, la red de Hopfield, y la red de tiempo real (RTRL) [65].

En la

Figura 14 se muestra la arquitectura de una red neuronal recurrente donde $X_n(t)$ son los valores de entrada a la red, $S_h(t)$ representa la salida de los nodos ocultos de la red y $O_o(t)$ representa las salidas de la red, y n es el número de neuronas de la capa entrada. Los valores de las matrices V , U que corresponden a los pesos de la red, aplican las mismas ponderaciones en las diferentes elementos de la serie de entradas [66].

Existe algunos modelos de entrenamiento para una red recurrente los cuales pueden ser: (i) el entrenamiento por épocas la red recurrente inicializa con un estado predeterminado hasta alcanzar el estado esperado se inicializa la red y comienza una nueva época, y (ii) el entrenamiento continuo cuya característica principal es que la red recurrente aprende durante el procesamiento de los ejemplos de entrada.

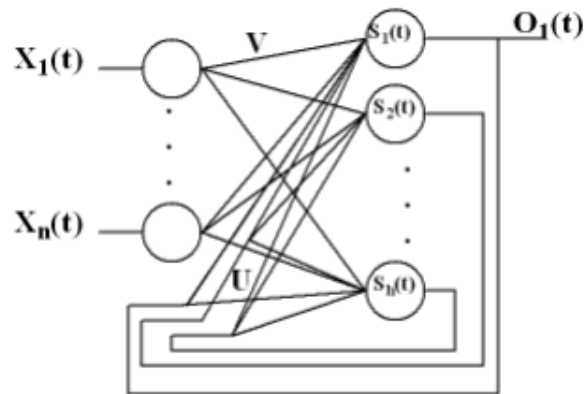


Figura 14 . Red neuronal recurrente [66].

Red neuronal convolucional (CNN). Una red neuronal convolucional (*CNN*) consta de múltiples capas de convulsión y reducción siendo son muy potentes para el análisis de las imágenes debido a que son capaces de detectar características como bordes, líneas. Utilizan perceptrones multicapas y se aplica con mayor frecuencia en el análisis de imágenes [67]. Una *CNN* es una capa de agrupamiento en la se cual trata de reducir la dimensionalidad dividiendo en un conjunto de $m \times n$ bloques, luego se evalúa una función de agrupamiento sobre las respuestas en cada bloque y esto da como resultado un mapa de respuestas de dimensión más pequeño.

Como se muestra en la Figura 15 en un ejemplo de mapa de respuestas convolucionales de entrada 4×4 , al aplicar una función de agrupamiento promedio sobre cuatro bloques de 2×2 , la respuesta es el promedio de los valores en el bloque que representa una reducción de dimensionalidad del mapa de respuesta. Para entrenar una *CNN* puede usar la técnica de propagación del error hacia atrás [68].

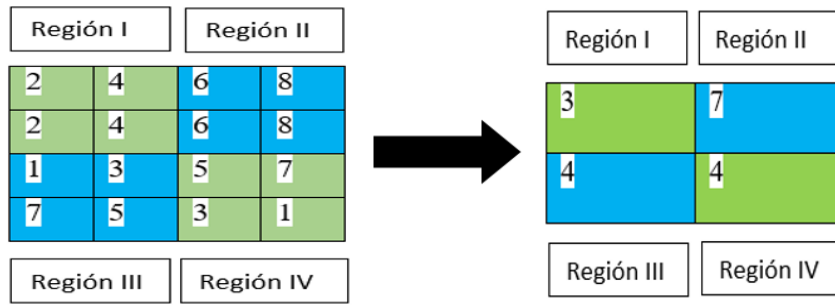


Figura 15. Agrupamiento promedio en una CNN [68].

En la Figura 16 se muestra la arquitectura de una *CNN* la cual consiste en diferentes capas, la primera es convolución en la que se aplican filtros que sirven para la detección de bordes y permiten resaltar las características de una imagen. Existen diferentes filtros entre ellos figuran: (i) laplaciano que es una operación que resalta todos los bordes en una imagen independientemente de sus orientaciones, (ii) gradiente de Prewitt que realiza un realce de borde direccional y por último, (iii) el filtro sobel que detecta los bordes horizontales y verticales separadamente sobre una imagen en escala de grises [69].

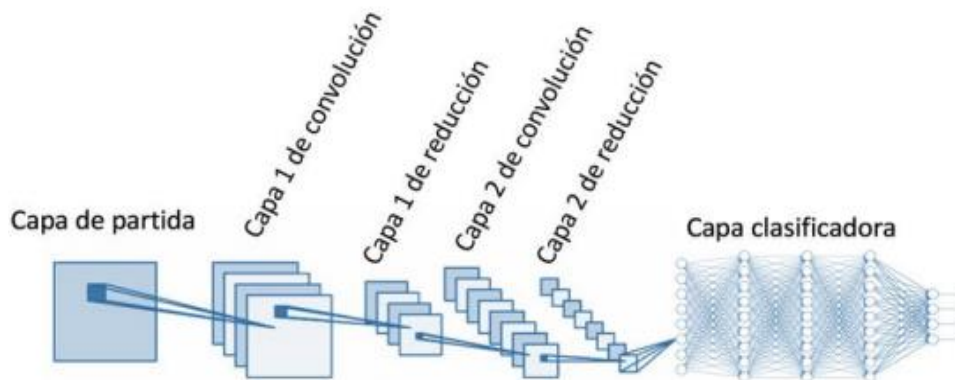


Figura 16. Arquitectura de una *CNN* [69].

Luego se obtiene la salida que está dada por k mapas de características y este pasan por la función de activación ReLu esta función toma valores mayores a 0 y los menores lo convierte en 0. Luego de esto los datos pasan por un proceso de submuestreo a lo largo de la dimensión espacial de la imagen, reduciendo el tamaño de la misma. Se realizan diferentes operaciones como valor máximo del submuestreo (*max-pooling*) o valor promedio del sub-muestreo (*mean-pooling*). En la última capa clasificadora se usan redes neuronales como la *Backpropagation*, y entre otros clasificadores [69].

Memoria larga a corto plazo (LSTM). LSTM son unidades RNN que utilizan las llamadas células de memoria a largo plazo, además, este evade el problema de desaparición del gradiente cuando se propagan los errores [70]. El componente básico es el bloque de memoria que contiene una o más celdas de memoria las cuales contienen una unidad lineal con una conexión recurrente llamada carrusel de error constante (CEC) y una compuerta de entrada y salida. En la Figura 17 se muestra el bloque de memoria con una celda de entrada se representa con Z , donde cada celda de entrada recibe una colección de valores. También se muestra la activación de la compuerta de entrada ϕ , la activación de la compuerta de salida con γ , la activación del CEC x , y la activación global de la celda z . La compuerta de entrada se encarga de permitir o impedir el acceso al CEC al interior de la celda.

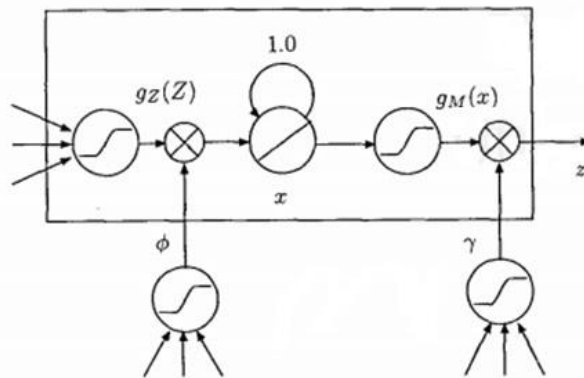


Figura 17. Bloque de memoria con única celda [71].

Estos bloques de memoria configuran a una LSTM y el estado está formado por las activaciones de las compuertas y el CEC y las celdas de los bloques de memoria. En la Figura 18 se muestra una red LSTM donde la entrada es definida como $u[t]$, la salida como $y[t]$, y $W^{\phi,z}$ es el peso para calcular la activación de una compuerta de entrada ϕ partir de una celda z . Además, W^{γ} indica el sesgo se usa para calcular la activación de la compuerta de salida [71].

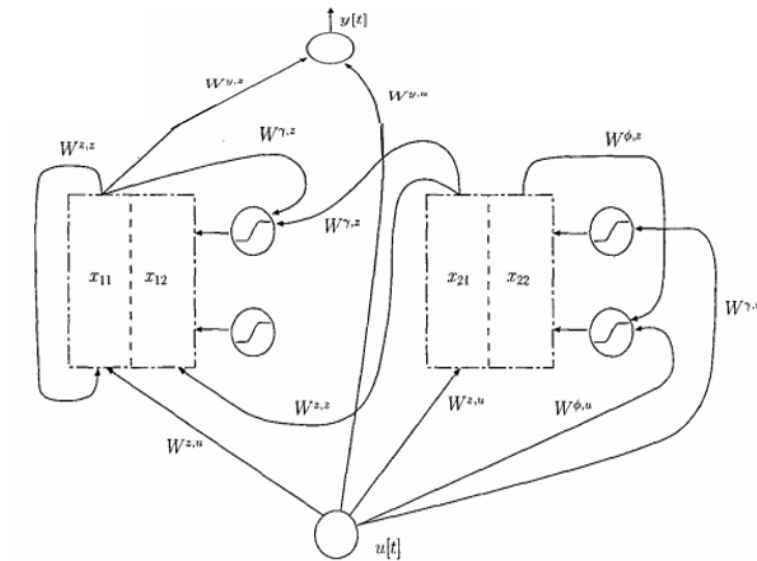


Figura 18. Red LSTM [71].

Herramientas usadas para aprendizaje profundo. Tal como en el área de aprendizaje automático, para crear modelos de redes neuronales profundas existen varias herramientas especializadas. A continuación, se describe brevemente las más extendida hoy en día.

- **Caffe.** Es una biblioteca que permite entrenar y ejecutar redes neuronales, están escrita en C++ / CUDA, además puede ser usada mediante Python y C++ [72]. Además, es uno de los frameworks más antiguos y soporta diferentes tipos de arquitectura de aprendizaje profundo.
- **Tensorflow.** Es una herramienta de código abierto y permite compilar código de redes neuronales profundas orientada al cálculo numérico utilizando grafos de flujo de datos. Fue desarrollado por Google para aprendizaje profundo y soporta Python y C++. También, permitirte distribuir los cálculos en CPU y GPU (varias de forma simultánea) e incluso el escalado horizontal usando sistema de llamada a procedimiento remoto (gRPC) [59].
- **NNTool de Matlab.** Es una herramienta que permite crear, visualizar y simular un tipo de red neuronal, su interfaz es muy agradable permiten ingresar las entradas y salidas y realizar todo el entrenamiento de la red neuronal.
- **Keras.** Es una biblioteca de redes neuronales de código abierto escrita en Python, es amigable para el usuario, modular y extensible y es capaz de

ejecutarse sobre TensorFlow. Contiene varias implementaciones de los bloques para la construcción de las redes neuronales [59].

2.1.2.4. Minería de datos

La minería de datos tiene como tarea el análisis automático de grandes cantidades de datos para extraer patrones y comprender mejor los datos. Como se observa en la Figura 19 la minería de datos tiene un proceso en donde el primero es delimitación de los objetivos; es decir, plantear qué objetivos se quieren alcanzar usando la minería de datos. Luego se realiza el preprocesamiento de datos que permite realizar una limpieza a los datos permitiendo tener lo más esencial [37].

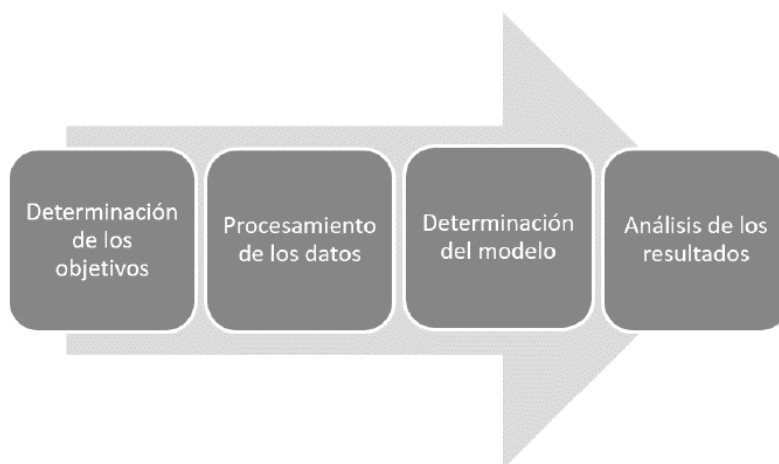


Figura 19. Proceso en minería de datos [37].

A continuación, se procede con la determinación del modelo en donde se debe realizar un análisis estadístico para luego visualizar los datos que fueron analizados y dependiendo de los objetivos planteados usar los diferentes algoritmos propuestos para el descubrimiento de patrones sobre los datos. Finalmente, se aplica una etapa de análisis de los resultados proveídos por los algoritmos de minería de datos aplicados viendo si estos aportan nuevo conocimiento con la que se pueda tomar decisiones [37].

Herramientas para minería de datos. Existen varias herramientas utilizadas para realizar minería de datos. Entre las más relevantes se ha identificado las siguientes:

- **Weka.** Es una herramienta que agrupa algoritmos de aprendizaje automático para tareas de minería de datos, se desarrolló en Java es de distribución de licencia GNU-GLP y dispone de colección de algoritmos para el análisis de datos [73].
- **RapidMiner.** Es una herramienta para procesos de minería de datos que, además consta con algoritmos de aprendizaje automático. Esta herramienta cuenta con interfaces sencillas y utiliza Lenguaje de Marcado Extensible (*XML*) y permite el diseño de cadenas de operadores complejos anidados para un gran número de problemas de aprendizaje [74].
- **Knime.** Está desarrollado sobre la plataforma Eclipse y ha sido programado en Java en el 2006. Es un software de código abierto y ofrece facilidades para crear de forma visual flujos de datos y ejecutarlos para el análisis y con esto estudiar los resultados [75].

Algoritmos de minería de datos en Weka. Una de las herramientas ampliamente usadas para realizar minería de datos es Weka. Esta herramienta implementa un conjunto de 20 algoritmos. Algunos de los algoritmos más populares corresponden a algoritmos de aprendizaje automático (i.e., árbol de decisión o RJ48) y aprendizaje profundo (i.e., Perceptrón Multicapas o MLP) ya descritos. Sin embargo, esta herramienta dispone de una colección de algoritmos adicionales entre los que figuran los siguientes:

- **Naive bayes (NB).** Es un algoritmos de clasificación de aprendizaje automático basado en el teorema de Bayes, este utiliza métodos de búsqueda y aplica que las características son independientes [49].
- **Regresión logística (LR).** Es un algoritmo utilizado para la clasificación de dos clases.
- **Multilayer perceptrón.** Perceptrón Multicapa está formada por múltiples capas de neuronas, redes de clasificadores lineales y los nodos de esta red son todos sigmoide.
- **RBF Network.** RBF implementa una red de función de base radial, son funciones cuya salida depende de la distancia a un punto y utiliza algoritmos de agrupación como k-means [76].

- **Regresión lineal simple.** Es un algoritmo de aprendizaje supervisado su enfoque es lineal para modelar la relación entre una variable dependiente y la variable independiente.
- **SGD.** Implementa el descenso de gradiente estocástico para aprender varios modelos lineales, esta se la puede hacer de forma incremental y también se reemplaza los valores perdidos y permite transformarlos a binarios [76].
- **SMO.** Llamado optimización mínima secuencial es un algoritmo que permite resolver el problema de optimización de programación cuadrática (*QP*) en *SVM* que surge durante el entrenamiento, en la que divide el problema de *QP* en una serie de problemas más pequeños. También este reemplaza globalmente todos los valores perdidos y transforma los atributos nominales en binarios [76].
- **IB1.** El algoritmo IB1 es el vecino más cercano, utiliza la distancia euclidiana normalizada, si esta son iguales a la de prueba se usara la primera [76].
- **IBK.** El algoritmo IBK es idéntico a *KNN* excepto que normaliza los rangos de sus atributos, procesa instancias de forma incremental. Y este algoritmo también puede realizar ponderación a distancia [76].
- **J48.** Este algoritmo está encapsulado en una clase, este crea un árbol de decisiones [76].
- **BAGGING.** Puede realizar clasificación y regresión, además es una clase para embolsar un clasificador permite reducir la variación [76].

2.2 Antecedentes

Se han realizado varios estudios sobre *Fake News* en interés de la sociedad de los últimos años. Un análisis de los trabajos por técnica de *IA* aplicada determinó que mayormente la técnica empleada para la detección de noticias falsas en redes sociales fue la minería de textos. Además, se pudo determinar que la minería de textos se apoyó de técnicas como el aprendizaje automático, el aprendizaje profundo y la minería de datos. Una descripción de los trabajos relacionados publicados en los últimos 5 años (2015-2020) que fueron recuperados, aplicando un proceso de revisión sistemática en las bases de datos científicas *ACM*, *ScienceDirect* y usando una cadena de búsqueda especializada ((method OR technique OR mechanism OR technology) AND detection AND fake news) son descritos como sigue:

En primer lugar, la inteligencia artificial ha servido de apoyo para detección de noticias falsas, con la técnica de aprendizaje automático. En este campo Kareem, y Awan [34] investigaron dos técnicas que permitieron detectar una noticia falsa empleando algoritmo de aprendizaje automático (*ML*). Utilizaron dos técnicas de extracción de características de datos de texto como frecuencia de término (*TF*) y frecuencia de documento inversa de frecuencia de término (*TF-IDF*) y entrenaron siete algoritmos diferentes de *ML* para obtener resultados. Los resultados del experimento mostraron que el mejor clasificador de rendimiento es k-vecinos más cercanos (*KNN*) que proporcionó una precisión del 70% y la regresión logística una precisión del 69%.

Ozbay, Alatas [7] propusieron un método para la identificación de noticias falsas en las redes sociales donde convirtieron el conjunto de datos no estructurado a estructurados con técnicas TF y TF-IDF, y entrenaron 23 algoritmos utilizando los métodos de minería de texto. Los resultados obtenidos mostraron que los modelos basados en árbol de decisión tuvieron una adecuada exactitud (*accuracy*) (74%) y precisión (74%). También, los algoritmos ZeroR, CVPS y WIHW, con un valor de 100% de recall, parecieron ser los mejores algoritmos en términos de métrica de recuperación.

Siguiendo la misma línea, Zhou y Zafarani [77] estudiaron los patrones de noticias falsas en las redes sociales con teorías psicológicas sociales que explicaron la existencias de estos patrones. Los artículos de noticias en estos conjuntos de datos se recopilaron de PolitiFact y BuzzFeed, y entrenaron 5 clasificadores de aprendizaje automático. El clasificador que tuvo mejor resultado fue el de bosques aleatorios que funcionaron mejor en ambos conjuntos de datos, logrando una precisión de alrededor de 93% en PolitiFact y alrededor de 84% en BuzzFeed.

En segundo lugar, otra técnica que también tiene cabida es el aprendizaje profundo. Algunos modelos de redes neuronales funcionan bien para la detección de noticias falsas. Por ejemplo, Qawasmeh, Tawalbeh y Abdullah [25] propusieron una identificación automática del modelo de noticias falsas utilizando técnicas modernas de aprendizaje automático, principalmente aprendizaje profundo y redes neuronales. Con dos modelos Bidireccional *LSTM* concatenado y Multi-head *LSTM*. Utilizaron el conjunto de datos *The Fake News Challenge* (FNC-1) que se presentó en una competencia pública que tenía como objetivo encontrar métodos automáticos para detectar noticias falsas y utilizaron 2 técnicas de extracción de características de datos de texto. Al final se determinó que es mejor usar Multi-head *LSTM* ya que proporcionó buena precisión y exactitud. Siguiendo

la misma línea, Bahad, Saxena y Kamal [78] propusieron un modelo que utilizó la inclusión de palabras GloVe donde usaron modelos de aprendizaje profundo y utilizaron dos conjunto de datos. De los 4 algoritmos el Bidireccional LSTM-RNN fue significativamente más efectivo que los modelos unidireccionales.

En tercer lugar, otra de las técnicas usadas para la detección automática de noticias falsas es la técnica de minería de datos [79]. En general, dicha técnica permitió ver la credibilidad de las noticias acompañado con los modelos de aprendizaje profundo. En esta línea, los resultados demostraron que todos los modelos funcionaron mejor cuando se combinan con las secciones de minería de datos. Siendo *LSTM* aquel que tuvo mejores resultados en combinación con la representación de word2vec.

Finalmente, es importante resaltar que los estudios previamente descritos han sido aplicados con datos generados de redes sociales como Facebook, Twitter [77] [80] y la web en general [79]. Estas son las redes más populares en la sociedad de hoy, y son por tanto, las que los usuarios leen noticias o usan para mantenerse informados de lo que acontece tanto a nivel regional, local o mundial. Por tanto, estos estudios corresponden a los estudios de interés para esta investigación.

2.3. Fundamentación legal

Dentro de las bases legales que deben tomarse en cuenta para el presente proyecto se estipulan las siguientes: Ley Orgánica de Comunicación [81], Constitución de la República del Ecuador [82] y el Código Orgánico Integral Penal (COIP) [83] .

En la Constitución de la República del Ecuador se menciona, en los artículos 16 y 18, que entre los derechos de las personas esta la comunicación al acceso de tecnologías, acceder a información verificada. Con el crecimiento de las tecnologías ha surgido las redes sociales con las cuales las personas se pueden comunicar y acceder a información, pero ésta se puede llegar a ser falsa o verdadera, y extenderse en todo Internet a un ritmo aterrador, con el trabajo de investigación va a ayudar crear sistemas más precisos y la información en la cual se acceda sea verificada.

Por otro lado, en el COIP se menciona en el artículo 307 que las personas que difundan noticias falsas en el dominio financiero y que este ocasione daño a la economía del país con el fin de beneficiarse tiene una penalización con cárcel, además en el artículo 322 manifiesta que si una institución financiera es víctima de noticias falsas se aplica la

misma sanción. Con el presente proyecto se busca combatir la desinformación que manipula la opinión pública en línea que estas juegan con las emociones de las personas y ponen en riesgo la vida y su economía.

Finalmente, en la Ley de Comunicación en los artículos 4, 8, 10, 17, 22, se menciona, que las personas tienen derecho a la información, a compartir contenido personal en las redes sociales, pero hay que ser cuidadosos con lo que se publica pues esto no excluye a otros delitos. Los medios de comunicación deben ser pulcros con la información que transmiten, es decir, dar contenido de calidad. La información transmitida no debe ser contenidos discriminatorios y que afecten a la persona. Los ecuatorianos tienen derechos a expresarse, pero estas opiniones deben ser correctas que no dañen a otra persona, o le ocasionen un mal o muerte, porque será juzgado por la ley. Los ciudadanos tienen derecho a la información y que esta sea contrastada por los medios de comunicación antes de emitirla.

Se ha considerado leyes que protejan a las personas ya que una noticia falsa difundida puede ser perjudicial, en vista que el Internet ofrece plataformas para la libertad de expresión y de pensamiento. Con el presente proyecto se busca evaluar la efectividad de las técnicas de IA para crear clasificadores automáticos de noticias falsas. Esto coadyuvará a que la información que es transmitida por las redes sociales pueda ser validada y determinar así su veracidad, con la finalidad de brindar información de calidad, es decir información contrastada y veraz.

3. METODOLOGÍA

En esta sección se presentan los aspectos relacionados con la delimitación de la investigación, el tipo de investigación desarrollada y los métodos que se utilizaron para su ejecución. Además, se describe la metodología usada para llevar a cabo el proceso de mapeo sistemático y las preguntas de investigación que se planteó investigar. Dichas preguntas permitieron determinar las variables e indicadores investigadas, y a la vez, permitieron definir los instrumentos de recolección de datos que se emplearon. Todos estos aspectos se describen en esta sección.

3.1. Delimitación de la investigación

La delimitación espacial del proyecto propuesto no se restringió a un espacio físico en particular. En el estudio de tipo mapeo sistemático se ha considera relevantes estudios internacionales indexados en bibliotecas digitales y bases de datos de impacto como: ACM, IEEE Xplore, Scopus, Springer y Wiley, Web of science. Esto implica que los estudios considerados pudieron haber sido publicados por investigadores de cualquier parte del mundo y la única restricción es que hayan sido escritos en inglés. Por otro lado, en relación con la delimitación temporal, se ha planteado que la investigación estuviera acotada a los últimos 5 años (2015-2020) ya que para un estudio de mapeo sistemático se sugiere contemplar estudios actuales de manera que se describa la realidad actual y los avances más recientes en el área estudiada.

3.2. Tipo de investigación

El tipo de estudio desarrollado es explicativo debido a que se estudió con un nivel de profundidad cómo operaban los mecanismos publicados en la literatura enfocados en la detección de noticias falsas que empleaban técnicas, modelos y algoritmos de Inteligencia Artificial.

Esta investigación también es consideró documental, debido a que toda la información que se obtuvo fue extraída de bases de datos bibliográficas que sirven como base para organizar la información y responder a ciertas preguntas de investigación de interés para la comunidad científica.

Además, la presente investigación se enmarcó en un paradigma metodológico mixto. Es mixto porque se emplearon dos enfoques, el cualitativo y cuantitativo; el primero se aplicó a través de una revisión bibliográfica que permitió determinar las técnicas de Inteligencia Artificial usadas en la detección de noticias falsas en las redes sociales. Además, en enfoque cuantitativo se aplicó para determinar los niveles de precisión que han alcanzado los modelos de clasificación automática basados en aprendizaje automático, aprendizaje profundo y minería de datos, para la detección de noticias falsas en las redes sociales.

3.3. Método de investigación

Para el logro de los objetivos planteados en la investigación se empleó el método inductivo, que parte de lo particular a lo general, con el propósito de determinar cuál eran los principales métodos usados por otros investigadores para detectar noticias falsas en las redes sociales y en función de ello determinar las técnicas de IA más comunes para ello. Ello ayudó a definir el aprendizaje automático, aprendizaje profundo y la minería de datos como las tres técnicas más usuales. usando la IA. De manera complementaria, se aplicó el método deductivo, que parte de lo general a lo particular, para determinar el alcance de las técnicas de IA que se han empleado en la detección de noticias falsas, y así determinar aquellos modelos que han tenido mayor precisión en la detección de noticias falsas.

También, el método ayudó a determinar los principales factores que determinan el buen o mal funcionamiento de clasificadores automáticos en la detección de noticias falsas. Además, se usó el método experimental puesto que en el estudio se llevó a cabo la manipulación de varias técnicas de aprendizaje en el cual se realizó pruebas modificando argumentos de los clasificadores lo cual ayudó en la ejecución de los métodos investigados y que, además, permitieron comparar los modelos estudiados. Finalmente, es importante señalar que también se utilizó el método descriptivo con el objetivo de describir las tres técnicas estudiadas para la detección de noticias falsas (aprendizaje automático, aprendizaje profundo, minería de dato) y sus modelos y algoritmos de clasificación (algoritmos de aprendizaje supervisado y redes neuronales).

3.4. Población y muestra

Se consideró como población los estudios recuperados, la búsqueda que se realizó en este estudio fueron trabajos disponibles en la literatura que han abordado la detección automática de noticias falsas en redes sociales usando tres técnicas específicas de la IA tales como: aprendizaje automático, aprendizaje profundo y minería de datos.

Para buscar y recuperar los estudios primarios en donde se aplicó la IA en la detección de noticias falsas se emplearon como fuente de información dos bibliotecas digitales (ACM, IEEE Xplore) y dos bases de datos documentales (Scopus, Web of Science, Springer y Wiley). Sobre estas fuentes de información, se aplicó una cadena de búsqueda especializada la cual se detalla como sigue: ((“machine” OR “deep”) AND (“learning”) AND (“fake new*”).

Una vez aplicada la cadena de búsqueda, y luego de aplicar criterios de exclusión que permitieran descartar estudios no relevantes (i.e., documentos repetidos, documentos escritos en un idioma diferente al inglés, documentos inaccesibles, y documentos publicados antes del 2015) se obtuvo como población (artículos sujetos a análisis) un total de 45 estudios (la forma en cómo se obtuvieron estos estudios se describe en el capítulo de Resultados). Estos estudios se describen de forma más detallada siguiendo el formato de la Tabla 7.

3.5. Variable e indicadores investigados

Dentro de la investigación se planteó tres espectros en relación con la detección de noticias falsas en redes sociales usando las técnicas de IA. La primera corresponde a los modelos de clasificación automática, la segunda corresponde a la eficacia de los clasificadores y la tercer a la eficiencia de dichos clasificadores. Ello ha permitido definir 4 variables con sus respectivos indicadores como se ilustra en la Tabla 3.

La primera variable, modelos de clasificación automática, tiene como finalidad recoger información cualitativa en relación con los modelos de aprendizaje automático, de aprendizaje profundo y de minería de datos usados para implementar clasificadores automáticos, que a partir de un proceso de entrenamiento se especializan en detectar noticias falsas. Para ello, se han definido seis indicadores que enfatizan principalmente en: el tipo de aprendizaje, algoritmos o modelos y set de datos usados.

La segunda variable, precisión que se encarga de evaluar la calidad de los modelos de aprendizaje y es de tipo cuantitativa, esta variable permitió decidir si los modelos tienen una adecuada tasa de precisión para clasificar las noticias falsas. Se definieron cuatro indicadores de los cuales son *accuracy*, *precisión*, *f1-measure*, *recall*. Para el tratamiento de estas medidas no se las ha traducido ya que, aunque tienen una traducción, no se suele usar los términos traducidos.

La tercera variable, rendimiento tiene como finalidad recoger valores cuantitativos sobre el desempeño de los modelos experimentados. Para ello, se han definido cuatro indicadores en los cuales están tiempo de respuesta, tiempo de entrenamiento, tiempo de prueba, tiempo de entrega de respuesta al usuario.

Finalmente, la cuarta variable, herramientas de desarrollo tiene como finalidad conocer con qué herramienta de software se trabajaron los investigadores para crear los modelos predictivos propuestos para la detección de noticias falsas. Se establecieron tres indicadores de los cuales son lenguaje de programación, librerías, versión.

3.6. Técnicas e instrumentos de recolección de datos

En esta investigación se adoptó la técnica de *Mapping Study* que se basa en el modelo propuesto por los autores Petersen et. al [84], que permitió realizar el análisis documental. A través de esta técnica se pudo identificar los estudios en los que se aplica técnicas de aprendizaje para la detección automática de noticias falsas en redes sociales. A continuación, se puede describir brevemente los pasos que siguieron para realizar el mapeo sistemático propuesto en este estudio.

1. **Definición de pregunta de investigación (alcance de la investigación).** El objetivo que llega a tener un mapeo sistemático es identificar cierta información que permitió tener una descripción general del tema. En lo cual se define ciertas preguntas que sirven para realizar el estudio [84].
2. **Realizar búsqueda de estudios primarios (todos los artículos).** Luego se aplican las cadenas de búsqueda en las bases de datos y esto permite obtener los estudios que sirven como base para el mapeo sistemático [84].

Tabla 3. Principales variables e indicadores sujeto a estudio

VARIABLES	INDICADORES	DESCRIPCIÓN	TIPO DE VARIABLE	ENTIDADES INVESTIGADAS
MODELOS DE CLASIFICACIÓN AUTOMÁTICA	Tipo de aprendizaje	El tipo de aprendizaje del algoritmo.	Cualitativa	Modelos predictivos Set de datos
	Tipo de algoritmo	El tipo de algoritmo usado en el modelo.	Cualitativa	
	Algoritmo de aprendizaje	Algoritmos utilizados.	Cualitativa	
	Fuente de datos	Conjunto de datos utilizado para el proyecto.	Cualitativa	
	Características	Características usadas para el análisis de texto.	Cualitativa	
	Etiqueta de salida	Las etiquetas para el análisis de texto.	Cualitativa	
PRECISIÓN	<i>Accuracy</i>	Porcentaje total de elementos clasificados correctamente.	Cuantitativa	Modelos predictivos
	<i>Precision</i>	Relación entre las observaciones positivas predichas correctamente y el total de observaciones positivas predichas.	Cuantitativa	
	<i>F1-measure</i>	Promedio ponderado de <i>precision</i> y <i>recall</i> .	Cuantitativa	
	<i>Recall</i>	Proporción de observaciones positivas predichas correctamente a todas las observaciones en la clase real.	Cuantitativa	
RENDIMIENTO	Tiempo de respuesta	El tiempo de respuesta del algoritmo.	Cuantitativa	Modelos predictivos
	Tiempo de entrenamiento	El tiempo que toma el modelo en realizar el entrenamiento de los datos.	Cuantitativa	
	Tiempo de prueba	El tiempo que toma el modelo en realizar el entrenamiento de la prueba de los datos previamente entrenados.	Cuantitativa	
	Tiempo de entrega de respuesta al usuario	El tiempo que toma en mostrar los resultados obtenidos al usuario.	Cuantitativa	
HERRAMIENTAS DE DESARROLLO	Lenguaje de programación	Lenguaje de programación utilizado para el desarrollar el proyecto	Cualitativa	Sistemas de clasificación
	Librería/marco de trabajo/API	Librerías usadas para desarrollar el clasificador automático	Cualitativa	
	Herramientas de apoyo	Herramientas complementarias a las de IA usadas para el desarrollo del sistema y clasificador automático	Cualitativa	

3. **Selección de artículos para inclusión y exclusión (documentos relevantes).** El tercer paso es aplicar ciertos criterios tanto de inclusión como exclusión para obtener los estudios más relevantes obtenidos en la sección anterior y que permitan resolver las preguntas definidas [84].
4. **Palabras clave de los resúmenes (esquema de clasificación).** Aquí los revisores leen los resúmenes y con esto buscar las palabras claves que permita conocer el contexto del artículo para luego crear categorías, en caso de que los resúmenes no brinden mucha información se opta por leer o revisar las otras partes de los artículos como introducción o conclusión. Luego que se defina se puede utilizar esas categorías para la investigación [84].
5. **Extracción de datos y mapeo de estudios (mapa sistemático).** En esta sección se utiliza tablas que tiene las categorías definidas y se agrupan los artículos con una breve explicación [84].

Es importante señalar que para el desarrollo del estudio no se empleó un instrumento específico de recolección de datos. El instrumento empleado constó básicamente de un conjunto de preguntas a las que el estudio respondió de forma detallada. Dichas preguntas son descritas en la sección de resultados y fueron definidas con el apoyo de un experto.

3.7. Técnicas de procesamiento y análisis de datos

El análisis de datos empleado por el mapeo sistemático fue en su mayoría cualitativo. Dicho análisis consistió en responder, a partir de la lectura de los estudios recuperados, las siete preguntas de investigación previamente descritas. Es importante señalar que el análisis de los estudios también se realizó por el tipo de estudio, autores, fecha, título y país; complementariamente al análisis hecho por cada pregunta de investigación (RQ).

Por otro lado, en vista de que en el estudio se planteó también evaluar las tres técnicas de IA más empleadas para la detección automática de noticias falsas en medios sociales, se empleó también en análisis cualitativo de datos. Para ello, se usó como herramienta de apoyo Microsoft Excel. Se procesó principalmente métricas de exactitud y de rendimiento de los clasificadores. Adicionalmente a las tablas, la técnica de visualización de datos tabulados graficados también se empleó como mecanismo de comparación.

3.8. Normas éticas

El presente proyecto se realizó de acuerdo con los lineamientos establecidos en el Reglamento de Grados de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas. Además, es importante señalar que en todas las instancias del estudio se respetaron los derechos de autor dándole el reconocimiento que se merecen por las ideas, conceptos y modelos de Inteligencia Artificial que se analizaron en esta investigación. Se ha citado de manera correcta los estudios luego de cada idea de terceros en el estudio. También, con el fin de no caer en la piratería se utilizó software libre para realizar la experimentación planteada en el estudio.

4. RESULTADOS

4.1 Descripción del estudio de mapeo sistemático

Una revisión sistemática consiste en identificar, evaluar e interpretar los estudios más relevantes de un tema. En esta investigación, se realizó el proceso de revisión sistemática propuesto por Petersen et. al [84] para el desarrollo de revisión de literatura. Este proceso consiste en definir las preguntas de investigación, luego realizar búsqueda de estudios primarios, después buscar palabras clave de los resúmenes para con eso elegir las categorías, y por último se realizó extracción de datos y mapeo de estudios, y la agrupación de los estudios dependiendo de la categoría.

4.1.1. Preguntas de investigación

Se preparó las preguntas de investigación (RQ1 a RQ6), todas ellas enfocadas en determinar cómo las técnicas de aprendizaje automático aprendizaje profundo y minería de texto, han sido empleadas para crear sistemas automáticos de clasificación para detectar noticias falsas en medios sociales.

Las siete preguntas formuladas se escriben como sigue:

- **RQ1.** ¿Cuál es el rol que han desempeñado las técnicas de aprendizaje de IA en la detección de noticias falsas en medios sociales?
- **RQ2.** ¿Cuáles han sido las principales propuestas tecnológicas basadas en las técnicas de aprendizaje de IA para la detección de noticias falsas en redes sociales?
- **RQ3.** ¿En qué ámbitos se han empleado las técnicas de aprendizaje de la IA para la detección de noticias falsas? (i.e., política, ciencia, etc.)
- **RQ4:** ¿Cuál es el alcance que han tenido los modelos de ML y DL en términos de precisión, para la detección de noticias falsas?
- **RQ5.** ¿Qué herramientas de software se emplearon para crear los modelos predictivos de detección de noticias falsas?
- **RQ6.** ¿En qué medios sociales se basaron las investigaciones, o de qué medios sociales se obtuvo los datos para entrenar los modelos propuestos?

4.1.2. Definición de estrategia de búsqueda

La búsqueda realizada en esta revisión consideró trabajos disponibles en la literatura sobre los estudios que han abordado la detección automática de noticias falsas en redes sociales desde una perspectiva teórica y desde una óptima experimental o aplicada. La estrategia de búsqueda se divide en tres pasos: definición de fuentes de búsqueda, definición de términos de búsqueda y selección de estudios y extracción de datos.

4.1.2.1. Definición de fuentes de búsqueda

El primer paso de la estrategia de búsqueda fue identificar y definir qué fuentes de búsqueda se emplearían para realizar la revisión sistemática. Para el caso particular de esta investigación, se seleccionaron cuatro bases de datos bibliográficas digitales (ACM, IEEE Xplore) y dos bases de datos documentales (Scopus y Web of Science, Springer, Wiley). En cada una se realizaron considerando los campos de título y artículos publicados entre los años 2015 a 2020. Ambos tipos de fuentes de información especializadas en el campo de las ciencias computacionales. Los detalles para el acceso a dichas fuentes se observan en la Tabla 4.

Tabla 4. Bases de datos científicas y librerías digitales empleadas.

CÓDIGO	FUENTE DE INFORMACIÓN	DE URL
BD1	Scopus	https://www.scopus.com/
BD2	ACM	https://dl.acm.org/
BD3	WoS	https://idp.fecyt.es/
BD4	IEEE	https://ieeexplore.ieee.org/
BD5	Springer	https://www.springer.com/la
BD6	Wiley	https://onlinelibrary.wiley.com/

4.1.2.2. Estrategia de búsqueda

Para realizar la búsqueda de estudios primarios, a partir de los cuales se responden las preguntas de investigación (RQ), se formuló una cadena de búsqueda la cual fue adaptada a cada una de las fuentes de información usadas. Su correspondiente adaptación para cada una de las plataformas que sirvieron como fuente de información y que fueron sujetas a estudio se muestra en la Tabla 5.

Tabla 5. Cadenas de búsqueda

FUENTE	CADENA DE BÚSQUEDA
IEEE	((("Document Title": machine) OR "Document Title": deep) AND "Document Title": learning) AND "Document Title": fake news)
ACM	[[Publication Title: machine] OR [Publication Title: deep]] AND [All: learning] AND [Publication Title: fake news] AND [Publication Date: (01/01/2015 TO 10/31/2020)]
WILEY	"(machine OR deep)" in Title and "learning" in Title and "fake news" in Title
SPRINGER	(TITLE (machine) OR TITLE (deep) AND TITLE (learning) AND TITLE (fake AND news))
SCOPUS	(TITLE (machine) OR TITLE (deep) AND TITLE (learning) AND TITLE (fake AND news))
WOS	(TI= (machine OR deep) AND TI= (learning) AND TI= (fake news)) AND IDIOMA: (English)

El proceso de búsqueda se aplicó sobre las bases de datos de la Tabla 5 restringiendo la búsqueda a los términos de las cadenas formuladas, aplicándola al título, palabras claves y resumen. Además, se consideró solo los estudios publicados entre el periodo 2015-2020. Este período seleccionado (cinco años) es un período de tiempo prudente para evitar la selección de trabajos obsoletos. Los resultados (artículos recuperados) tras aplicar la cadena de búsqueda definida en las fuentes de información sujetas a estudio se muestran en la Tabla 6.

Tabla 6. Cantidad de artículos brutos recuperados en las fuentes de información empleadas.

Fuente	Número de artículos
IEEE	26
ACM	36
WILEY	1
SPRINGER	214
SCOPUS	63
WOS	6

4.1.2.3. Criterio de inclusión y exclusión

Los criterios de inclusión se refieren primero a los estudios que analicen y empleen mecanismos para detectar una noticia falsa en redes sociales con inteligencia artificial. De manera complementaria, los criterios de exclusión se refieren a los documentos

repetidos, documentos escritos en un idioma que no sea inglés o español, documentos en otras áreas o documentos no disponibles, documentos publicados antes del 2010.

4.1.2.4. Conducta de la búsqueda

Para poder seleccionar los estudios se realizó dos etapas; la primera consistió en dar lectura al título del artículo y al resumen y a las conclusiones de cada artículo. Luego de determinar que el artículo corresponde al tema damos paso a la siguiente etapa que consiste en la lectura del artículo completo para luego analizarlo.

4.2. Estudios analizados en el mapeo sistemático

El proceso de búsqueda se la hizo en cada una de las bases de datos definidas anteriormente. Los estudios están enfocados en proponer modelos para detectar una noticia falsa, de los 346 artículos que se encontraron; 311 fueron no repetidos. La segunda fase que consistió en aplicar los criterios de exclusión y inclusión basado en no disponibles en la cual se excluyeron 27, por lo que quedaron 284. En la fase 3 redujeron a 70 luego de aplicar los criterios de inclusión y exclusión basados en el título y descartar 214. Pero se redujeron a 67 debido a que 3 libros eran libros. En la fase 4 se aplicó los criterios de inclusión y exclusión basados en el texto completo de los documentos, así, se excluyeron 22 artículos. Un resumen de estos datos se muestra en la Figura 20.

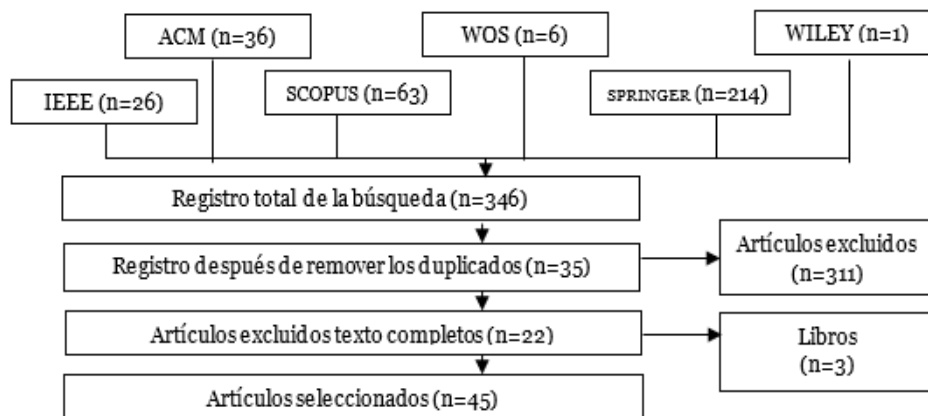


Figura 20. Proceso para seleccionar los estudios sujetos a análisis.

De manera más detallada, a continuación, en la Tabla 7, se presenta la lista de los estudios que fueron leídos de manera integral, y a partir de los cuales se han planteado las respuestas a las preguntas de investigación (RQs) formuladas en la investigación.

Tabla 7. Estudios orientados a detectar noticias falsas

#	Título del artículo	Autor/es	País	Año	Ref.
S4	A Deep Transfer Learning Approach for Fake News Detection	Saikh, Tanik Haripriya, B. Ekbal, Asif Bhattacharyya, Pushpak	India	2020	[85]
S5	Identification of Fake News Using Machine Learning	Mandical, Rahul R. Mamatha, N. Shivakumar, N. Monica, R. Krishna, A. N.	India	2020	[86]
S6	Performance Comparison of Machine Learning Classifiers for Fake News Detection	Smitha, N. Bharath, R.	India	2020	[87]
S7	Not Everything You Read Is True! Fake News Detection using Machine learning Algorithms	Tiwari, Vanya Lennon, Ruth G. Dowling, Thomas	Irlanda	2020	[88]
S8	Fake News Detection using Deep Learning	Kong, Sheng How Tan, Li Mei Gan, Keng Hoon Samsudin, Nur Hana	Malasia	2020	[89]
S9	Fake news detection using deep learning models: A novel approach	Kumar, Sachin Asthana, Rohan Upadhyay, Shashwat Upreti, Nidhi Akbar, Mohammad	India	2020	[90]
S12	Fake news stance detection using deep learning architecture (CNN-LSTM)	Umer, Muhammad Imtiaz, Zainab Ullah, Saleem Mehmood, Arif Choi, Gyu Sang On, Byung Won	Pakistán	2020	[36]

S15	DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning	Popat, Kashyap Mukherjee, Subhabrata Yates, Andrew Weikum, Gerhard	USA	2018	[91]
S16	Fake News Detection Using Machine Learning Ensemble Methods	Ahmad, Iftikhar Yousaf, Muhammad Yousaf, Suhail Ahmad, Muhammad Ovais	Pakistán	2020	[92]
S17	Integrating Machine Learning Techniques in Semantic Fake News Detection	Adrian M. P. Brasoveanu, Razvan Andonie	Austria	2020	[93]
S19	An empirical comparison of fake news detection using different machine learning algorithms	Albahr, Abdulaziz Albahar, Marwan	Arabia Saudita	2020	[94]
S20	A predictive model based on machine learning methods to recognize fake persian news on twitter	Sadr Mohammad Mohsen, Chelak Afshin Mousavi, Ziaei Soraya, Tanha Jafar	Irán	2020	[95]
S24	A machine learning approach to fake news detection using knowledge verification and natural language processing	Ibrishimova, Marina Danchovsky Li, Kin Fun	Canadá	2020	[96]
S25	A Location Independent Machine Learning Approach for Early Fake News Detection	Ibrishimova, Marina Danchovsky Li, Kin Fun	Singapur	2020	[96]
S26	Multiclass Fake News Detection using Ensemble Machine Learning	Kaliyar, Rohit Kumar Goswami, Anurag Narang, Pratik	India	2019	[24]
S27	Fake News Detection in Social Networks Using Machine Learning and Deep Learning: Performance Evaluation	Wenlin Han, Varshil Mehta	USA	2019	[97]
S28	Deep learning methods for Fake News detection	Kresnakova, Viera Maslej Sarnovsky, Martin Butka, Peter	Eslovaquia	2019	[98]

S29	Merging deep learning model for fake news detection	Amine, Belhakimi Mohamed Drif, Ahlem Giordano, Silvia	Argelia	2019	[99]
S30	Pakistani Media Fake News Classification using Machine Learning Classifiers	Kareem, Irfan Awan, Shahid Mahmood	Pakistan	2019	[34]
S31	A closer look at fake news detection: A deep learning perspective	Abedalla, Ayat Al-Sadi, Aisha Abdullah, Malak	Jordania	2019	[100]
S32	Which machine learning paradigm for fake news DETECTION?	Katsaros Dimitrios, Stavropoulos George, Bridge Ivy	Chipre	2019	[101]
S33	Automatic Identification of Fake News Using Deep Learning	Qawasmeh, Ethar Tawalbeh, Mais Abdullah, Malak	Jordania	2019	[25]
S34	Fakebuster: Fake news detection system using logistic regression technique in machine learning	Mokhtar, Muhammad Syahmi Jusoh, Yusmadi Yah Admodisastro, Novia Pa, Norainiche Amruddin, Amru Yusrin	Malasia	2019	[102]
S35	A smart System for Fake News Detection Using Machine Learning	Jain, Anjali Shakya, Avinash Khatler, Harsh Gupta, Amit Kumar	India	2019	[103]
S36	Fake news detection of Indian and United States election data using machine learning algorithm	Kumar, Anant Singh, Satwinder Kaur, Gurpreet	India	2019	[104]
S37	FIND: Fake Information and News Detections using Deep Learning	Verma, Abhishek Mittal, Vanshika Dawn, Suma	India	2019	[105]

S38	An Efficient Fake News Detection System Using Machine Learning	A. Lakshmanarao, Y. Swathi, T. Srinivasa Ravi Kiran	India	2019	[106]
S40	Combining Similarity Features and Deep Representation Learning for Stance Detection in the Context of Checking Fake News	Luís Borges, Bruno Martins, Pável Calado	Portugal	2019	[107]
S41	natural language contents evaluation system for detecting fake news using Deep learning	Ye-Chan Ahn, Chang-Sung Jeong	Corea del Sur	2019	[108]
S42	F-NAD: An Application for Fake News Article Detection using Machine Learning Techniques	Ranojoy Barua, Rajdeep Maity, Dipankar Minj, Tarang Barua, Ashish Kumar Layek,	India	2019	[109]
S44	Detecting Fake News using Machine Learning and Deep Learning Algorithms	Abdullah-All-Tanvir Mahir, Ehasas Mia Akhter, Saima Huq, Mohammad Rezwanul	Bangladés	2019	[110]
S46	Comparison of Various Machine Learning Models for Accurate Detection of Fake News	Poddar, Karishnu Amali, Geraldine Bessie D. Umadevi, K. S.	India	2019	[111]
S47	Fake news detection using machine learning and natural language processing	Agarwalla, Kushal Nandan, Shubham Nair, Varun Anil Deva Hema, D.	India	2019	[112]
S48	Deep learning algorithms for detecting fake news in online text	Girgis, Sherry, Eslam Amer Gadallah, Mahmoud	Egipto	2019	[113]
S49	Detecting fake news with machine learning method	Supanya Aphiwongsophon, Prabhas Chingsitvatana	Tailandia	2019	[114]

S50	Fake News Detection Using Deep Learning	Dong-Ho Lee, Yu-Ri Kim, Hyeong-Jun Kim, Seung-Myun Park, y Yu-Jun Yang	Corea del Sur	2019	[115]
S53	A Comparison of Machine Learning Algorithms in Fake News Detection	Ahmad, Faraz Lokeshkumar, R.	India	2019	[116]
S55	Comparative Performance of Machine Learning Algorithms for Fake News Detection	Arvinder Pal Singh Bali, Mexson Fernandes, Sourabh Choubey, y Mahima Goel	India	2019	[117]
S57	Fake News Detection Using Machine Learning Ensemble Methods	Leung, Potsane Mohale and Wai Sze	Sudáfrica	2019	[92]
S58	Semantic Fake News Detection: A Machine Learning Perspective	Adrian M. P. Bra, soveanu y R˘azvan Andonie	Romania	2019	[118]
S55	Comparative Performance of Machine Learning Algorithms for Fake News Detection	Arvinder Pal Singh Bali, Mexson Fernandes, Sourabh Choubey, and Mahima Goel	India	2019	[119]
S60	Evaluating Machine Learning Algorithms for Fake News Detection	Shlok Gilda	India	2018	[120]
S61	The fake news challenge: Stance detection using traditional machine learning approaches	Razan Masood, Ahmet Aker	Alemania	2016	[121]
S62	Raising a Model for Fake News Detection Using Machine Learning in Python	Gerardo Ernesto Rolong Agudelo, Octavio José Salcedo Parra, and Julio Barón Velandia	Colombia	2018	[122]
S63	Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques	Hadeer Ahmed, Issa Traore, and Sherif Saad	Canadá	2017	[123]

A continuación, se describen los resultados de los estudios primarios seleccionados. Como se muestra en la Figura 21, la evolución en el tiempo de las propuestas encontradas. Se observa gran interés creciente en las propuestas de detección automática de noticias falsas en los últimos años. En 2019 y 2020 se encontraron el 89% del total de las propuestas identificadas en el mapeo.

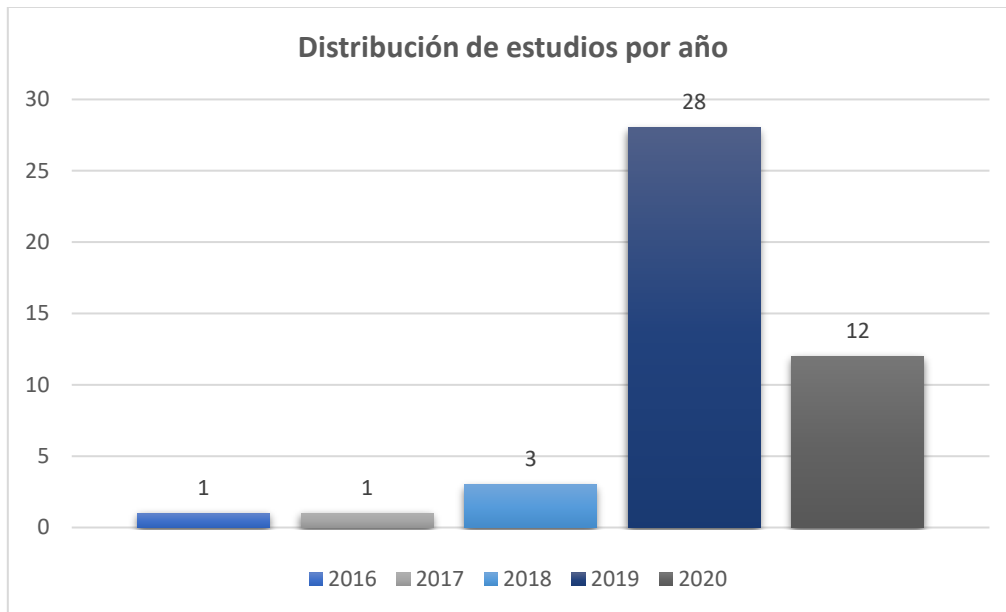


Figura 21. Distribución de estudios primarios por año

En segundo lugar, se muestran en la Figura 22 la distribución de los estudios según el año en que fueron publicados, esto demuestra que la mayoría de los trabajos provienen de India con 16, seguido Pakistán con 3 que estuvieron radicados en el 2019.

4.3. Respuesta a las preguntas de investigación

A continuación, se responden cada una de las preguntas de investigación formuladas, enfatizando en las propuestas de los estudios de la literatura y discutiendo acorde a la lógica de cada pregunta.

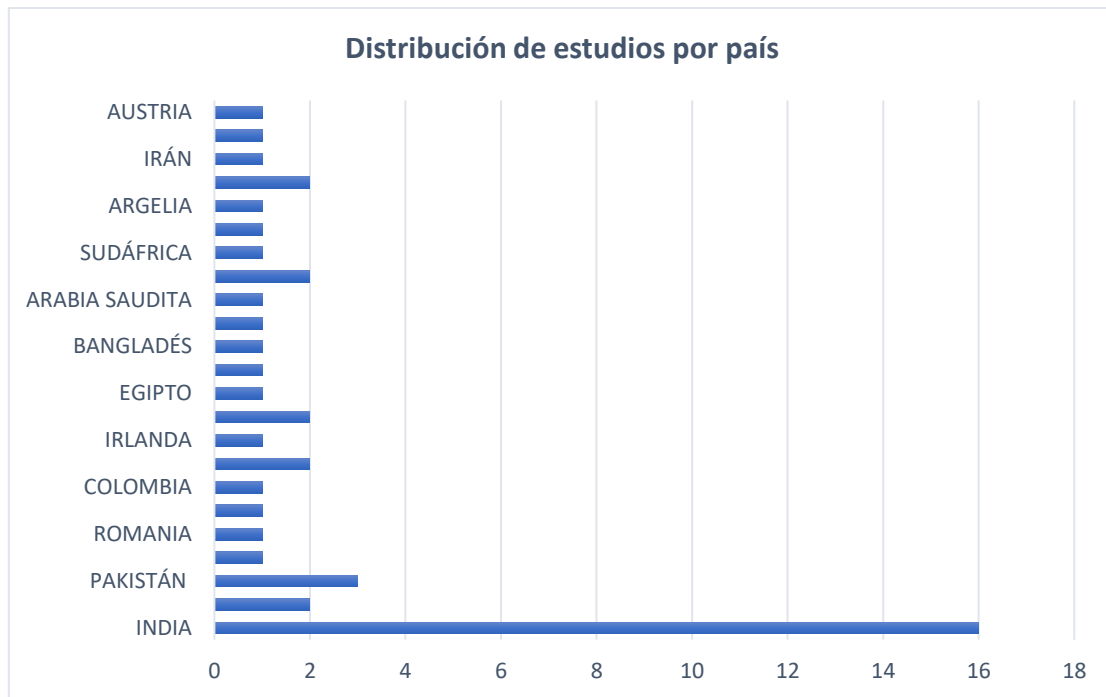


Figura 22. Distribución de estudios primarios por país

RQ1: ¿Cuál es el rol que han desempeñado las técnicas de aprendizaje de IA en la detección de noticias falsas en medios sociales?

El aprendizaje automático para la detección de noticias falsas tiene varios enfoques con respecto a los diferentes clasificadores ya que partiendo de noticias reales o falsas permiten crear algoritmos de aprendizaje supervisados que logran aprender de los datos y con esto llevan a cabo el proceso para la clasificación automática de noticias falsas.

Estos clasificadores son integrados generalmente en sistemas de manera que faciliten la verificación de noticias falsas; por lo tanto, los algoritmos pertenecientes a esta categoría fueron de los primeros cuya eficacia se ha investigado, los algoritmos de aprendizaje automático que se han examinado en el contexto de noticias falsas figuran: los árboles de decisión, bosques aleatorios, regresión logística y máquinas de vectores de soporte.

Por otro lado, para aprendizaje profundo tiene a implementar modelos híbridos usando la combinación de dos redes neuronales. En esta categoría de aprendizaje se han empleado modelos que pertenecen a la categoría de: redes convolucionales, memorias a corto plazo, unidades recurrentes cerrada, perceptrón multicapas, y vainilla.

Finalmente, en lo que respecta a la minería de datos, esta técnica no ha sido ampliamente usada en los últimos años para la detección de noticias falsas.

Como se muestra en la Figura 23, los artículos categorizados en las 3 áreas antes comentadas se han empleado de la siguiente forma: aprendizaje automático con 14 estudios que corresponde al 31%, uso de modelos de aprendizaje profundo con 14 estudios con 31% y ambas, combinadas para optimizar los modelos y algoritmos genéricos, que con 17 estudios que respecta al 38%.

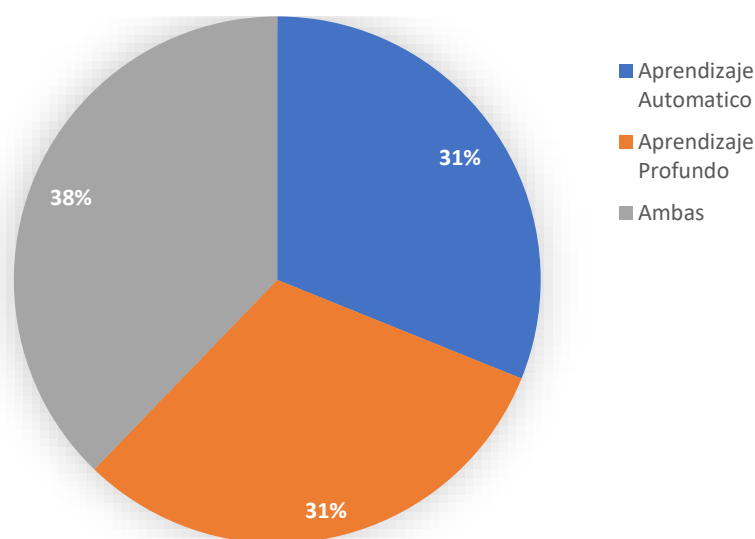


Figura 23. Gráficos de las técnicas usadas.

La mayoría de los estudios están empleando las dos técnicas de aprendizaje más populares hoy en día, de forma combinada se está buscando crear clasificadores más precisos para a futuro implementar dichos modelos en las redes sociales, y así, combatir el fenómeno de las redes sociales, que aqueja hoy en día por el masivo uso de redes sociales por cualquier usuario, que muchas veces comparten contenidos sin previa verificación.

RQ2: ¿Cuáles han sido las principales propuestas tecnológicas basadas en las técnicas de aprendizaje de IA para la detección de noticias falsas en redes sociales?

Aprendizaje automático y la detección de noticias falsas. En la Tabla 8 se muestra la clasificación de categorías referente a los algoritmos de aprendizaje automático publicados en la literatura para crear sistemas de detección de noticias falsas.

Tabla 8. Algoritmo de aprendizaje automático para clasificación de noticias falsas.

Categoría	Artículos	Cantidad	Porcentaje
NB	S17, S19, S26, S30, S39, S44, S46, S47, S49, S53, S57, S62	12	11%
DT	S5, S6, S7, S16, S17, S19, S20, S26, S27, S30, S32, S38, S46, S58, S60, S63	16	15%
RF	S6, S7, S16, S17, S19, S20, S26, S30, S32, S38, S39, S55, S58, S60,61	16	15%
KNN	S7, S16, S20, S27, S30, S38, S55	7	6%
SVM	S6, S9, S16 S17, S20, S27, S32, S35, S36, S38, S39, S44, S46, S47, S49, S53, S55, S57, S58, S60, S63	23	21%
LR	S6, S7, S9, S16, S17, S26, S30, S32, S34, S39, S44, S46, S47, S58, S61 S63	17	16%
GB	S6, S26, S27, S53, S55, S60	6	5%
AUC	S55	1	1%
SGB	S17, S27, S57, S60, S63	5	5%
AB	S16	1	1%
PASIVO-AGRESIVO	S5	1	1%
XG-BOOST	S6, S16	2	2%
BASELINE	S27	1	1%

A continuación, se describe brevemente cómo los principales clasificadores de noticias falsas han empleado los algoritmos de aprendizaje automático para implementar clasificadores que aprenden a partir de datos.

Naïve Bayes (NB) y la detección de noticias falsas

En la categoría NB se plantearon 12 estudios (11%). En el artículo 19 usaron el conjunto de datos de liar, el cual tiene información política y contiene 12,836 declaraciones, agregaron algunas otras funciones para detectar las noticias. Eliminaron la puntuación de las oraciones de la declaración. Luego eliminaros los caracteres repetidos; también limpiaron hipervínculos y otros caracteres especiales del texto de las noticias de declaraciones. Evaluaron el desempeño de cada modelo con validación cruzada de K-fold con 5 para Naïve Bayes en las 5 se obtuvo 99% de exactitud.

Por otra parte, en el artículo 26 utilizaron el modelo multinomial y configuraron el parámetro con alfa en 1.0 obteniendo un 72% en exactitud. Para el artículo 30 utilizaron el alfa al 0,70 y obtuvieron un porcentaje de 72%. Así mismo en el artículo 47 los autores trabajaron NB y con un conjunto de datos obtenido de Kaggle este contiene 4008 filas con 4 atributos (url, titulo, cuerpo y etiqueta), de los cuales 2136 son artículos de noticias falsas y 1872 de noticias reales. NLTK en Python se utilizó para tokenizar el cuerpo y el

título y la eliminación de las palabras vacías. Además, los investigadores crearon una nube de palabras para el título y el cuerpo presentes en su conjunto de datos. Usando NB con suavizado de Lidstone obtuvieron con 82% de precisión.

En el artículo 62 se utilizó un conjunto de datos de Kaggle con un total de 10558 noticias recopiladas en total entre los años 2015 y 2017 escritas en inglés con su título, texto completo y etiqueta falsa o verdadera que fueron extraídas de diferentes medios. Para la extracción de características emplearon el método TF-IDF, *count vectorizer*. Los resultados obtenidos fueron que NB con TF-IDF 84.8 y *count vectorizer* obtuvo 88.10 % que tuvo la mayor exactitud.

En el resto de los estudios se basaron en las mismas bases ya descritas, sin embargo, es importante señalar, que NB tiende a tener mayor porcentaje cuando los datos son pocos y también se limpian los datos y se usan validaciones cruzadas.

Árbol de decisión (DT) y la detección de noticias falsas

En la segunda categoría DT con 16 estudios (15%) figura el artículo 26 en el que los investigadores trabajaron con un conjunto de datos obtenidos de Kaggle para enviar como entrada al modelo de árbol de decisión. Se configuró el parámetro profundidad máxima (*maximum depth*) con 5 y se obtuvo un 74% de exactitud. También en el artículo 30 lo usaron; pero los parámetros configurados fueron, el criterio entropía y estado aleatorio en 0 en la que obtuvieron 64% de exactitud.

Random Forest (RF) y la detección de noticias falsas

En la categoría RF con 16 estudios (15%) en el artículo 19 para el modelo configuraron los parámetros como: *n_estimators*, *min_samples_split*, *random_state*, *max_depth* para obtener el mejor rendimiento, esto es, un 93% de exactitud.

Por otro lado en el artículo 16 usaron 4 conjuntos de datos del cual el primero tiene un total de 44.898 artículos, el segundo total de 20.386 artículos, el tercero 3.352 artículos y el último, la combinación para el procesamiento se eliminaron los artículos sin cuerpo de texto o con menos de 20 palabras, también los artículos de varias columnas se transforman en artículos de una sola columna para lograr uniformidad de formato, estas operaciones se realizaron en todos los conjuntos de datos en el cuerpo del artículo. Para configurar el

algoritmo RF se utilizó el índice de Gini como una función de costo para estimar una división en el conjunto de datos en la cual se obtuvo un porcentaje de 99% exactitud. Este porcentaje se obtuvo con el primer conjunto de datos y se vio que entre el conjunto de datos sea más grande el porcentaje aumentaba.

En el artículo 26 trabajaron con un conjunto de datos de Kaggle en las funciones utilizaron TF-IDF con el modelo RF y utilizaron el parámetro de 10 estimadores obteniendo un puntaje de 70% en exactitud. Con *count vector* y TF-IDF y realizando limpieza en los datos el porcentaje de exactitud tiende a ser igual como el artículo 6. Otra opción para obtener un puntaje alto sería probando con la herramienta LIWC2015 que clasifica el texto en diferentes variables discretas y continuas.

Vecino más cercano (KNN) y la detección de noticias falsas

En la categoría KNN con 7 estudios que representa el 6% de la cual el artículo 30 que usaron vecinos más cercanos han utilizado con valores predeterminados los cuales fueron $n_neighbor = 5$, $weights = 'uniform'$, $metric = 'minkowski'$, p con 2 y obtuvieron 70% de exactitud. Para el resto fue lo mismo; pero se probó que con 3 estrategias de características TF-IDF, Hashing y TF de la cual se obtuvo el resultado más alto con Hashing en el artículo 7. Para el artículo 16 que se probó con 4 conjuntos de datos; el primero contiene 44.898 artículos, el segundo contiene un total de 20.386 artículos, el tercero incluye un total de 3.352 artículos y el cuarto que es una combinación de los tres conjuntos este se creó para evaluar el rendimiento de los algoritmos. La exactitud más alta se obtuvo con el primer conjunto de datos con un 98%.

Máquina de vectores de soporte (SVM) y la detección de noticias falsas

Otro algoritmo relevante que fue usado en los estudios fue SVM con 23 estudios (21%) en la que en el artículo 30 se trabajó con el modelo SVM configurando dicho algoritmo con propiedades de núcleo lineal y núcleo sigmoide con estado aleatorio cero en la que obtuvieron un 68 % de exactitud. Por otra parte, en el artículo 26 los autores trabajaron con un conjunto de datos de Kaggle en las funciones utilizaron TF-IDF y el clasificador SVM obtuvo un 76%.

Regresión logística (LR) y la detección de noticias falsas

En la siguiente categoría LR con 17 estudios (16%) de la cual el artículo 47 los autores trabajaron con un conjunto de datos de noticias que fueron extraídas de Twitter que cuenta con 20,360 datos y que contiene información sobre los terremotos de Chile de 2010. Para procesar los datos usaron *numpy* y *scikit-learn* y bibliotecas de *keras* y *matplotlib* y como es texto el procesamiento lo realizaron con TF-IDF y *count vectors*. Para este modelo se obtuvo un porcentaje de 68% de exactitud.

Por parte del artículo 26 que usaron un conjunto de datos obtenido de Kaggle configurando el parámetro $C=1e5$ de la que obtuvieron un 68% de exactitud. Este algoritmo tiende a tener mayor exactitud cuando usa con grandes conjuntos de datos y se aplica la extracción TF-IDF.

Potenciación adaptativa (GB) y la detección de noticias falsas

Con respecto a la categoría GB con 6 estudios (7%) en el artículo 26 los autores trabajaron con un conjunto de datos de Kaggle. En las funciones utilizaron TF-IDF y los clasificadores fueron RF, NB multinomial, GB, SVM LINEAL, DT, LR. Los resultados obtenidos alcanzaron una precisión del 86% con el modelo de potenciación adaptativa, esto demuestra su idoneidad para un problema de clasificación textual de clases múltiples.

AUC y la detección de noticias falsas

En cuanto a la categoría AUC con un estudio (1%), el artículo 55 usó 3 conjuntos de datos, luego se realizó la limpieza del conjunto de datos y el texto del título y el cuerpo se tokenizaron, se derivó y se pasó por el modelo n-gram para hacer las pruebas en la cual dividieron los datos en 80% para entrenamiento y el 20% pruebas.

XGBoost y la detección de noticias falsas

Por lo que se refiere a la categoría XG-BOOST con 2 estudios (2%) entre ellos está el artículo 25 en la que trabajaron con un conjunto de datos de código abierto, compuesto por 5060 artículos de noticias extraídos de una lista seleccionada de 1001 dominios de noticias de Estados Unidos y Singapur. El modelo propuesto se usó *count vectorizer*, TF-IDF y *glove* en la que se dieron 2 etapas. Para la primera etapa se aplicó random forest,

XGBoost, AB, GB, LR todos con 500 estimadores. Luego los modelos de primer nivel generan una probabilidad de que una noticia sea falsa y un modelo de segundo nivel que es XGBoost. Se hace una clasificación binaria final basada en las probabilidades generadas obteniendo 3 porcentaje para cada modelo de procesamiento de características, para TF-IDF fue 96.70 % para *count-vectorizer* 95.2% y 300-dimensional GloVe con 90.5% de precisión.

Descenso estocástico de gradiente y la detección de noticias falsas

En el artículo 60 [120] los autores para realizar su experimento utilizaron el conjunto de datos llamado *signal media* que lo obtuvieron de una variedad de fuentes de noticias de septiembre 2015, realizando una limpieza tuvieron un total de 11051 y etiquetaron las fuentes confiables con 0 y no confiables como 1. Para el preprocesamiento eliminaron los artículos que mencionó el nombre de la fuente y también los identificadores de Twitter y las direcciones de correo electrónico. En los conjuntos de características utilizaron TF-IDF de bi-grama, Frecuencia de dependencia sintáctica normalizada PCFG y la tercera fue la unión de las 2, para generar estas características usaron *spacy* Python para realizar la tokenización. En los resultados obtenidos demostraron que TF-IDF de bi-gramas introducidos en el modelo de SGB tiene mayor precisión con 77,2%.

Pasivo-agresivo y la detección de noticias falsas

En el siguiente punto trata de la categoría pasivo-agresivo con un estudio (1%) de la cual en el artículo 5 [86] los autores usaron algunos conjuntos de datos con diferentes atributos. Para la limpieza de los datos se eliminaron ciertos atributos que no se consideraron relevantes y para la extracción de características usaron TF-IDF; luego se introdujo en los modelos basados en el clasificador Naïve Bayes y el clasificador pasivo-agresivo y DNN y usaron Keras. Los resultados que obtuvieron en este estudio fue que pasivo-agresivo tuvo una precisión del 100% en las pruebas.

Baseline y la detección de noticias falsas

El resto de los estudios están en la categoría baseline con 1 estudio de 1% de la cual en el artículo 27 trabajaron con un conjunto de datos que incluyen 5.800 tweets. Los tweets

fueron etiquetados como no rumores y rumores, luego llevaron a cabo la tokenización, el reconocimiento de entidades con nombre. Para la extracción de características se usaron 3, esto es, frecuencia normalizada de dependencias sintácticas analizadas, frecuencia de términos bi-gramas y frecuencia inversa de documentos (TF-IDF), y finalmente, la unión de las dos primeras. Además, trabajaron con aprendizaje automático BOUND DT, GB, SGD, SVM y aprendizaje profundo con LSTM, LSTM DROP y enfoque híbrido LSTM Y CNN. El resultado fue el mismo para las 3 extracciones de características con 67,89% de exactitud. El que mejor se desempeñó por parte de aprendizaje automático fue SGD con 77% utilizando frecuencia de términos bi-gramas.

Como se ha descrito anteriormente los modelos implementados no solo siguen una misma estructura; estos tienden a variar en términos del conjunto de datos usado, el mecanismo de limpieza de los datos usados, las técnicas de extracción que aplican, y los parámetros de los clasificadores usados; siendo el que más sobresale la máquina de vectores de soporte con 23 estudios. Cabe recalcar que para estas técnicas los parámetros en los clasificadores no aplican para todos, lo que lleva a que por esta parte no sean muy comparativos. Sin embargo, el porcentaje de exactitud varió según las técnicas usadas.

Aprendizaje profundo y la detección de noticias falsa. En lo referente al uso de aprendizaje profundo para la detección de noticias falsas en medios sociales, en la Tabla 9, se listan los modelos de redes neuronales profundas que hasta la fecha se han empleado para desarrollar este tipo de clasificadores. A continuación, se describe brevemente cómo los modelos de redes neuronales profundas se han integrado en sistemas de detección de noticias falsas en medios sociales.

Tabla 9. Clasificación de artículos con aprendizaje automático

CATEGORÍA	CÓDIGO DE LOS ARTÍCULOS	CANTIDAD DE ARTÍCULOS	PORCENTAJE
CNN	S9, S12, S16, S17, S25, S28, S29 S32, S48, S50, S58	11	24%
LSTM	S4, S9, S16, S17, S25, S27, S28, S31, S33, S37, S44, S48, S50, S58	14	29%
GRU	S17, S37, S40, S44, S48, S58	6	13%
BERT	S41	1	2%
MLP	20, S28, S32, S55	4	9%

MODELO GENERICO	S5, S6, S19, S39, S42, S49, S53	7	16%
VAINILLA	S48	1	2%
CASNET	S17, S58	2	5%

Redes convolucionales (CNN) y la detección de noticias falsas

En el artículo 9, se empleó una CNN simple con 3 bloques de convolución, cada una de los cuales consta de una sola capa Conv-1D y una capa MaxPooling. Además de estos bloques, se agregaron una capa plana y una capa completamente conectada con 128 nodos. Finalmente, este modelo clasifica noticias falsas y reales usando la función de activación de LeakyReLU. La precisión media fue del 73,29%.

Para el artículo 17, los autores trabajaron con dos conjuntos de datos, LIAR con un total de 12,836 y POLITIFACT con un total de 3831. Incluyeron limpieza de texto, eliminación de palabras vacías y tokenización. El modelo CNN contiene una capa de incrustación con dropout establecido en 0.2, una Convolution1D que aprende cómo filtrar grupos de palabras, una capa GlobalMaxPool, así como una capa oculta básica que contiene capa dense, dropout establecida en 0.2 y función de activación relu. El resultado se proyecta en una capa de salida de una sola unidad aplasta con softmax. Obtuvieron un 27% exactitud.

Para el artículo 25, los autores trabajaron con un conjunto de datos de código abierto, compuesto por 5060 artículos de noticias extraídos de una lista seleccionada de 1001 dominios de noticias de Estados Unidos y Singapur. El modelo CNN tiene una agrupación máxima global. El modelo se alimenta primero con una capa de incrustación, seguida de una capa de convolución 1D, agrupación máxima de 1D, capa de convolución 1D, operación de agrupación máxima global 1D y una capa densa. A continuación, la entrada auxiliar que consta de las tres características basadas en atributos se concatena con la entrada principal. Luego, el modelo se somete a tres capas de redes neuronales densamente conectadas. Luego, la salida principal se genera con la función de activación sigmoide. El resultado de exactitud obtuvo 92.4%.

Por otra parte, en el artículo 29 usaron una CNN en la que incluyeron varias capas que se describen como sigue: La primera de incrustación en la capa de entrada, cada palabra en el texto que es una ficha en el nivel de palabra, está incrustada en un vector con una longitud de 100. Cualquier texto que contenga menos del número máximo de

tokens se rellena para que tenga la misma longitud de 100. Por lo tanto, se usó la capa de incrustación de flujo de tensor con un tamaño de vocabulario de 50000 y una longitud de oración de 100. Para el parte del autor de la noticia, la longitud de la oración se fijó en 2. Para los metadatos del título, utilizaron una matriz de incrustación de 20×20 . Las salidas se alimentaron en capas convolucionales separadas.

La segunda capa convolucional usó 32 filtros con tamaño 5 para extraer características de palabras del texto de los artículos. Cada filtro detectó múltiples características en el texto usando la función ReLu para representarlas en el mapa de características. Lo mismo se hace en la parte del autor con un tamaño fijo en 1. Además, se utilizó la regularización (regularizador de actividad L2) para evitar el sobreajuste en la red.

La tercera capa *maxpooling* minimiza y reduce la muestra en el mapa de características. La operación o función *maxpooling* es una técnica comúnmente utilizada para esta capa y se utiliza en este experimento. La razón de seleccionar el valor más alto es capturar la característica más relevante, y reducir el cálculo en las capas avanzadas.

La cuarta capa es *flatten*, es necesaria para fusionar los dos modelos CNN y para alimentar la salida de la fusión a la capa completamente conectada. Por tanto, se concatenaron ambas salidas de las capas planas de cada modelo.

La quinta capa completamente conectada, que después de combinar la capa aplanadora anterior en una matriz, la salida se pasó a una capa completamente conectada. La matriz se convirtió en una única salida en el rango entre 0 y 1. Se utilizó la función sigmoide. Los datos se dividieron en 90 % entrenamiento, 10 % prueba. Los autores realizaron 3 modelos, el primero con CNN contenido del artículo, el segundo CNN con contenido y título, y el último modelo con contenido y autor; en todos se usó 50 épocas para el entrenamiento. Para el primer modelo se obtuvo 89%, el segundo 93% y para el tercero 96% de precisión. Es importante argumentar que la integración de metadatos relacionados con el autor en el texto de las noticias mejoró el rendimiento de la detección.

Memoria corto plazo (LSTM) y la detección de noticias falsas

En el artículo 9 realizaron un modelo híbrido donde combinaron dos arquitecturas CNN y LSTM usando 3 Conv-1D y capas 3MaxPooling junto con la capa LSTM. Los resultados demostraron que la red logró una precisión promedio del 86,14 %.

Para el artículo 12, se usaron cuatro modelos híbridos con CNN+ LSTM. La primera capa del modelo es la capa de incrustación que acepta los titulares de entrada y los cuerpos de los artículos (noticias) y convierte cada palabra en un vector de tamaño 100. El número de funciones es 5000, por lo tanto, esta capa generó una matriz de tamaño $5000 * 100$. La matriz de salida tuvo pesos que obtenemos mediante la multiplicación de matrices, para producir un vector para cada palabra. Estos vectores se pasaron a la capa CNN para extraer características contextuales. La salida de la capa CNN se alimentó a LSTM y luego se pasó a una capa densa completamente conectada para producir una sola postura como salida final utilizando una activación Relu. Se usaron 4 modelos en el primero no realizó procesamiento, el segundo integró procesamiento, el tercer modelo usó chi-cuadrado, y el cuarto empleó el análisis de componentes principales (PCA). Entre el modelo sin procesamiento y con procesamiento, el mejor porcentaje fue del 93% con procesamiento, y la mejor técnica fue PCA con 97.8% de precisión.

En el artículo 17 también se propusieron dos modelos usando LSTM, el primero fue LSTM básico con dimensión de 300 una capa de GlobalMaxPool, un abandono espacial establecido en 0.2 y capas densas con activación softmax. Algunos de los hiperparámetros incluyen un tamaño de lote de 256, épocas establecidas en 20 y tasa de aprendizaje establecida en 0.001. El segundo modelo, es con atención, abandono y abandono recurrente establecido en 0,25 y una capa densa con activación softmax. Este último modelo empleó los mismos hiperparámetros que el modelo anterior.

Para el estudio 44, los autores trabajaron con un conjunto de datos de noticias que extraídas de Twitter que cuenta con 20.360 datos que contiene información sobre los terremotos de Chile de 2010. Usaron el clasificador LSTM que contiene una capa de incrustación, una capa oculta, una capa de entrada y una capa de salida, además usaron la activación sigmoide, inicializadores de kernel 'glorot_normal', pérdida de 'categorical_crossentropy' y 'RMSprop' como optimizador. El resultado alcanzado en términos de exactitud fue de 78%.

En el artículo 25, se usó el modelo BiLSTM que se alimentó primero con una capa de incrustación, seguida de una capa LSTM bidireccional y un grupo máximo global 1D que consideró todos los estados ocultos, pero solo devolvió una salida con el valor máximo. Tras la concatenación de la entrada auxiliar con la entrada principal, el modelo se sometió a tres capas de redes neuronales densamente conectadas. Luego, la salida principal se generó con la función de activación sigmoidea. Se obtuvo un 95.30 % de exactitud.

En el artículo 33, dos modelos con LSTM fueron propuestos, el primero consta de dos capas de CNN con 32 y 64 filtros. Las CNN son seguidas por la agrupación máxima para evitar el sobreajuste antes de pasarlo por una capa LSTM bidireccional con 100 unidades de memoria. La salida del LSTM bidireccional pasó a través de tres capas densas (512, 128, 4 unidades) separadas por una salida. Usando la función de activación soft-max en la capa de salida, el resultado es una clasificación de la postura (no relacionada, de acuerdo, en desacuerdo o discutir). En el segundo modelo, Multihead LSTM, los títulos y artículos fueron en una sola capa, se realizó la capa de incrustación utilizado el modelo previamente entrenado de Google. La capa de incrustación resultante del paso anterior se pasó a través de dos capas de CNN con 32 filtros, la capa de incrustación resultante del paso anterior se pasa a través de dos capas de CNN con 64 filtros, le sigue la agrupación máxima para evitar un ajuste excesivo. Luego, la salida pasó a través de 5 capas Multi-Head LSTM con 150 unidades de memoria y después se agregó una capa plana, luego el resultado se pasó a través de una capa densa con 4 unidades y una función de activación softmax para clasificar la postura en 4 clases (los mismos anteriores). La función de pérdida en el proceso de entrenamiento fue *sparse_categorical_crossentropy* con optimizador Adam. Se obtuvo 85.30 % de precisión.

Para el artículo 37, se usó LSTM con RNN y constó de una capa de entrada. Como, cada palabra debió estar representada por un número entero único y para eso los datos de entrada debieron estar codificados como números enteros, se utilizó un conjunto adicional de capas de incrustación. En cada actualización, durante el tiempo de entrenamiento, la deserción estableció aleatoriamente una tasa de fracción de unidades de entrada en 0, lo que ayudó a evitar el sobreajuste. En lugar de elementos individuales, 1D elimina mapas de características 1D completos. Otras capas incluyen la capa 1D de abandono espacial, la capa LSTM, la capa densa con activación RELU, la capa de abandono y la capa de salida densa, sigmoidea, con optimizador Adam. El algoritmo de optimización de Adam actualizó de forma iterativa los pesos de los parámetros de red en función de los datos de entrenamiento. Luego se calculó la pérdida de entropía cruzada binaria con 4 épocas.

Perceptrón multicapas (MLP) y la detección de noticias falsas

Con relación a la categoría de MLP, con cuatro estudios (9%), en el estudio 55 los investigadores usaron 3 conjuntos de datos, luego se realizó la limpieza del conjunto de datos y el texto del título y el cuerpo se tokenizaron y se derivaron para pasarlos por el

modelo n-gram de manera que se pudieron para hacer las pruebas dividiendo el set de dato en 80% para entrenamiento y el 20% pruebas. El mejor porcentaje lo obtuvo el segundo conjunto de datos con un 88% de precisión.

Por otro lado, en el estudio 16, usaron 4 conjuntos de datos del cual el primero tiene un total de 44.898 artículos, el segundo total de 20.386 artículos, el tercero 3.352 artículos y el último, la combinación, para la red neuronal se utilizó la función de activación ReLU y el solucionador de Adam, con 3 capas ocultas. El resultado fue 98% de exactitud.

Modelo genérico y la detección de noticias falsas

En el artículo 46, para el modelo red utilizaron el método de propagación hacia atrás y la función sigmoide. Siguiendo lineamientos similares, en el estudio 46, los autores utilizaron el conjunto de datos obtenidos de Kaggle donde el contenido y metadatos fueron extraídos de 244 sitios web, logrando emplear alrededor de 150000 artículos que contiene publicaciones estadounidenses entre los años 2016 y 2017.

Antes de pasar los datos a los algoritmos se realizó la etapa preprocesamiento de los datos en el cual consideraron el modelo bag of words y para implementarlo usaron el *count vectorizer* y TF-IDF. Este estudio trabajó con una red neuronal en la que los parámetros utilizados fueron para la llevar a cabo una clasificación binaria, además emplearon la entropía cruzada binaria para calcular la pérdida. Asimismo, la función sigmoidea se utilizó como función de activación para la capa de salida y ReLU para las capas ocultas. Se obtuvo un 49,99 % de exactitud tanto en *count vectorizer* y TF-IDF.

Unidades recurrentes cerradas (GRU) y la detección de noticias falsas

En el estudio 17, los autores trabajaron con un GRU básico con dimensión de 300, una capa de *GlobalMaxPool*, para *spatial dropout* establecido en 0.2, abandono recurrente establecido en 0,25 y capas densas con activación softmax, además incluyeron un tamaño de lote de 256, épocas establecidas en 20 y tasa de aprendizaje establecida en 0.001.

También en el artículo 37 utilizaron GRU trabajaron con un conjunto de datos que se acumuló a partir del raspado web utilizando *beautiful soup* en Python que contiene noticias de India y rastro internacional. Realizaron eliminación de palabras vacías, luego lematizaron las palabras a su origen y usaron *FastText* para hacer incrustaciones de palabras y tokenizar el texto de preprocesamiento de Keras. El modelo fue híbrido,

combinando GRU+RNN con parámetro de 4 épocas, los resultados obtenidos fueron de 92% de exactitud.

CapsNet y la detección de noticias falsas

En la categoría CapsNet. estuvieron enmarcados dos estudios (5%), entre ellos están en el artículo 17 que presenta un modelo que empleó una capa *capsule* en lugar de la capa *GlobalMaxPool* y obtuvo un 59.80% de exactitud. Mientras que el artículo 58 planteó el conjunto de datos de liar que contiene artículos relacionados con la política. Para los clasificadores utilizaron 10, los cuales son 6 de aprendizaje automático, estos son: multinomial Naïve Bayes, SGD, LR, RF, DT, SVM y 5 para aprendizaje profundo CNN, básico LSTM, BI-LSTM, GRU, CapsNet. Las herramientas utilizadas es Keras. Los resultados obtenidos son SVM con 28% y CapsNet con 64%.

Resumiendo, se han visto involucrado diferentes redes para la detección de noticias falsas y para aprendizaje profundo. Sus parámetros tienden a variar con diferentes capas agregadas o funciones que se le activa a la red. También han sido bastante utilizados los modelos híbridos obteniendo buenos resultados de exactitud. Entre el modelo más usado estuvo memoria a corto plazo este fue el usado para los modelos híbridos ya sea un LSTM Bidireccional o acompañada de la red convolucional.

RQ3: ¿En qué ámbitos se han empleado las técnicas de aprendizaje de la IA para la detección de noticias falsas? (i.e., política, ciencia, etc.)

Como se ilustra en la Tabla 10, se identificaron siete categorías incluida la categoría no especificada. Entre las siete categorías en las que se ha intentado detectar noticias falsas usando IA figuran las siguientes: ciencia, tecnología y salud con 1 estudio (2%) en las que está el S16; política con 22 estudios que representa el (58%); categoría en las que figuran los estudios: S9, S16, S17, S19, S24-S26, S28, S29, S31, S32, S40-S42, S46, S48, S50, S55, S58, S60, S61 y S63; negocio y economía con dos estudios, eso es, un 5% que incluye los estudios S41 y S50; deporte, sociedad y cultura con un estudio (3%); desastre naturales con un estudio 3%; entretenimiento con dos estudios (5%); y 9 estudios no especificaron el ámbito.

Tabla 10. Categoría de pregunta RQ4

Categorías	Artículos	Cantidad	Porcentaje
Ciencia, tecnología y salud	S16	1	2%
	Política	S9, S16, S17, S19, S24, S25, S26, S28, S29, S31, S32, S40, S41, S42, S46, S48, S50, S55, S58, S60, S61, S63	22
Negocio y economía	S41, S50	2	5%
Deporte, sociedad y cultura	S16	1	3%
Desastres naturales	S49	1	3%
Entretenimiento	S16, S50	2	5%
No específico	S4, S12, S34, S35, S37, S38, S39, S47, S3 S	9	24%

La Tabla 10 también muestra que el ámbito en el que más se ha intentado trabajar para la detección de noticias falsas es en política. Esto puede ser debido a que en las elecciones de EE. UU en 2016 se publicó una variedad de información falsa sobre los candidatos y por ese motivo las redes sociales se enfrentan a la presión de descartar noticias falsas influyen la opinión pública.

Siendo el ámbito político que más resalta para realizar los modelos, estos no deben ser generalizados para la identificar solo noticias falsas en esta área; sino también, otros ámbitos susceptibles y delicados que también pueden afectar a la sociedad como es el caso de la salud. Un ejemplo claro de noticia falsa en este ámbito fueron las recetas mágicas o químicos que ayudaban a combatir o eliminar el COVID19 en las personas. Muchas personas han perdido la vida al auto medicarse sin supervisión médica dejados llevar por el contenido que se publica en las redes sociales.

RQ4: ¿Cuál es el alcance que han tenido los modelos de ML y DL en términos de precisión, para la detección de noticias falsas?

Las investigaciones en el campo de la detección de noticias falsas en medios sociales, aplicando técnicas de IA, han sido llevadas a cabo realizando diferentes experimentos a partir de los cuales se ha logrado entrenar modelos de detección de noticias falsas.

Como se puede apreciar en la Tabla 12, los porcentajes de exactitud que se obtuvieron difieren no solo de los datos; sino también de los diferentes algoritmos que han sido aplicados para entrenar los modelos. Los resultados obtenidos mostraron que mayormente se ha aplicado el algoritmo de máquinas de vectores soporte (20 estudios), seguido de los

árboles de decisión con 14 estudios y la regresión logística con 14 estudios; luego, se posiciona el algoritmo de bosques aleatorios con 12 estudios, Naïve Bayes con 9 estudios, y el algoritmo de vecinos más cercanos con 7 estudios. Asimismo, se han aplicado otros algoritmos menos populares como la potenciación de gradiente con 6 estudio, descenso estocástico de gradiente con dos 2 estudios, y un estudio que empleó el algoritmo basado en el área bajo la curva, pasivo agresivo y *xg-boost*.

Por parte del algoritmo Naïve Bayes, éste alcanza un porcentaje de 99 % de exactitud. Por parte de árbol de decisión el estudio S16 alcanzó un 98%, y el algoritmo de bosque aleatorio llegó a alcanzar un 99%. En relación con el algoritmo vecinos más cercanos, éste obtuvo un porcentaje de exactitud del 88% y la máquina de vectores de soporte llegó a alcanzar el 99,90% en términos de exactitud. Finalmente, es importante señalar que la regresión logística logró un 97%, el algoritmo de potenciación de gradiente un 91,05%, el algoritmo del área bajo la curva obtuvo un 89,95 %, el algoritmo de potenciación adaptativa 98 %, pasivo agresivo el 100% y *xg-boost* logró un 98% de exactitud.

Por parte de aprendizaje automático, el algoritmo que mejor se desempeñó fue el algoritmo pasivo agresivo, siendo uno de los clasificadores menos usados, obtuvo un puntaje de 100% para las pruebas. También, cabe señalar que SVM fue el algoritmo de aprendizaje supervisado que más se usó en las investigaciones publicadas en la literatura, llegando a alcanzar un porcentaje de exactitud aceptable del 99,90 %.

En cuanto a aprendizaje profundo, que también el que mejor se desempeño fue modelo genérico con 99,90 exactitud, pero también las CNN que tuvieron 97,8, aunque las LSTM que han sido también unas más usadas en estas técnicas no se han obtenidos porcentaje tan alto lo máximo que alcanzó fue de 95.30% como se muestra en Tabla 13. Cabe mencionar que se esperó a que LSTM fuera el que tuvieron entre los porcentajes más altos; pero no fue así siendo este el más usado, su porcentaje fue inferior al menos usado que fue *passive-agresive*.

Cabe recalcar que la exactitud de los diferentes modelos depende de varios factores como es el conjunto de datos y el algoritmo de aprendizaje usado. Eso se ha discutido en varios de los estudios analizados. El resaltado amarillo representa los modelos que obtuvieron un porcentaje mayor en los diferentes algoritmos y el turquesa que fue el mejor de todos y en negrita al mejor pero referente al estudio.

RQ5: ¿Qué herramientas software se emplearon para crear los modelos predictivos de detección de noticias falsas?

Las herramientas con las cuales trabajaron los investigadores para plantear sus propuestas prácticas enfocadas en detectar noticias falsas figuran las siguientes: Python (4 estudios (8%); *sci-kit* con 4 estudio (26%); y *Keras* con 12 estudios (26%). También figuran herramientas como: *Tensorflow* con 6 estudios (13%); NLTK para el procesamiento del texto de las noticias con 9 estudios (19%); herramientas de Google con 2 estudios (4%); y *Spacy* con 2 estudios (4%). Estos resultados, se muestran en la Tabla 11.

Tabla 11. Herramientas de software empleadas para desarrollar los clasificadores de noticias falsas.

Categoría	Artículos	Cantidad	Porcentaje
Python	S4, S7, S38, S47	4	8%
Sci-kit learn	S7, S32, S60, S62	4	9%
Keras	S4, S5, S7, S12, S17, S19, S25, S28, S29, S37 S42, S58	12	26%
Tensoflow	S7, S17, S37, S42, S28, S29	6	13%
Nltk	S4, S25, S31, S32, S37, S47, S55, S62, S63	9	19%
Google	S24, S42	2	4%
Spacy	S27, S60	2	4%
No detallaron	S16, S24, S34, S40, S49 S20, S26, S53	8	17%

Es importante señalar, que también fueron analizados varios estudios en los que no se describieron detalles respecto a las herramientas usadas para la implementación de los clasificadores. Este es el caso particular de 8 estudios, que equivale al 17% de los trabajos analizados. Una dificultad de este tipo de estudios es que resultan complejos o imposibles de replicarlos. También, esto imposibilita comprobar su grado de eficacia para la clasificación de noticias falsas y mejorarlos si fuera el caso.

Los resultados evidenciaron que se está empleando Python y las librerías y marcos de desarrollo orientados a este lenguaje para el desarrollo de sistemas de clasificación que sean capaces de aprender a detectar noticias falsas. También se evidenció el uso de estas herramientas que operan desde la nube, esto se debe a que el proceso de entrenamiento es complejo y requiere tiempo de procesamiento.

Tabla 12. Porcentaje de exactitud de los algoritmos de aprendizaje automáticos usados en la detección de noticias falsas.

#	NB	DT	RF	KNN	SVM	LR	GB	AUC	SGD	AB	PASSIVE AGGRESSIVE	XG- BOOST	BASELINE
5	-	96	-	-	-	-	-	-	-	-	100	-	-
6	-	82	90	-	94	93	90	-	-	-	-	89	-
7	-	65	68	66	-	71	-	-	-	-	-	-	-
9	-	-	-	-	58,68	57,58	-	-	-	-	-	-	-
16	-	98	99	88	98	97	-	-	-	98	-	98	-
17	29,50	26,90	24,40	-	28,10	26,90	-	-	29,40	-	-	-	-
19	99	96	93	-	-	-	-	-	-	-	-	-	-
20	-	90,2	89,99	67,32	73,28	-	-	-	-	-	-	-	-
26	72	74	70	-	76	68	86	-	-	-	-	-	-
27	-	67,60	-	67,60	76,20	-	68,70	-	77,20	-	-	-	67,89
30	68	64	65	70,00	68	69,00	-	-	-	-	-	-	-
32	-	47,00	55,00	-	58,00	58,00	-	-	-	-	-	-	-
34	-	-	-	-	-	77,00	-	-	-	-	-	-	-
35	-	-	-	-	93,60	-	-	-	-	-	-	-	-
36	-	-	-	-	59,00	-	-	-	-	-	-	-	-
38	-	82,70	90,70	79,20	75,50	-	-	-	-	-	-	-	-
39	89,00	-	77,00	-	79,00	75,00	-	-	-	-	-	-	-
44	89,06	-	-	-	89,34	69,47	-	-	-	-	-	-	-
46	86,30	82,5	-	-	92,80	91,60	-	-	-	-	-	-	-
47	83,16	-	-	-	81,65	66,57	-	-	-	-	-	-	-
49	96,08	-	-	-	99,90	-	-	-	-	-	-	-	-
53	68	-	-	-	76	-	87	-	-	-	-	-	-

55	-	-	86,63	72,54	63,55	-	91,05	89,25	-	-	-	-	-
57	60		60		60	60			60				
58	-	26,20	23,80	-	28,40	26,00	-	-	-	-	-	-	-
60	-	67,60	64,80	-	73,60	-	65,77		65,70	-	-	-	-
61	-	-	89,18	-	-	89,1	-	-	-	-	-	-	-
62	88,10	-	-	-	-	-	-	-	-	-	-	-	-
63	-	89	-	-	92	89	-	-	89	-	-	-	-
Σ	12	16	16	7	23	17	6	1	5	1	1	2	1

Tabla 13. Porcentaje de exactitud de los algoritmos de aprendizaje profundo usados en la detección de noticias falsas.

#	CNN	LSTM	GRU	BERT	MLP	MODELO GENERICICO	VAINILLA	CAPSNET
4	-	90,20	-	-	-	-	-	-
5	-	-	-	-	-	99	-	-
6	-	-	-	-	-	94	-	-
8	-	-	-	-	-	-	-	-
9	73,29	80,62	-	-	-	-	-	-
12	97,8	-	-	-	-	-	-	-
16	87	86	-	-	-	-	-	-
17	27	28,90	49,60	-	-	-	-	59,80
19	-	-	-	-	-	94	-	-
20	-	-	-	-	78,11	-	-	-
25	92,40	95,30	-	-	-	-	-	-
27	-	82,29	-	-	-	-	-	-

28	97,5	91,8	-	-	89,8	-	-	-
29	96	-	-	-	-	-	-	-
31	-	71,20	-	-	-	-	-	-
32	60,00	-	-	-	58,00	-	-	-
33	-	85,30	-	-	-	-	-	-
37	-	94,30	91,90	-	-	-	-	-
39	-	-	-	-	-	91,00	-	-
40	-	-	83,38	-	-	-	-	-
41	-	-	-	96,25	-	-	-	-
42	-	82,61	-	-	-	-	-	-
44	-	76,00	74,00	-	-	-	-	-
46	-	-	-	-	-	49,90	-	-
48	27	21,66	21,70	-	-	-	21,5	-
49	-	-	-	-	-	99,90	-	-
50	72,60	41,70	-	-	-	-	-	-
53	-	-	-	-	-	92	-	-
55	-	-	-	-	88,36	-	-	-
58	29,00	32,40	54,90	-	-	-	-	64,4
Σ	11	14	6	1	4	7	1	2

RQ6: ¿En qué medios sociales se basaron las investigaciones, o de qué medios sociales se obtuvo los datos para entrenar los modelos propuestos?

Los modelos creados que utilizaron los conjuntos de datos para hacer las pruebas fueron usados solo el 16% para las dos principales redes sociales contempladas en este estudio, esto es, Twitter y Facebook con un total de 7 estudios. El resto, con 84% tomaron los datos de diferentes sitios web utilizando el método de raspado web aplicándolo ya sea en medios de comunicación o no para extraer la información.

En la Figura 24 se evidencia que falta trabajar en datos que sean extraídos de las redes sociales, no solo en Facebook y Twitter; sino también, en Instagram o WhatsApp que son también ampliamente usadas por los usuarios. Cabe mencionar, que de los diferentes modelos que han sido propuestos ninguno está implementado en estas redes sociales, solo se ha trabajado con los datos extraídos de éstas. Solo dos trabajos de los analizados fueron aplicados en redes reales; el primero en una aplicación Android la cual verificó si la noticia es falsa o verdadero en base a su URL (estudio S42), mientras que el otro modelo llevado a una aplicación real fue integrado en un portal web y se basó en la URL y en el contenido de las noticias (estudio 34) esto permitió a las personas verificar antes de compartir una noticia.

Lo antes expuesto, demuestra que la detección de noticias falsas en redes sociales en tiempo real es un campo que necesita ser estudiado; pero sobre todo requiere de sistemas que sean integrados en las propias redes sociales. No obstante, esta es una tarea que le compete a los propietarios de las redes sociales más empleadas a nivel mundial como Facebook y Twitter.

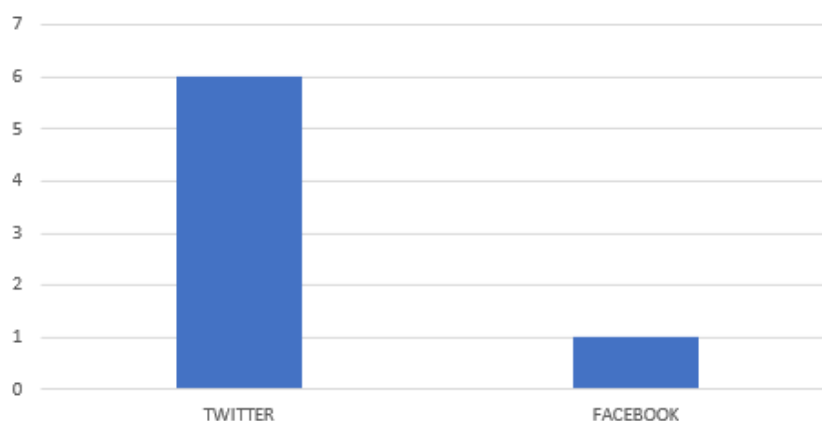


Figura 24. Principales redes sociales desde las que se ha extraído datos para entrenar clasificadores para la detección de noticias falsas.

Es importante señalar, que los conjuntos de datos usados para entrenar los modelos de aprendizaje para detectar noticias falsas han sido accedidos desde diferentes repositorios o creados por los propios autores. Sin embargo, esto requiere experiencia humana para identificar artículos para ser utilizados en la fase de entrenamiento y prueba de los modelos desarrollados.

En las investigaciones se trabajó con algunos conjuntos de datos y con diferentes tamaños. Se realizó un análisis basado en seis categorías, entre las que se definieron las siguientes: 2 -2000 noticias, 2001- 4000, 4001- 6000, 6001- 8000, 8001- 10000 y más de 10000 noticias. Así, de los 45 artículos que realizaron una práctica para la detección de noticias falsas y que han tomado los conjuntos de datos, solo 41 artículos dan detalles del número de noticias usadas en su set de datos; los restantes no detallaron la cantidad. Como se puede observar en la Figura 25, las investigaciones trabajan mayoritariamente con un conjunto de datos mayor a 10.000 noticias, lo que puede ser positivo para los modelos debido a que tienen más datos para entrenar y la predicción será más precisa. Esto se debe a que tendrán mayor cantidad de datos para aprender de ellos.

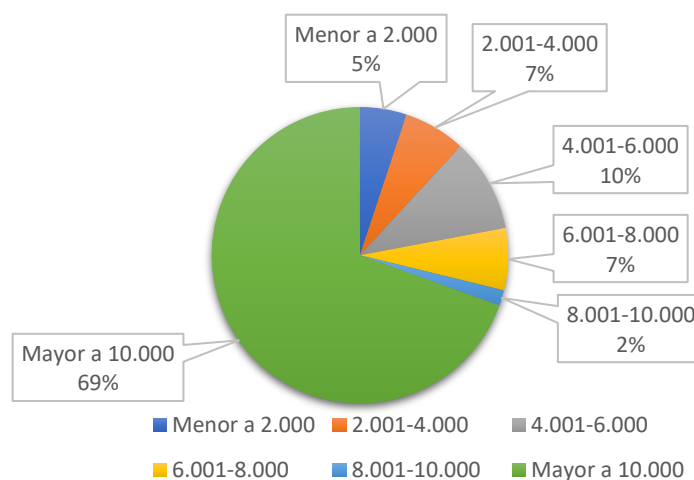


Figura 25. Tamaño de los conjuntos de datos utilizados en las investigaciones

Con las propuestas descritas anteriormente se pudo notar que es necesario limpiar el conjunto de datos, ya que esto permite obtener información más relevante y esto se lo hace a través del proceso de vectorización o la fase de preprocesamiento de datos. En los artículos aplicaron en los corpus la eliminación de palabras vacías como preposiciones y

conjunciones y algunos pronombres. También realizaron un proceso llamado derivación que consiste en cambiar las palabras a su forma original y disminuir el número de tipos o clases de palabras en los datos. Por ejemplo, las palabras "corriendo ", "corrió " y "corredor " se redujo a la palabra "correr".

Las características ruidosas irrelevantes en los datos reducen el rendimiento de los clasificadores. Por este motivo, los diferentes trabajos han requerido una reducción de características innecesarias en la que ayuda a reducir el tamaño de las características del texto. En la Figura 26, se muestran los principales algoritmos de extracción de características utilizadas en los artículos, siendo la más utilizada TF-IDF en 11 artículos. Esto se debe a que este método para generar vectores a partir del texto es una mejor alternativa en relación con las incrustaciones de palabras, ya que reduce el impacto de los *tokens* que ocurren con mucha frecuencia en los corpus por lo que dichos token son menos informativos, siendo una buena alternativa de usarla en los diferentes modelos y algoritmos ya que brinda mejores resultados.

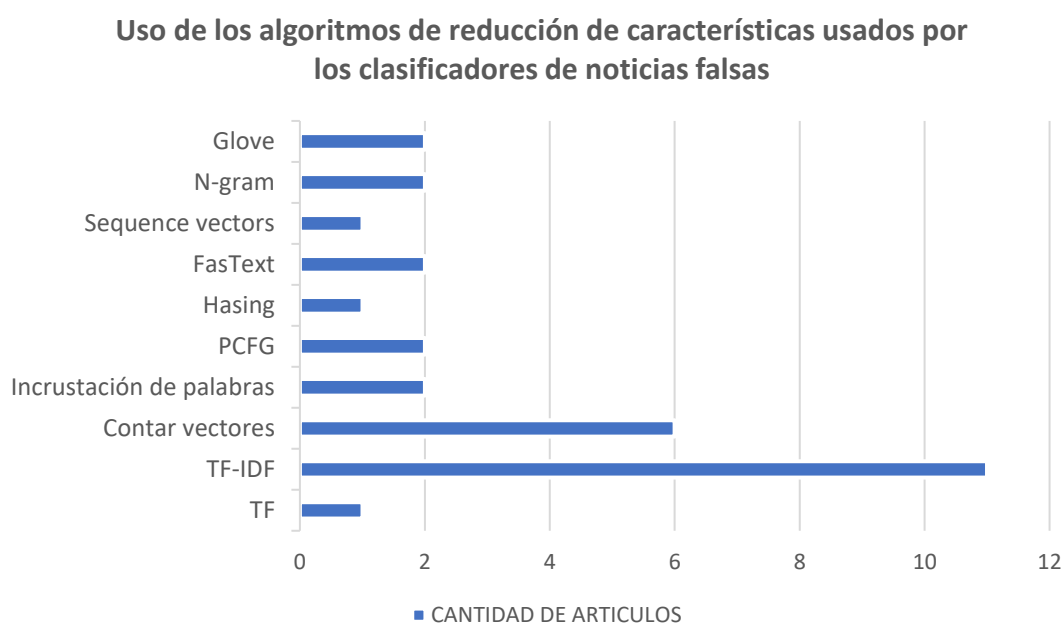


Figura 26. Extracción de característica utilizados en los modelos.

4.4. Evaluación de modelos de clasificación de noticias falsas

Con la finalidad de analizar de manera práctica el funcionamiento de los modelos de clasificación de noticias falsas en medios sociales, se usaron para ello, cuatros modelos la

aplicación 1, 2 y 3 referente a aprendizaje automático, y la aplicación 4 en la que se empleó un modelo de aprendizaje profundo. Dichos sistemas fueron tomados en repositorios públicos, accesibles a través de las siguientes direcciones electrónicas:

- **Aplicación 1:** <https://github.com/singh08prashant/FAKE-NEWS-DETECTION>.
Autor: Prashant Singh. Fecha de publicación: 25/05/2020.
- **Aplicación 2:** <https://github.com/FakeNewsDetection/FakeBuster>
Autor: Keval Khara. Fecha de publicación: 11/05/2018
- **Aplicación 3:** <https://github.com/Sudip-Pandit/Fake-News-Detection-with-Machine-Learning->
Autor: Sudip Pandit. Fecha de publicación: 9/02/2021.
- **Aplicación 4:**
https://github.com/mohamedScikitLearn/fake_news_detection_LSTM-TF
Autor: Mohamed Berrimi. Fecha de publicación: 5/06/2020

Evaluación cualitativa y cuantitativa de sistemas propuestos en repositorios

La evaluación cualitativa consistió en describir los algoritmos usados para experimentar desde una perspectiva práctica con algunos de los clasificadores creados y subidos a repositorios públicos. En la exploración en los repositorios en Internet, se pudo descargar y poner en operación clasificados de noticias falsas basados en los siguientes algoritmos: *passive agresive*, máquina de soporte, regresión logística, bosques aleatorios, y memoria a largo y corto plazo. Las técnicas usadas para la extracción de características fue TF-IDF y los trabajos fueron desarrollados por sus autores en Python. En relación con la evaluación cuantitativa, ésta estuvo enfocada principalmente en la medición de métricas de rendimientos de los clasificadores testeados como: *accuracy*, *recall*, *precisión*, y *f1-score*.

A continuación, se describe el set de datos empleado para entrenar los clasificadores evaluados. Dicho set de datos fue el mismo para todos los modelos y fue obtenido de Kaggle [124]. El set de datos, denominado “*Fake News*”, se constituyó de los siguientes atributos: atributos id, título de la noticia, autor, texto de la noticia, etiqueta 1 representa a falso o 0 real y con un total de 10413 real y falsas 10387. En la Figura 27 se muestra un

ejemplo de cuatro noticias que formaron parte del set de datos usado.

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Aistr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

Figura 27. Conjuntos de datos trabajado para las pruebas.

Sistema de detección de noticias falsas usando los modelos regresión logística, Naïve Bayes¹ desarrollado por Prashant Singh.

El algoritmo 1 plantea un clasificador para la detección de noticias falsas usando la regresión logística y Naïve Bayes. El procedimiento incluyó bloques en los que se incluyen las librerías (líneas 1-9), se carga y se visualiza el conjunto de datos (líneas 10-20), se hace la técnica TF-IDF (líneas 21-26), se entrena un clasificador usando el algoritmo Naïve Bayes (línea 27-36) y el algoritmo de regresión logística y las métricas de evaluación (línea 37-48).

Algoritmo 1. Clasificador de noticias falsas usando el algoritmo de regresión logística y Naïve Bayes.

```

1. import numpy as np
2. import pandas as pd
3. from sklearn.metrics import confusion_matrix
4. from sklearn.metrics import precision_score, recall_score, f1_score,
   accuracy_score
5. from sklearn.feature_extraction.text import CountVectorizer
6. from sklearn.feature_extraction.text import TfidfTransformer
7. from sklearn.model_selection import train_test_split
8. from sklearn.naive_bayes import MultinomialNB
9. from sklearn.linear_model import LogisticRegression
10. train = pd.read_csv('train.csv')
11. test = pd.read_csv('test.csv')
12. train.info()
13. test.info()
14. train.head(10)
15. test['label'] = '1'
16. test.head()
17. train.fillna(' ', axis=0, inplace=True)
18. test.fillna(' ', axis=0, inplace=True)
19. train['total'] = train['title'] + ' ' + train['author'] + train['text']
20. test['total'] = test['title'] + ' ' + test['author'] + test['text']
21. transformer = TfidfTransformer(smooth_idf=False)
22. count_vectorizer = CountVectorizer()
23. counts = count_vectorizer.fit_transform(train['total'].values)

```

¹ <https://github.com/singh08prashant/FAKE-NEWS-DETECTION>

```

24. tfidf = transformer.fit_transform(counts)
25. targets = train['label'].values
26. X_train,X_test, y_train, y_test = train_test_split(tfidf, targets, random_state=0)
27. classifier_NB = MultinomialNB()
28. classifier_NB.fit(X_train,y_train)
29. y_pred=classifier_NB.predict(X_test) #modifico
30. matriz = confusion_matrix(y_test, y_pred)
31. print('Matriz de Confusión:')
32. print(matriz)
33. print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
34. print('Precision: %.3f' % precision_score(y_test, y_pred))
35. print('Recall: %.3f' % recall_score(y_test, y_pred))
36. print('F1 Score: %.3f' % f1_score(y_test, y_pred))
37. lasso = LogisticRegression(C=1e5)
38. lasso.fit(X_train,y_train)
39. y_pred=lasso.predict(X_test)
40. from sklearn.metrics import confusion_matrix
41. matriz = confusion_matrix(y_test, y_pred)
42. print('Matriz de Confusión:')
43. print(matriz)
44. from sklearn.metrics import precision_score, recall_score, f1_score,
accuracy_score
45. print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
46. print('Precision: %.3f' % precision_score(y_test, y_pred))
47. print('Recall: %.3f' % recall_score(y_test, y_pred))
48. print('F1 Score: %.3f' % f1_score(y_test, y_pred))

```

Sistema de detección de noticias falsas usando el algoritmo máquina de vectores de soporte² desarrollado por Keval Khara.

El algoritmo 2 usa SVM para la detección de noticias falsas. El procedimiento incluye bloques para importar las librerías necesarias (líneas 1-7), muestra la matriz de confusión (líneas 8-11), carga y visualiza el conjunto de datos (líneas 12-25), entrena el modelo basado en el algoritmo SVM (líneas 26-33).

Algoritmo 2. Clasificador de noticias falsas usando el algoritmo SVM.

```

1. from getEmbeddings import getEmbeddings
2. import numpy as np
3. from sklearn.svm import SVC
4. import matplotlib.pyplot as plt
5. import scikitplot.plotters as skplt
6. import pandas as pd
7. import os
8. def plot_cmat(yte, ypred):
9.     '''Plotting confusion matrix'''
10.    skplt.plot_confusion_matrix(yte,ypred)
11.    plt.show()
12.    # Read the data
13.    if not os.path.isfile('./xtr.npy') or \
14.    not os.path.isfile('./xte.npy') or \
15.    not os.path.isfile('./ytr.npy') or \
16.    not os.path.isfile('./yte.npy'):

```

² <https://github.com/FakeNewsDetection/FakeBuster>

```

17. xtr,xte,ytr,yte = getEmbeddings("train.csv")
18. np.save('./xtr', xtr)
19. np.save('./xte', xte)
20. np.save('./ytr', ytr)
21. np.save('./yte', yte)
22. xtr = np.load('./xtr.npy')
23. xte = np.load('./xte.npy')
24. ytr = np.load('./ytr.npy')
25. yte = np.load('./yte.npy')
26. clf = SVC()
27. clf.fit(xtr, ytr)
28. y_pred = clf.predict(xte)
29. m = yte.shape[0]
30. n = (yte != y_pred).sum()
31. print("Accuracy = " + format((m-n)/m*100, '.2f') + "%") # 88.42%
32. # Draw the confusion matrix
33. plot_cmat(yte, y_pred)

```

Sistema de detección de noticias falsas usando el algoritmo passive-agresive³ desarrollado por Sudip Pandit.

El algoritmo 3 entrena un clasificador para la detección de noticias falsas. El código incluyó los siguientes bloques: inclusión de las librerías necesarias (líneas 1-22), carga del conjunto de datos y conversión de la etiqueta real a 0 y falso a 1 (líneas 23-28), visualización la distribución anterior de la etiqueta y verificación de los valores faltantes (líneas 29-43), llenado de los valores faltantes y se dividen los conjuntos de datos (línea 43-46), aplicación de la técnica de bolsa de palabras y parte del clasificador passive agresive (línea 46-56) y cálculo de las métricas de rendimientos (línea 57-64),

Algoritmo 3. Clasificador de noticias falsas usando el algoritmo passive agresive.

```

1. import pandas as pd
2. import numpy as np
3. import seaborn as sns
4. import matplotlib.pyplot as plt
5. %matplotlib inline
6. from sklearn.feature_extraction.text import CountVectorizer
7. from sklearn.feature_extraction.text import TfidfVectorizer
8. from sklearn.naive_bayes import MultinomialNB
9. from sklearn import metrics
10. from sklearn.pipeline import Pipeline
11. #from sklearn.metrics import accuracy_score, confusion_matrix
12. from sklearn.metrics import precision_score, recall_score, f1_score,
    accuracy_score, confusion_matrix
13. from sklearn.linear_model import PassiveAggressiveClassifier
14. from sklearn.model_selection import KFold
15. from sklearn.model_selection import train_test_split
16. from sklearn.model_selection import cross_val_score
17. import itertools
18. import seaborn as sb
19. import pickle
20. from sklearn.metrics import classification_report

```

³ <https://github.com/Sudip-Pandit/Fake-News-Detection-with-Machine-Learning->

```

21. from sklearn.metrics import confusion_matrix
22. import matplotlib.pyplot as plt
23. df_train=pd.read_csv('train.csv')
24. #Convertir la etiqueta real a 0 y falso a 1
25. df_train.head()
26. convert_dic={0:'Real', 1:'Fake'}
27. df_train['label']=df_train['label'].replace(convert_dic)
28. df_train.label.value_counts()
29. def create_distribution(datafile):
30. return sb.countplot(x='label', data=df_train, palette='mako')
31. create_distribution(df_train)
32. def data_quality_check():
33. print("Checking data qualities....")
34. cm=metrics.confusion_matrix(y_test, y1_pred, labels=['Fake', 'Real'])
35. sns.heatmap(cm, square = True, annot = True, fmt = 'd', cbar=False)
36. df_train.isnull().sum()
37. df_train.info()
38. print("check_finished.")
39. data_quality_check()
40. df_train['text']
41. df_train.isnull().any()
42. df=df_train.fillna(' ')
43. x_train, x_test, y_train, y_test=train_test_split(df['text'], df['label'],
test_size=0.30, random_state=63)
44. x_train.head(10)
45. vectorizer=TfidfVectorizer(stop_words='english', max_df=0.75)
46. xv_train=vectorizer.fit_transform(x_train)
47. xv_test=vectorizer.transform(x_test)
48. pass_aggClassifier=PassiveAggressiveClassifier(max_iter=50)
49. pass_aggClassifier.fit(xv_train, y_train)
50. y_pred=pass_aggClassifier.predict(xv_test)
51. print(classification_report(y_test, y_pred))
52. pipeline = Pipeline([('tfidf', TfidfVectorizer(stop_words='english')),
53. ('nbmodel', MultinomialNB())])
54. pipeline = Pipeline([('tfidf', TfidfVectorizer(stop_words='english')),
55. ('nbmodel', MultinomialNB())])
56. y1_pred=pipeline.predict(x_test)
57. matriz = confusion_matrix(y_test, y_pred1)
58. print('Matriz de Confusión:')
59. print(matriz)
60. from sklearn.metrics import precision_score, recall_score, f1_score,
accuracy_score
61. print('Accuracy: %.3f' % accuracy_score(y_test, y_pred1))
62. print('Precision: %.3f' % precision_score(y_test, y_pred1, pos_label='Real'))
63. print('Recall: %.3f' % recall_score(y_test, y_pred1, pos_label='Real'))
64. print('F1 Score: %.3f' % f1_score(y_test, y_pred1, pos_label='Real'))

```

Sistema de detección de noticias falsas usando el modelo LSTM⁴ desarrollado por Mohamed Berrimi.

El algoritmo 4 entrena un modelo de red profunda para la detección de noticias falsas con el modelo de red LSTM. El código incluye las librerías (líneas 1-7), determina los parámetros de la red y procesamiento del lenguaje natural (líneas 8-24), y calcula las métricas del modelo (líneas 26-36).

⁴ https://github.com/mohamedScikitLearn/fake_news_detection_LSTM-TF

Algoritmo 4. Clasificador de noticias falsas usando el modelo LSTM.

```
1. import tensorflow as tf
2. from tensorflow.keras.layers import Embedding, LSTM, Dense ## Neural networks layers
3. from tensorflow.keras.models import Sequential
4. from tensorflow.keras.preprocessing.text import one_hot # to encode the depending
   variable
5. from tensorflow.keras.preprocessing.sequence import pad_sequences
6. from sklearn.model_selection import train_test_split
7. from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
8. voc_size=5000
9. X=[i.lower() for i in X]
10. onehot=[one_hot(words,voc_size) for words in X]
11. sen_len=30
12. embedded_doc=pad_sequences(onehot, padding='pre', maxlen=sen_len) # pad sequence
   your texts
13. print(embedded_doc)
14. embedding_vector_feature=40
15. model=Sequential()
16. model.add(Embedding(voc_size,embedding_vector_feature, input_length=sen_len))
17. model.add(LSTM(100))
18. model.add(Dense(1, activation='sigmoid'))
19. # sigmoid : to handle the output ( binary case )
20. model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
21. print(model.summary())
22. X_final=np.array(embedded_doc)
23. y_final=np.array(y)
24. X_train, X_test, y_train, y_test=train_test_split(X_final, y_final, test_size=0.20,
   random_state=0)
25. model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10,
   batch_size=64)
26. yhat_probs = model.predict(X_test, verbose=0)
27. yhat_classes = model.predict_classes(X_test, verbose=0)
28. yhat_probs = yhat_probs[:, 0]
29. yhat_classes = yhat_classes[:, 0]
30. print('Accuracy: %f' % accuracy)
31. precision = precision_score(y_test, yhat_classes)
32. print('Precision: %f' % precision)
33. recall = recall_score(y_test, yhat_classes)
34. print('Recall: %f' % recall)
35. f1 = f1_score(y_test, yhat_classes)
36. print('F1 score: %f' % f1)
```

En cuanto a las tecnologías para el desarrollo de los clasificadores automáticos, los sistemas evaluados trabajaron con las librerías sklearn, numpy, matplotlib. Nltk. Se limpió el conjunto de datos eliminando las palabras vacías, y la extracción de características se usaron TF-IDF y incrustación de palabras para SVM. El primer experimento se desarrolló usando el algoritmo passive aggressive y su accuracy (exactitud) fue de 85.94 %. El segundo experimento trabajó con un algoritmo bosque aleatorio y se obtuvo 88.40 % de accuracy. El tercer experimento, empleó el algoritmo Naïve Bayes y se obtuvo un 82,51 % de accuracy. El cuarto clasificador empleó un algoritmo de regresión logística y obtuvo 97.67 % de accuracy, el quinto con máquina de soporte se obtiene 91.62%.

Con lo que respecto a aprendizaje profundo el quinto experimento se trabajó con las librerías numpy, pandas, nltk, tensorflow, con LSTM, se limpió el conjunto de datos eliminando las palabras vacías, lematización y la extracción de características se usaron TF-IDF y se modificó el atributo ocurrió en demasiados documentos (max_df) con 0.75. Se dividió el conjunto de datos en 70 % para entrenamiento y 30 % pruebas y se utilizó el tipo de pérdida (loss), el optimizador de los pesos de las conexiones de las neuronas Adam. Se tomó como métricas de evaluación del clasificador el accuracy. La cantidad de iteraciones de aprendizaje de entrenamiento fue de 10 pasadas hacia adelante y 10 hacia atrás, para el tamaño de lote se estableció en 64.

En la Tabla 14 se muestra los resultados obtenidos de los experimentos realizados y los porcentajes que obtuvieron en modelos de las investigaciones el cual se seleccionaron los que mejores se desempeñaron por algoritmo entre ellos están passive-agresive, SVM, RF, LR y LSTM. Para el modelo con el algoritmo SVM que se usó datos recopilados de Twitter se perfilan con veintidós atributos y se realizó la eliminación de los datos de replicación, da más alto que el modelo de prueba esto es debido a que nuestro conjunto de datos aproximadamente 20 mil siendo este pequeño, contrario al que usaron en el modelo propuesto. Así mismo ocurrió passive-agresive y bosque aleatorio. Para el modelo LSTM el experimento se obtuvo un 99.71 % de exactitud, contrario al modelo propuesto esto es debido a que utilizan un modelo LSTM con bidireccional contrario al experimento que se realizó incrustación de palabra TF-IDF.

Tabla 14. Comparación de resultados con respecto a exactitud

Algoritmo	Modelos propuestos	Experimento
PASIVE AGRESIVE	100	85.94
SVM	99,90	91.62
RF	99	88,40
LR	97	97.67
LSTM	95,30	99.71

5. DISCUSIÓN

En respuesta a los objetivos planteados se realiza el análisis de los resultados alcanzados, así en la investigación realizada por los autores Kareem, y Awan que usaron técnicas de extracción de características frecuencia de término (*TF*) y frecuencia de documento inversa de frecuencia de término (*TF-IDF*) obtuvieron que el mejor clasificador fue *k*-vecinos más cercanos (*KNN*) con una precisión del 70% y la regresión logística de 69%; los que difieren de los resultados obtenidos en esta investigación. Se utilizó la herramienta de LIWC2015 para extraer características lingüísticas en la que se obtienen buenos puntajes con *k*-vecinos más cercanos con 88% y para regresión logística 97%.

De modo similar [7] los resultados mostraron que los modelos basados en árbol de decisión tuvieron una adecuada exactitud (*accuracy*) (74%) y precisión (74%). También, los algoritmos ZeroR, CVPS y WIHW, con un valor de 100% de recall, parecieron ser los mejores algoritmos en términos de métrica de recuperación; lo que difiere a los resultados utilizar la técnica de ensacado para reducir la tasa de error del clasificador. El obtenido fue mayor a 98% con el conjunto de datos de 44.898 artículos.

En Zhou y Zafarani [77], el clasificador que tuvo mejor resultado fue el de bosques aleatorios que funcionaron mejor en ambos conjuntos de datos, logrando una precisión de alrededor de 93 % en PolitiFact y alrededor de 84% en BuzzFeed; en esta investigación utilizaron la herramienta de LIWC2015 obteniendo mayor puntaje con árbol de decisión de 98%. Siendo esta herramienta poca conocida se la puede usar en detección de noticias falsas ya que da buenos resultados para diferentes algoritmos de aprendizaje automático.

Qawasmeh, Tawalbeh y Abdullah [25] propusieron dos modelos Bidireccional *LSTM* concatenado y Multi-head *LSTM*. Utilizaron el conjunto de datos *The Fake News Challenge* (FNC-1) determinando que es mejor usar Multi-head *LSTM* ya que proporcionó buena precisión y exactitud; en los resultados obtenidos las BI-*LSTM* no tuvieron buenos resultados, por la práctica que se realizó solo *LSTM* se puede obtener en los experimentos 99.71% utilizando la extracción de TF-IDF. Para los algoritmos de aprendizaje profundo, la parte de las extracciones de características tienen a dar mejores resultados con PCA ya que es más efectivo para la reducción severa de dimensionalidad.

Siguiendo la misma línea, Bahad, Saxena y Kamal [78] propusieron un modelo que utilizó la inclusión de palabras GloVe, de 4 algoritmos, el Bidireccional LSTM-RNN fue significativamente más efectivo que los modelos unidireccionales; los resultados obtenidos de la práctica se obtuvo buenos resultados con modelos unidireccionales con un puntaje de 99.71.

Otra de las técnicas usadas para la detección automática de noticias falsas es la técnica de minería de datos [79]. Los resultados demostraron que todos los modelos funcionaron mejor cuando se combinan con las secciones de minería de datos. Siendo *LSTM* aquel que tuvo mejores resultados en combinación con la representación de word2vec; por parte de esta técnica no se obtuvieron resultados, pero puede ser una técnica que se puede desarrollar a futuro y que pueden obtener patrones para que ayude a los sistemas de inteligencia artificial a identificar las noticias y que se evite la desinformación.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se concluye que se está trabajando en técnicas de aprendizaje automático y profundo para crear modelos que permitan la detección de noticias falsas. Los algoritmos que mejor se desempeñaron fue uno de los pocos usados en los trabajos el cual fue *passive agresive*. Dicho algoritmo tuvo porcentajes muy alentadores de 100%. Otro que algoritmo que cabe mencionar que es muy usado para la detección de noticias falsas es el algoritmo de máquina de vectores de soporte. Por parte del aprendizaje profundo el que mejor se desempeña es un modelo de red neuronal con 99.90 % de exactitud. Con esto los clasificadores se puede utilizar para crear sistemas que permita detectar noticias falsas, además se han creados modelos para que los usuarios puedan probarlos haciendo uso de la URL y que puedan verificar si la noticia es verdad, pero integrarlo en las redes sociales todavía es un reto debido a que para entrenar los modelos solo usan datos extraídos de dos redes sociales o sitios web.

Con la investigación se obtuvo que si se aplican técnicas de minería de texto los modelos aplican eliminación de palabras debido a que no aportan información importante y lo que ocasiona es que el rendimiento del clasificador sea bajo. Entre estas técnicas de eliminación de palabras estuvieron las eliminaciones de palabras vacías, derivación y tokenización. Por otra parte, está la extracción de característica TF-IDF, que es la más utilizada, pero algo interesante fue que la herramienta LIWC2015 puede dar buenos resultados ya que permite extraer diferentes características textuales de los artículos de noticias. Ésta permitió obtener los mayores porcentajes de exactitud en diferentes algoritmos por lo que usarlo puede beneficiar a ser más precisos en la detección de noticias falsas.

De acuerdo con los resultados de la revisión de la literatura, se concluye que las técnicas de inteligencia artificial evitan el contenido falso, los algoritmos de aprendizaje y herramientas de selección de características aportan significativamente en la detección de noticias falsas. En relación con la parte práctica, se obtuvieron puntajes mayores a 85% en los clasificadores probados. Se usó el mismo conjunto de datos para todas las pruebas y se usó el TF-IDF debido a que es una de las técnicas de extracción de características más utilizadas. Todos los modelos fueron entrenados de manera exitosa y fueron probados en Python. No obstante, ninguno de ellos fue probado en una red social como tal, es un

trabajo que no estaba considerado en este trabajo; pero que se podría llevar a cabo como trabajo futuro.

6.2. Recomendaciones

Para no caer en noticias falsas se recomienda revisar la fuente de publicación o que sean oficiales de los medios de comunicación que brindan información verificada. También poder verificarlas con sistemas de herramientas de inteligencia artificial existentes.

Por otro lado, para crear un clasificador de noticias falsas, se recomienda hacer uso de algoritmo de passive-agresive y máquinas de soporte con referente a aprendizaje automático, asimismo, se recomienda involucrar técnicas de preprocesamiento de texto y estudiar más la herramienta de LIWC que es muy alentadora y da resultados competitivos.

Clasificar una noticia falsa en las redes sociales puede involucrar muchos factores, pero con la ayuda de la inteligencia artificial puede ser posible crear modelos eficientes en donde las empresas deberían ver este campo importante para resolver este problema, se recomienda trabajar con una gran cantidad de datos del mundo real para los usuarios estén bien informados.

En vista que la difusión de noticias falsas causa un impacto negativo se recomienda trabajar con modelos no solo texto, sino también en otros formatos como imágenes, video o audio debido a que son muy comunes en difundirse en los medios sociales.

REFERENCIAS

- [1] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017, doi: 10.1257/jep.31.2.211.
- [2] C. Carvalho, N. Klagge, and E. Moench, “The persistent effects of a false news shock,” *J. Empir. Financ.*, vol. 18, no. 4, pp. 597–615, 2011, doi: 10.1016/j.jempfin.2011.03.003.
- [3] J. A. H.-T. Nathaly Álvarez-Daza, Pablo Pico-Valencia, “Detección de Noticias Falsas en Redes Sociales Basada en Aprendizaje Automático y Profundo. Una Breve Revisión Sistemática,” *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*
- [4] Ruth Gabriela Ortega Jaramillo and E. Luzuriaga, “Campañas Electorales y Fake News: Caso Consulta Popular y Referéndum Ecuador 2018,” vol. 4, no. 1, pp. 75–84, 2019, doi: .1037//0033-2909.I26.1.78.
- [5] R. Goyena and A. . Fallis, “Comportamientos generacionales y contenidos informativos que construyen el mundo de las Fake News en Ecuador,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004.
- [6] M. V. J. CARLOS, “Los fake news. Caso: paro nacional, Babahoyo 2019. Análisis de su influencia en los estudiantes de la carrera de comunicación social.,” 2020.
- [7] F. A. Ozbay and B. Alatas, “Fake news detection within online social media using supervised artificial intelligence algorithms,” *Phys. A Stat. Mech. its Appl.*, vol. 540, Feb. 2020, doi: 10.1016/j.physa.2019.123174.
- [8] D. O. Klein and J. R. Wueller, “A Legal Perspective: ‘What is fake news?’” *J. Internet Law*, vol. 20, no. 10, pp. 6–13, 2017, [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=iih&AN=122227863&lang=pt-br&site=ehost-live>.
- [9] D. M. J. Lazer *et al.*, “The science of fake news: Addressing fake news requires a multidisciplinary effort,” *Science (80-.)*, vol. 359, no. 6380, pp. 1094–1096, 2018, doi: 10.1126/science.aao2998.
- [10] J. L. Egelhofer and S. Lecheler, “Fake news as a two-dimensional phenomenon: a

- framework and research agenda,” *Ann. Int. Commun. Assoc.*, vol. 43, no. 2, pp. 97–116, 2019, doi: 10.1080/23808985.2019.1602782.
- [11] A. S. y V. B. Banic, “Caso Macedonia,” 2016. <https://www.nbcnews.com/news/world/fake-news-how-partying-macedonian-teen-earns-thousands-publishing-lies-n692451>.
- [12] J. Clavero, “Posverdad y exposición selectiva a fake news. Algunos ejemplos concretos de Argentina,” *Contratexto*, no. 029, pp. 167–180, 2018, doi: 10.26439/contratexto2018.n029.1857.
- [13] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media: A Data Mining Perspective.” [Online]. Available: <http://www.journalism.org/2016/05/26/news-use-across->.
- [14] Internetworldstats, “Estadísticas mundiales de Internet.” <https://www.internetworldstats.com/stats10.htm>.
- [15] J. E. García-Guerrero, “Social Networks and political interest,” *Icono14*, vol. 17, no. 2, pp. 231–253, 2019, doi: 10.7195/ri14.v17i2.1308.
- [16] INEC, “Tecnologías de la Información y Comunicación Contenido,” *Inec*, 2018, [Online]. Available: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2018/201812_Principales_resultados_TIC_Multi proposito.pdf.
- [17] H. Allcott, M. Gentzkow, and C. Yu, “Trends in the diffusion of misinformation on social media,” *Res. Polit.*, vol. 6, no. 2, 2019, doi: 10.1177/2053168019848554.
- [18] C. Silverman, “Teoría de la conspiración detrás de ‘Pizzagate,’” 2016. https://www.buzzfeed.com/craigsilverman/fever-swamp-election?utm_term=.rfNy7EWMKg#.eqla7JZY8j.
- [19] C. S. Atodiresei, A. Tănăselea, and A. Iftene, “Identifying Fake News and Fake Users on Twitter,” in *Procedia Computer Science*, 2018, vol. 126, pp. 451–461, doi: 10.1016/j.procs.2018.07.279.
- [20] Homebusinessmag, “¿Cuál es el impacto de las ‘noticias falsas’ en los precios del mercado de valores?,” 2017. <https://homebusinessmag.com/blog/money-matters/fake-news-impact-stock-market-prices/>.

- [21] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, “Faking Sandy: Characterizing and Identifying Fake Images on Twitter during Hurricane Sandy,” pp. 729–736, 2013, doi: 10.1145/2487788.2488033.
- [22] K. Kwang, “El Comité Selecto hace 22 recomendaciones para lidiar con la amenaza de noticias falsas a Singapur,” 2018. <https://www.channelnewsasia.com/news/singapore/select-committee-fake-news-online-falsehoods-recommendations-10739834>.
- [23] F. B. y P. O. Cristina Tardáguila, “Las noticias falsas están envenenando la política brasileña.” <https://www.nytimes.com/2018/10/17/opinion/brazil-election-fake-news-whatsapp.html>.
- [24] R. K. Kaliyar, A. Goswami, and P. Narang, “Multiclass Fake News Detection using Ensemble Machine Learning,” *Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019*, pp. 103–107, 2019, doi: 10.1109/IACC48062.2019.8971579.
- [25] E. Qawasmeh, M. Tawalbeh, and M. Abdullah, “Automatic Identification of Fake News Using Deep Learning,” *2019 6th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2019*, pp. 383–388, 2019, doi: 10.1109/SNAMS.2019.8931873.
- [26] C. Choque, “Minería de Datos para construir un modelo de minería de datos para recomendación de ofertas laborales,” no. 052, pp. 1–18, 2015.
- [27] L. Kumar and P. K. Bhatia, “Available Online at www.jgrcs.info text mining : concepts , process and applications,” vol. 4, no. 3, pp. 36–39, 2013.
- [28] S. VijayGaikwad, A. Chaugule, and P. Patil, “Text Mining Methods and Techniques,” *Int. J. Comput. Appl.*, vol. 85, no. 17, pp. 42–45, 2014, doi: 10.5120/14937-3507.
- [29] T. Mikolov, “Word2vec,” 2014. <https://code.google.com/archive/p/word2vec/>.
- [30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.
- [31] Y. Porras-Garcia, R. Calderon-Moreno, and A. Cruz-Roa, “Análisis de Desempeño Computacional del Procesamiento Distribuido de una Implementación de Bolsa de Palabras en Apache SparkTM,” *2018 IEEE Colomb. Conf. Commun. Comput.*

- COLCOM 2018 - Proc.*, pp. 1–6, 2018, doi: 10.1109/ColComCon.2018.8466725.
- [32] M. del C. J. de la Torre, *Nuevas Tecnicas De Minenia De Textos: Aplicaciones*. 2017.
- [33] G. Liberatore, A. Vuotto, and G. Fernández, “Desarrollo de una herramienta para el análisis y representación semántica de colecciones documentales a través del factor TF-IDF Se describe el desarrollo y aplicación de un plugin para el análisis semántico de u,” *VII Jornadas Temáticas Actuales en Bibl.*, 2016, [Online]. Available: <http://humadoc.mdp.edu.ar:8080/xmlui/handle/123456789/631>.
- [34] I. Kareem and S. M. Awan, “Pakistani Media Fake News Classification using Machine Learning Classifiers,” *3rd Int. Conf. Innov. Comput. ICIC 2019*, no. Icic, pp. 1–6, 2019, doi: 10.1109/ICIC48496.2019.8966734.
- [35] S. Deegalla and H. Boströmt, “Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification,” *Publ. Finnish Artif. Intell. Soc.*, pp. 23–30, 2006.
- [36] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, “Fake news stance detection using deep learning architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.
- [37] Y. Karla, “Construcción de una memoria organizacional a partir de textos no estructurados usando herramientas de minería de texto,” p. 71, 2016, [Online]. Available: <https://core.ac.uk/download/pdf/47250428.pdf>.
- [38] M. F. Correa Cordero, “Adaptación de una herramienta de procesamiento de lenguaje natural para el etiquetado de sentimientos y el análisis de lenguaje en español,” pp. 1–4, 2015.
- [39] Megaputer, “TextAnalys.” <https://www.megaputer.com/what-is-text-analytics/>.
- [40] F. S. Jadidian, “Applications of Named Entity Recognition in Customer Relationship Management Systems,” no. September, 2014.
- [41] E. Ezequiel and B. Pareja, “Minería de texto para el análisis de la colaboración en SIENA,” 2014.
- [42] Gate, “Gate.” <https://gate.ac.uk/sale/tao/splitch1.html#x4-40001>.

- [43] M. E. G. PICO, “desarrollo de una herramienta de minería de datos para el análisis de influencia de cuentas automatizadas en temas de tendencia sobre la opinión de los usuarios de twitter en ecuador,” 2018.
- [44] D. Cedeno-Moreno and M. Vargas, “Aprendizaje automático aplicado al análisis de sentimientos,” *I+D Tecnológico*, vol. 16, no. 2, 2020, doi: 10.33412/idt.v16.2.2833.
- [45] J. A. Neria, “Análisis de incertidumbre para resultados de código Termohidráulicos de mejor estimación,” 2010.
- [46] J. Lilja and M. Ostling, “A comparison of machine learning algorithms for automatic classification of neurons by their morphology,” pp. 1–24, 2018, [Online]. Available: <https://pdfs.semanticscholar.org/8f81/ca954a766797d584dfac5b19db08ae0ca45b.pdf>.
- [47] R. Castillo, J. Junior, M. Ochoa, and M. Madeleine, “Desarrollo De Una Aplicación Informática Basada En Un Modelo De Machine Learning Para Mejorar La Evaluación De Préstamos Crediticios,” 2017, [Online]. Available: http://refi.upnorte.edu.pe/bitstream/handle/11537/12294/Rodríguez_Castillo%2C_Jorge_Junior_-_Miñano_Ochoa%2C_Milagros_Madeleine.pdf?sequence=1&isAllowed=y.
- [48] J. P. Mestre, “La metodología cuantitativa aplicada al estudio de la reincidencia en menores infractores,” *Sci. Med. (Phila)*, vol. 6, no. 6, pp. 18–27, 1999, [Online]. Available: https://www.tdx.cat/bitstream/handle/10803/432779/2016_Tesis_Pallares_Mestre_Jacinto.pdf?sequence=1.
- [49] C. Fan, “Classifying Fake News,” pp. 1–14, 2017, [Online]. Available: <http://www.conniefan.com/wp-content/uploads/2017/03/classifying-fake-news.pdf>.
- [50] K. Jolly, *Machine Learning with scikit-learn Quick Start Guide: Classification, regression, and clustering techniques in Python*. 2018.
- [51] V. R. Porcada, “Clasificación Supervisada basada en Tesis Doctoral Clasificación Supervisada basada en Computacional Tesis Doctoral,” 2003.

- [52] A. Cordova and K. Torres, “Aplicación de minería de datos para pronosticar el riesgo de morosidad de los estudiantes de la universidad autónoma del Perú,” 2018.
- [53] U. P. Unión, “Modelo de Árboles de decisión para pronosticar la morosidad de los alumnos de la Resúmen Revista de Business Intelligence | MAYO 2011 Revista de Business Intelligence | MAYO 2011,” pp. 26–32, 2011.
- [54] Y. L. L. Manuel, “Diseño de un Modelo de Predicción de Éxito para Proyectos Tecnológicos con Financiación en Masa Aplicando Técnicas de Machine Learning Yagual López Luis Manuel Ing . Gallardo Posligua Vicente Adolfo , Mgs .,” 2019.
- [55] K. P. Murphy, *Machine learning: a probabilistic perspective*. 2012.
- [56] J. H. Gallego Orrego and C. A. Rodríguez Londoño, “Métodos de Aprendizaje de Máquina para la Detección de Arritmias Cardíacas en Tiempo Real,” *Univ. Pontif. Boliv.*, p. 78, 2013.
- [57] L. A. Aguilar Gutiérrez and Y. O. Vásquez Valdivia, “Principal Component Analysis (PCA) para mejorar la performance de aprendizaje de los algoritmos Support Vector Machine (SVM) y Red Neuronal Multicapa (MLNN),” *Tesis*, p. 139, 2016.
- [58] V. A. E. A. Jácome Escobar José David, “Desarrollo de una aplicación para detección de patrones en imágenes mediante el uso de aprendizaje profundo,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.
- [59] Á. De La Cruz Casado Manuel Oreja Valverde Alejandro Pascua Piña Marcos Robles Palencia and G. Botella Juan, “Técnicas de Machine Learning para Conducción,” 2018, [Online]. Available: <https://eprints.ucm.es/56572/>.
- [60] Apache, “Apache.” <http://singa.apache.org/>.
- [61] y E. F. Sean Owen, Robin Anil, Ted Dunning, *Mahout in Action*. 2012.
- [62] L. Deng and D. Yu, “Deep learning: Methods and applications,” *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/20000000039.
- [63] T. D. E. L. Tfg, T. Autor, E. Salam, and S. J. Data, “Deep learning para el reconocimiento facial de emociones básicas,” 2019.
- [64] X. B. Olabe, “Redes neuronales artificiales y sus apl,” [Online]. Available:

http://cvb.ehu.es/open_course_ware/castellano/tecnicas/redes_neuro/contenidos/pdf/libro-del-curso.pdf.

- [65] O. N. Sánchez, “Redes neuronales recurrentes: Principios y aplicaciones,” 2006.
- [66] M. P. Cuéllar, Sistemas evolutivos para entrenamiento y optimización de modelos neuronales recurrentes dinámicos. Aplicación en modelado y predicción de series de datos., vol. 4, no. 1. 2016.
- [67] F. B. Gereme and W. Zhu, “Early detection of fake news “before it flies high,” in *ACM International Conference Proceeding Series*, Aug. 2019, pp. 142–148, doi: 10.1145/3358528.3358567.
- [68] F. J. Ascarza Mendoza, “Segmentación Automática de Textos, mediante Redes Neuronales Convolucionales en Imágenes Documentos Históricas,” p. 67, 2018, [Online]. Available: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/13050>.
- [69] Y. B. Rodríguez, “Integración de la red neuronal convolucional con el algoritmo de función de frontera de objeto para reconocimiento de piezas y detección de defectos.,” 2018, [Online]. Available: [https://comimsa.repositorioinstitucional.mx/jspui/bitstream/1022/66/1/Monografía a FIDENCIOJGM 09-2010-0071.pdf](https://comimsa.repositorioinstitucional.mx/jspui/bitstream/1022/66/1/Monografía%20a%20FIDENCIOJGM%2009-2010-0071.pdf).
- [70] J. Hochreiter, S. and Schmidhuber, “Long Short-Term Memory, Neural Computation,” 1997.
- [71] J. Antonio and P. Ortiz, “Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto. Juan Antonio Pérez Ortiz Tesis doctorales de la Universidad de Alicante. Tesi doctoral de la Universitat d’Alacant. 2002,” 2002.
- [72] P. Loncomilla, “Deep learning: Redes convolucionales,” *Recuper. https://ccc. Ina. mx/~ pgomez/deep/presentations*, 2016.
- [73] Waikato, “Weka.” <https://www.cs.waikato.ac.nz/ml/weka/>.
- [74] Á. M. A. R. Jose Antonio Garcia Bermúdez, “Análisis para predicción de ventas utilizando minería de datos en almacenes de ventas de grandes superficies,” *Int. Inst. Environ. Dev.*, vol. 07/80, no. 2, p. 125, 2010, [Online]. Available: <https://arxiv.org/pdf/1707.06526.pdf>⁰<https://www.yrpri.org>⁰<http://weekly>

- .cnbnews.com/news/article.html?no=124000%0Ahttps://www.fordfoundation.org/%0Ahttp://bibliotecavirtual.clacso.org.ar/Republica_Dominicana/ccp/20120731051903/rep%0Ahttp://webpc.cia.
- [75] O. for I. Knime, “Knime.” <https://www.knime.com/software-overview>.
- [76] Stratebi, “Comparativa de Algoritmos de Herramientas de Data mining,” *Strat. open Bus. Intell.*, no. Agosto, 2010.
- [77] X. Zhou and R. Zafarani, “Network-based Fake News Detection: A Pattern-driven Approach.” [Online]. Available: <https://www.snopes.com/>.
- [78] P. Bahad, P. Saxena, and R. Kamal, *Fake News Detection using Bi-directional LSTM-Recurrent Neural Network*, vol. 165. Elsevier B.V., 2019.
- [79] S. Deepak and B. Chitturi, “Deep neural approach to Fake-News identification,” in *Procedia Computer Science*, 2020, vol. 167, pp. 2236–2243, doi: 10.1016/j.procs.2020.03.276.
- [80] S. M. Jang *et al.*, “A computational approach for examining the roots and spreading patterns of fake news: Evolution tree analysis,” *Comput. Human Behav.*, vol. 84, pp. 103–113, Jul. 2018, doi: 10.1016/j.chb.2018.02.032.
- [81] I. N. G. Hugo and D. E. L. Pozo, “Ley orgánica de comunicación 24,” pp. 1–24, 2013.
- [82] Asamblea del Ecuador, “Constitución del Ecuador,” *Regist. Of.*, no. 20 de Octubre, p. 173, 2008, doi: 10.1017/CBO9781107415324.004.
- [83] A. Nacional., “Código Orgánico Integral Penal- Ley 0,” *Editor. Nac.*, p. 144, 2014.
- [84] M. M. Kai Petersen, Robert Feldt, Shahid Mujtaba, “Systematic Mapping Studies in Software Engineering,” *Pediatr. Neonatol.*, vol. 58, no. 6, pp. 484–489, 2017, doi: 10.1016/j.pedneo.2016.08.011.
- [85] T. Saikh, B. Haripriya, A. Ekbal, and P. Bhattacharyya, “A Deep Transfer Learning Approach for Fake News Detection,” *Proc. Int. Jt. Conf. Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9207477.
- [86] R. R. Mandical, N. Mamatha, N. Shivakumar, R. Monica, and A. N. Krishna, “Identification of Fake News Using Machine Learning,” *Proc. CONECCT 2020 -*

- 6th *IEEE Int. Conf. Electron. Comput. Commun. Technol.*, 2020, doi: 10.1109/CONECCT50063.2020.9198610.
- [87] N. Smitha and R. Bharath, “Performance Comparison of Machine Learning Classifiers for Fake News Detection,” *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 696–700, 2020, doi: 10.1109/ICIRCA48905.2020.9183072.
- [88] V. Tiwari, R. G. Lennon, and T. Dowling, “Not Everything You Read Is True! Fake News Detection using Machine learning Algorithms,” *2020 31st Irish Signals Syst. Conf. ISSC 2020*, 2020, doi: 10.1109/ISSC49989.2020.9180206.
- [89] S. H. Kong, L. M. Tan, K. H. Gan, and N. H. Samsudin, “Fake News Detection using Deep Learning,” *ISCAIE 2020 - IEEE 10th Symp. Comput. Appl. Ind. Electron.*, no. M1, pp. 102–107, 2020, doi: 10.1109/ISCAIE47305.2020.9108841.
- [90] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti, and M. Akbar, “Fake news detection using deep learning models: A novel approach,” *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, pp. 1–23, 2020, doi: 10.1002/ett.3767.
- [91] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, “DeClarE: Debunking fake news and false claims using evidence-aware deep learning,” *arXiv*, pp. 22–32, 2018.
- [92] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, “Fake News Detection Using Machine Learning Ensemble Methods,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/8885861.
- [93] R. A. Adrian M. P. Brasoveanu, “Integrating Machine Learning Techniques in Semantic Fake News Detection,” 2020.
- [94] A. Albahr and M. Albahar, “An empirical comparison of fake news detection using different machine learning algorithms,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, pp. 146–152, 2020, doi: 10.14569/IJACSA.2020.0110917.
- [95] S. M. Mohsen, C. A. Mousavi, Z. Soraya, and T. Jafar, “A predictive model based on machine learning methods to recognize fake Persian news on twitter,” *Int. J. Nonlinear Anal. Appl.*, vol. 11, no. Special Issue, pp. 119–128, 2020, doi: 10.22075/ijnaa.2019.4429.

- [96] M. D. Ibrishimova and K. F. Li, “A machine learning approach to fake news detection using knowledge verification and natural language processing,” *Adv. Intell. Syst. Comput.*, vol. 1035, pp. 223–234, 2020, doi: 10.1007/978-3-030-29035-1_22.
- [97] W. Han and V. Mehta, “Fake news detection in social networks using machine learning and deep learning: Performance evaluation,” *Proc. - IEEE Int. Conf. Ind. Internet Cloud, ICII 2019*, no. Icii, pp. 375–380, 2019, doi: 10.1109/ICII.2019.00070.
- [98] V. M. Kresnakova, M. Sarnovsky, and P. Butka, “Deep learning methods for Fake News detection,” *IEEE Jt. 19th Int. Symp. Comput. Intell. Informatics 7th Int. Conf. Recent Achiev. Mechatronics, Autom. Comput. Sci. Robot. CINTI-MACRo 2019 - Proc.*, pp. 143–148, 2019, doi: 10.1109/CINTI-MACRo49179.2019.9105317.
- [99] B. M. Amine, A. Drif, and S. Giordano, “Merging deep learning model for fake news detection,” *2019 Int. Conf. Adv. Electr. Eng. ICAEE 2019*, pp. 5–8, 2019, doi: 10.1109/ICAEE47123.2019.9015097.
- [100] A. Abedalla, A. Al-Sadi, and M. Abdullah, “A closer look at fake news detection: A deep learning perspective,” *ACM Int. Conf. Proceeding Ser.*, pp. 24–28, 2019, doi: 10.1145/3369114.3369149.
- [101] D. Katsaros, G. Stavropoulos, and I. Bridge, “Which machine learning paradigm for fake news detection?,” pp. 383–387, 2019.
- [102] M. S. Mokhtar, Y. Y. Jusoh, N. Admodisastro, N. Pa, and A. Y. Amruddin, “Fakebuster: Fake news detection system using logistic regression technique in machine learning,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 2407–2410, 2019, doi: 10.35940/ijeat.A2633.109119.
- [103] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, “A smart System for Fake News Detection Using Machine Learning,” *IEEE Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2019*, pp. 2–5, 2019, doi: 10.1109/ICICT46931.2019.8977659.
- [104] A. Kumar, S. Singh, and G. Kaur, “Fake news detection of Indian and United States election data using machine learning algorithm,” *Int. J. Innov. Technol. Explor.*

- Eng.*, vol. 8, no. 11, pp. 1559–1563, 2019, doi: 10.35940/ijitee.K1829.0981119.
- [105] A. Verma, V. Mittal, and S. Dawn, “FIND: Fake Information and News Detections using Deep Learning,” *2019 12th Int. Conf. Contemp. Comput. IC3 2019*, pp. 1–7, 2019, doi: 10.1109/IC3.2019.8844892.
- [106] A. Lakshmanarao, Y. Swathi, and T. Srinivasa Ravi Kiran, “An effecient fake news detection system using machine learning,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10, pp. 3125–3129, 2019, doi: 10.35940/ijitee.J9453.0881019.
- [107] L. Borges, B. Martins, and P. Calado, “Combining similarity features and deep representation learning for stance detection in the context of checking fake news,” *J. Data Inf. Qual.*, vol. 11, no. 3, 2019, doi: 10.1145/3287763.
- [108] C.-S. J. Ye-Chan Ahn, “Natural Language Contents Evaluation System for Detecting Fake News using Deep Learning,” *ペインクリニック学会治療指針 2*, no. February, pp. 1–9, 2019, doi: .1037//0033-2909.I26.1.78.
- [109] R. Barua, R. Maity, D. Minj, T. Barua, and A. K. Layek, “F-NAD: An Application for Fake News Article Detection using Machine Learning Techniques,” *2019 IEEE Bombay Sect. Signal. Conf. IBSSC 2019*, vol. 2019Januar, pp. 0–5, 2019, doi: 10.1109/IBSSC47189.2019.8973059.
- [110] Abdullah-All-Tanvir, E. M. Mahir, S. Akhter, and M. R. Huq, “Detecting Fake News using Machine Learning and Deep Learning Algorithms,” *2019 7th Int. Conf. Smart Comput. Commun. ICSCC 2019*, pp. 1–5, 2019, doi: 10.1109/ICSCC.2019.8843612.
- [111] K. Poddar, G. B. D. Amali, and K. S. Umadevi, “Comparison of Various Machine Learning Models for Accurate Detection of Fake News,” *2019 Innov. Power Adv. Comput. Technol. i-PACT 2019*, pp. 1–5, 2019, doi: 10.1109/i-PACT44901.2019.8960044.
- [112] K. Agarwalla, S. Nandan, V. A. Nair, and D. Deva Hema, “Fake news detection using machine learning and natural language processing,” *Int. J. Recent Technol. Eng.*, vol. 7, no. 6, pp. 844–847, 2019.
- [113] E. A. Girgis, Sherry and M. Gadallah, “Deep learning algorithms for detecting fake news in online text,” *2018 13th Int. Conf. Comput. Eng. Syst.*, pp. 93–97, 2018.

- [114] P. C. Supanya Aphiwongsophon, “Detecting fake news with machine learning methd,” pp. 528–531, 2018.
- [115] y Y.-J. Y. Dong-Ho Lee, Yu-Ri Kim, Hyeong-Jun Kim, Seung-Myun Park, “Fake News Detection Using Deep Learning,” *arXiv*, vol. 15, no. 5, pp. 1119–1130, 2019, doi: 10.4018/978-1-7998-7291-7.ch025.
- [116] F. Ahmad and R. Lokeshkumar, “A comparison of machine learning algorithms in fake news detection,” *Int. J. Emerg. Technol.*, vol. 10, no. 4, pp. 177–183, 2019.
- [117] y M. G. Arvinder Pal Singh Bali, Mexson Fernandes, Sourabh Choubey and Asia, *Comparative Performance of Machine Learning Algorithms for Fake News Detection*, vol. 1046, no. July. Springer Singapore, 2019.
- [118] y R. A. Adrian M. P. Brasoveanu, “Semantic Fake News Detection: A Machine Learning Perspective,” *Springer Nat. Switz.*, vol. 2, no. June, pp. 283–296, 2019, doi: 10.1007/978-3-030-20521-8.
- [119] S. C. Arvinder Pal Singh Bali, Mexson Fernandes and and M. Goel, *Comparative Performance of Machine Learning Algorithms for Fake News Detection*, vol. 1046, no. July. Springer Singapore, 2019.
- [120] S. Gilda, “Evaluating machine learning algorithms for fake news detection,” *IEEE Student Conf. Res. Dev. Inspiring Technol. Humanit. SCORED 2017 - Proc.*, vol. 2018-Janua, pp. 110–115, 2018, doi: 10.1109/SCORED.2017.8305411.
- [121] R. Masood and A. Aker, “The fake news challenge: Stance detection using traditional machine learning approaches,” *IC3K 2018 - Proc. 10th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag.*, vol. 3, no. Kmis, pp. 128–135, 2018, doi: 10.5220/0006898801280135.
- [122] 2 Gerardo Ernesto Rolong Agudelo1, Octavio José Salcedo Parra1 and and J. B. Velandia, “Raising a Model for Fake News Detection Using Machine Learning in Python,” *Challenges Oppor. Digit. Era*, vol. 11195, no. October, pp. 325–336, 2018, doi: 10.1007/978-3-030-02131-3.
- [123] H. Ahmed, I. Traore, and S. Saad, “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques,” *First Int. Conf. Intelligent, Secur. Dependable Syst. Distrib. Cloud Environ.*, vol. 10618, pp. 169–181, 2017, doi:

10.1007/978-3-319-69155-8.

[124] “Kaggle.” <https://www.Kaggle.com/c/fake-news/data?select=train.csv>.

[125] R. Rodríguez, “Redes Neuronales Artificiales. Conceptos básicos y algunas aplicaciones en Energía,” no. June, p. 1, 2017, [Online]. Available: <https://www.researchgate.net/publication/317601698>.

