Análisis de Desastres en Redes Sociales: Clasificación de Tweets para la Identificación de Eventos Catastróficos mediante Modelos de Aprendizaje Profundo

González Emily y Coello Nayeli

*Facultad de Ciencias Matemáticas y Físicas*

*Universidad de Guayaquil* Guayaquil, Ecuador

***Resumen*—** Este proyecto aborda la tarea crítica de identificar eventos catastróficos a través del análisis de tweets en redes sociales. Se emplea un enfoque de aprendizaje profundo, evaluando modelos como 1D CNN, RNN, BiGRU y BiLSTM. Aunque todos demuestran un rendimiento comparable en F1 score, la eficiencia computacional se convierte en un factor determinante. Se destaca que el modelo 1D CNN, caracterizado por su velocidad tanto en entrenamiento como en inferencia, supera a sus contrapartes más complejas.

El modelo elegido consiste en una capa de convolución, max pooling, flatten y capas densas. El preprocesamiento, que incluye tokenización con NLTK, padding de secuencias y división de conjuntos de entrenamiento y prueba con sklearn, contribuye al éxito del modelo final. La evaluación exhaustiva se lleva a cabo utilizando métricas clave como F1, precisión y recall, así como consideraciones prácticas como el tiempo de entrenamiento, tiempo de inferencia y tamaño del modelo. Las conclusiones indican que, para textos cortos, el modelo 1D CNN simple se posiciona como una alternativa eficiente, superando incluso a modelos más complejos como BERT. Este enfoque no solo demuestra un rendimiento satisfactorio en términos de métricas de evaluación, sino que también destaca por su velocidad tanto en el entrenamiento como en la inferencia. Estos hallazgos tienen implicaciones importantes para la detección temprana y eficaz de eventos catastróficos a través de plataformas de redes sociales.

***Palabras clave— Identificación de eventos catastróficos, análisis de tweets, aprendizaje profundo, 1D CNN, RNN, BiGRU, BiLSTM, eficiencia computacional, F1 score, velocidad de entrenamiento, tokenización, conjuntos de entrenamiento y prueba, NLTK, sklearn, rendimiento del modelo, métricas clave, BERT, detección temprana.***

***Abstract***— This project addresses the critical task of identifying catastrophic events through the analysis of tweets on social media. A deep learning approach is employed, evaluating models such as 1D CNN, RNN, BiGRU, and BiLSTM. Although all demonstrate comparable performance in F1 score, computational efficiency becomes a determining factor. It is emphasized that the 1D CNN model, characterized by its speed in both training and inference, outperforms its more complex counterparts.

The chosen model consists of a convolutional layer, max pooling, flatten, and dense layers. Preprocessing, including tokenization with NLTK, sequence padding, and training-test set splitting with sklearn, contributes to the success of the final model. Comprehensive evaluation is carried out using key metrics such as F1, precision, and recall, as well as practical considerations like training time, inference time, and model size. Conclusions indicate that, for short texts, the simple 1D CNN model emerges as an efficient alternative, surpassing even more complex models like BERT. This approach not only demonstrates satisfactory performance in terms of evaluation metrics but also stands out for its speed in both training and inference. These findings have significant implications for the early and effective detection of catastrophic events through social media platforms.

***Keywords— Identification of catastrophic events, tweet analysis, deep learning, 1D CNN, RNN, BiGRU, BiLSTM, computational efficiency, F1 score, training speed, tokenization, training-test sets, NLTK, sklearn, model performance, key metrics, BERT, early detection.***

# **INTRODUCCIÓN**

La creciente interconexión en redes sociales ha proporcionado una fuente inigualable de información en tiempo real [1]. En particular, el análisis de desastres en redes sociales se ha vuelto esencial para la identificación temprana de eventos catastróficos. Este proyecto se sumerge en la clasificación de tweets con el objetivo fundamental de prever si están vinculados a desastres, abordando así una clasificación binaria crucial. La relevancia de esta tarea radica en la capacidad de las redes sociales para reflejar rápidamente eventos de gran magnitud y su impacto en la sociedad a través de la implementación de diversos modelos de aprendizaje automático en un entorno de Google Colab.

Se han investigado varios modelos de aprendizaje profundo, incluidos 1D CNN, RNN, BiGRU y BiLSTM, para evaluar su capacidad para procesar y comprender el contenido de los tweets. La utilización de GloVe como embeddings de palabras ha sido esencial para capturar la complejidad semántica de los mensajes [2]. Aunque todos los modelos muestran un rendimiento similar en términos de puntuación F1 (aproximadamente 0,78), vale la pena señalar que 1D CNN sobresale en eficiencia de velocidad durante el entrenamiento y la inferencia.

El modelo final es una 1D CNN [3], configurada con capas convolucionales, agrupación máxima, aplanamiento y capas densas para optimizar su rendimiento. El preprocesamiento, incluido el empoderamiento con NLTK, el llenado de secuencias y la partición de entrenamiento/prueba con sklearn, es esencial para garantizar la calidad de los datos y la generalización del modelo.

La evaluación exhaustiva del modelo se ha llevado a cabo utilizando métricas clave como F1 [4], precisión, recall, así como consideraciones prácticas como el tiempo de entrenamiento, tiempo de inferencia y tamaño del modelo. Los resultados destacan la eficacia del 1D CNN, concluyendo que, para textos cortos como los tweets, un modelo simple puede ser suficiente en lugar de modelos más complejos como BERT.

En el ámbito social, La capacidad de anticiparse y responder rápidamente a ocasiones terribles, por ejemplo, cataclismos o emergencias de ayuda, a través del análisis de los tweets hace que una interpretación directa se convierta en una mejora significativa del bienestar abierto. Este enfoque puede proporcionar a los especialistas y a las asociaciones de personas de guardia los datos necesarios para organizar sus esfuerzos y prestar la ayuda necesaria.

Actualmente se están realizando numerosos estudios sobre el tema de análisis de desastres en las redes sociales. Los científicos de datos comenzaron a emplear algoritmos de aprendizaje automático para desarrollar modelos de predicción de desastres. Investigaciones anteriores han empleado datos históricos [5], datos de redes sociales [6] o noticias [7] para predecir o identificar eventos catastróficos en las redes sociales utilizando un algoritmo de aprendizaje automático.

Máquina modelos de aprendizaje basados ​​​​en redes neuronales artificiales (ANN) [8], redes bayesianas [9] y memoria a largo plazo basada en redes neuronales recurrentes (LSTM) [10] ya se han utilizado para este tipo de investigaciones.

Este proyecto no solo contribuye al campo del análisis de desastres en redes sociales, sino que también destaca la importancia de la eficiencia computacional al seleccionar modelos para tareas específicas. La aplicación práctica de este enfoque ofrece valiosas percepciones para la rápida identificación de eventos catastróficos a través del análisis de tweets en redes sociales.

# **MATERIAL Y MÉTODOS**

***A. Materiales***

***1) Conjunto de datos (Dataset):*** Para este proyecto se utilizó un conjunto de datos obtenidos desdeKaggle [11], el cual contiene información sobre tweets. Cada entrada en el conjunto de entrenamiento y prueba contiene información sobre un tweet, incluyendo el texto del tweet, la ubicación desde donde se envió (que puede estar en blanco) y una palabra clave asociada (también puede estar en blanco). El objetivo es predecir si un tweet trata sobre un desastre real, indicado por la variable target en train.csv, donde 1 representa un desastre real y 0 no.

Las columnas en los conjuntos de datos son las siguientes:

* id: Identificador único para cada tweet.
* text: El texto del tweet.
* location: La ubicación desde donde se envió el tweet (puede estar en blanco).
* keyword: Una palabra clave particular del tweet (puede estar en blanco).
* target: Solo presente en train.csv, indica si un tweet trata sobre un desastre real (1) o no (0).

***2) Entorno de desarrollo:*** Para llevar a cabo el proyecto, se optó por utilizar Google Colab [12], un servicio en la nube sin costo, que es compatible con la ejecución de Jupyter Notebooks. Este entorno es muy popular en la comunidad de aprendizaje automático, principalmente por su sencillez y capacidad de acceder a hardware potente, como GPUs y TPUs.

***3) Bibliotecas:***  El lenguaje de programación empleado fue Python [13] y se utilizaron las siguientes bibliotecas:

*Time:* Es un módulo que proporciona varias funciones para trabajar con operaciones relacionadas con el tiempo. Se puede simplemente importar este módulo y empezar a usarlo sin tener que instalar ningún módulo adicional.

*Nltk:* Es una muy útil biblioteca para procesar datos del lenguaje humano en Python. Proporciona una interfaz fácil de usar para procesar datos lingüísticos como corpus de texto, recursos léxicos, gramáticas, etc.

*Numpy:* Es una biblioteca versátil para el procesamiento de matrices que ofrece un objeto de matriz multidimensional eficiente y herramientas para manipular estas matrices. Se trata del paquete esencial para realizar cálculos científicos en Python.

*Pandas:* Es una robusta biblioteca de Python de código abierto diseñada para la manipulación y análisis eficiente de datos. Ofrece estructuras de datos y funciones poderosas para facilitar estas operaciones.

*Keras:* Es una interfaz de programación de aplicaciones (API) de aprendizaje profundo desarrollada por Google, diseñada para la implementación de redes neuronales. Escrita en Python, su propósito es simplificar la creación de redes neuronales, y, además, es compatible con el cálculo de varias redes neuronales en la parte posterior (backend).

*TensorFlow:* Es una biblioteca de aprendizaje automático de código abierto creada por Google. TensorFlow se emplea para la construcción y entrenamiento de modelos de aprendizaje profundo, ya que simplifica la creación de grafos computacionales y permite una ejecución eficaz en distintas plataformas de hardware.

*Seaborn:* Es una impresionante biblioteca de Python especializada en la visualización de gráficos estadísticos. Ofrece estilos visuales predefinidos y paletas de colores atractivas que mejoran la estética de los gráficos estadísticos.

*Matplotlib:* Es una destacada librería de Python diseñada para la representación visual de gráficos en dos dimensiones a partir de matrices. Matplotlib se caracteriza por ser una herramienta de visualización de datos versátil y multiplataforma, construida sobre las matrices de NumPy, y está especialmente diseñada para integrarse con la amplia pila de SciPy.

*Sklearn:* Es una librería de código abierto en Python que incorpora diversos algoritmos de aprendizaje automático, así como funciones para el preprocesamiento, la validación cruzada y la visualización. Todo ello se realiza mediante una interfaz unificada que simplifica su utilización.

***B. Métodos***

***1) Revisión Literaria***

***Aprendizaje profundo***

Las estrategias de aprendizaje profundo, por ejemplo, las redes neuronales artificiales profundas, utilizan varias capas de procesamiento para encontrar patrones y diseñar en índices de información excepcionalmente enormes. Cada capa obtiene una idea a partir de la información que amplían las capas resultantes; cuanto más alto es el nivel, más conceptuales son las ideas que se aprenden. No existe una regla rígida para el número de capas que se espera que comprendan el aprendizaje profundo, pero la mayoría de los especialistas coinciden en que se requieren más de dos. En el aprendizaje profundo, no se espera un preprocesamiento amplio de la información ni la extracción manual de elementos, ya que, en consecuencia, el modelo puede beneficiarse de la información proporcionada. [14]

***Análisis de sentimiento***

El análisis de sentimientos [15] es actualmente un recurso típico en la recopilación de investigaciones sobre entretenimiento en línea realizadas por organizaciones, marketing y expertos políticos. La investigación sobre el análisis de los sentimientos extrae datos de las palabras buenas y pesimistas del texto, del entorno de esas palabras y del diseño semántico del texto. Esta breve auditoría analiza específicamente los compromisos que la información semántica puede asumir con la tarea de decidir en consecuencia el sentimiento.

Li et al. [16] se centraron en el efecto de la calidad de los datos en la ejecución de la clasificación de sentimientos. Pensaron en tres medidas, en concreto la información, la claridad y la subjetividad, para evaluar la naturaleza de las auditorías de artículos en línea.

El análisis de sentimiento se convierte en un componente esencial de este proyecto al considerar la naturaleza de los tweets y su capacidad para expresar percepciones y emociones. Debido a que la tarea principal del proyecto es clasificar los tweets en dos categorías: aquellos que están relacionados con desastres y aquellos que no lo están, lo cual implica un enfoque de clasificación binaria, identificando así eventos catastróficos en las redes sociales a través de la interpretación expresadas en los tweets.

***Redes Neuronales Convolucionales (CNN)***

(CNN) [17] son análogas a las ANN habituales

ya que están hechos de neuronas que se automejoran mediante el aprendizaje. En cualquier caso, cada neurona obtendrá una información y realizará una actividad (como un escalar), elemento seguido de una capacidad no directa): la premisa de RNA infinitas. Desde los vectores de imágenes en bruto de información hasta el último resultado de la puntuación de la clase, toda la organización comunicará en cualquier caso una única capacidad de puntuación aguda (el peso). La última capa contendrá capacidades de desgracia relacionadas con las categorías, y en realidad se aplican los consejos estándar en general y los engaños creados para las ANN tradicionales.

***Redes Neuronales Recurrentes (RNN)***

Los RNN están hechos de capas altas guardadas estados con elementos no directos [18]. La construcción de estados polizones sirve como memoria de la organización y condición de la capa secreta a la vez en su estado pasado. Esta construcción permite a los RNN almacenar, recuperar y procesar señales complejas durante largos períodos de tiempo. Los RNN pueden mapear un arreglo de información a la sucesión resultante en el momento actual paso de tiempo y anticipar el acuerdo en el siguiente paso de tiempo.

**CNN en la Clasificación de Tweets**

Los sitios web de microblogging como Twitter, en el transcurso de los últimos diez años, se han convertido en un canal de correspondencia proactivo durante sindicatos masivos y ocasiones de crisis, particularmente en situaciones de emergencia [19].

Liberando numerosos niveles de datos relacionados con la asombrosa cantidad de información de entretenimiento virtual producida durante tales circunstancias. Aunque sigue siendo un desafío increíble para las poblaciones afectadas por una catástrofe.

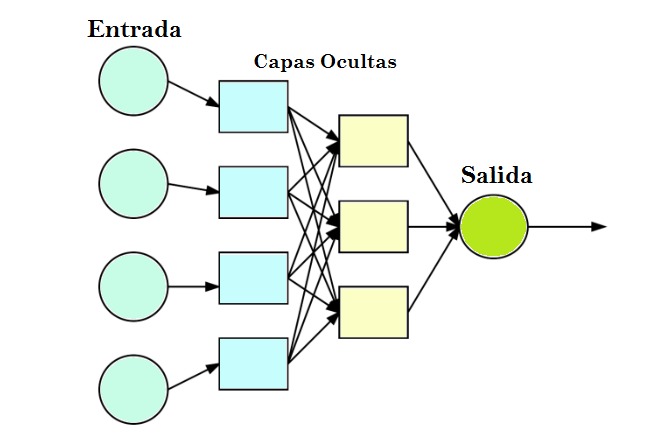
La información, aislada en varias clasificaciones útiles, puede ser utilizada por organizaciones gubernamentales, organizaciones filantrópicas, así como a los residentes para lograr una reacción más rápida en áreas de necesidad. En este trabajo, abordamos la situación anterior fomentando una profunda Organización Cerebral Convolucional (CNN) para la caracterización de emergencias con varios nombres, tuits relacionados. Ampliamos la CNN profunda con algunos elementos etimológicos separados de los tweets y la investigación de sus usos en la caracterización. La evaluación de un conjunto de datos de referencia muestra que nuestro enfoque propuesto logra la mejor ejecución de su clase.

**Hiperparámetros Específicos para el Análisis de Desastres en Redes Sociales**

* **Tamaño del filtro en capa plegable:** dado que los tweets son textos breves, el tamaño de los filtros en la capa convolucional afectará la capacidad del modelo para capturar patrones de texto específicos y relevantes.
* **Número de neuronas en una capa densa:** ajustar la cantidad de neuronas en la capa densa nos permitirá controlar la complejidad del modelo y su capacidad para extraer características importantes del texto del tweet.
* **Longitud máxima de secuencia:** dado que los tweets pueden tener diferentes longitudes, es importante establecer una longitud máxima de secuencia para un procesamiento de datos eficiente y garantizar que la red pueda manejar tweets de diferentes longitudes.
* **Dimensiones de incrustación de palabras:** la elección de las dimensiones de incrustación de palabras afectará la capacidad del modelo para comprender la semántica de los tweets y capturar información relevante.
* **Número de capas RNN (Red neuronal recurrente) o Red neuronal bidireccional (BiRNN):** Ajustar el número de capas en la red recurrente permitirá explorar la profundidad necesaria para capturar las dependencias temporales y contextuales en los tweets.

**Arquitectura de la Red Neuronal**

La arquitectura de una red neuronal alude al plan de juego y asociación de las distintas capas y neuronas que componen la organización. Básicamente, describe cómo los datos viajan a través de la organización, desde las capas de información hasta la capa de resultados, pasando por algunas capas secretas en el camino.



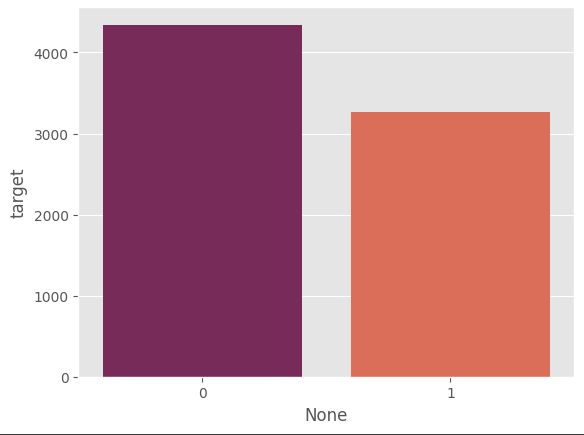
*Ilustración 1. Arquitectura de la red neuronal*

# **PRÁCTICA**

***Preprocesamiento de datos***

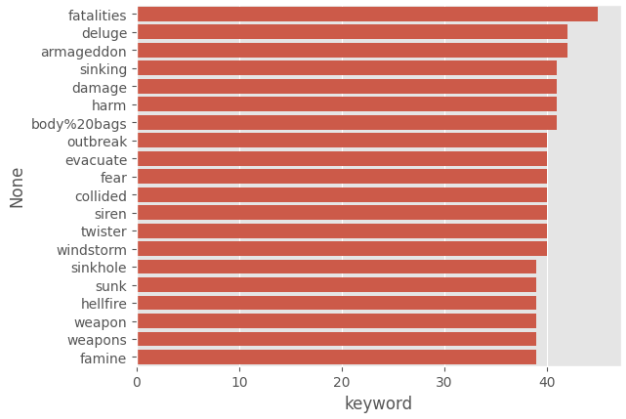
Antes de entrenar el modelo de análisis de desastres en redes sociales, se definieron los siguientes pasos de procesamiento de datos:

***División de conjuntos de datos:*** [20] Esto permite preparar el modelo en una colección informativa y evaluar su presentación en un conjunto gratuito para observar realmente las especulaciones. Para asegurar que la evaluación del modelo de clasificación sea precisa y justa, los datos de entrenamiento (padded\_sentences) y las etiquetas (train\_label) se dividen en conjuntos de entrenamiento y prueba usando train\_test\_split de scikit-learn.

******

*Ilustración 2. División de datos*

* Conjunto de entrenamiento (train): 80% de los datos
* Conjunto de prueba (test): 20% de los datos.

******

*Ilustración 3. Datos*

***Tokenización:*** se ajusta a las oraciones de entrenamiento (train\_sent), y se determina el tamaño del vocabulario.

***Embedding:*** toma un corpus y lo convierte en secuencias de enteros utilizando el tokenizador ajustado previamente. Esto convierte esencialmente los datos de texto en secuencias de enteros basadas en el vocabulario aprendido a partir de los datos de entrenamiento.

***Funciones de Métricas y Gráficos:*** La función metrics calcula e imprime la puntuación F1, precisión y recuperación entre las etiquetas reales y las predichas. La función plot se utiliza para trazar el historial de entrenamiento.

***Padding de Oraciones:*** Se identifica la oración más larga en los datos de entrenamiento (longest\_train) usando la función word\_tokenize de NLTK, y luego todas las oraciones se rellenan hasta esta longitud utilizando la función pad\_sequences de Keras.

***Carga de los Vectores de Palabras GloVe:*** Se cargan los vectores de palabras GloVe desde el archivo 'glove.6B.100d.txt' y se almacenan en el diccionario embeddings\_dictionary. Cada palabra en el archivo GloVe está asociada con un vector de 100 dimensiones.

***Creación de la Matriz de Embedding:*** Se crea una matriz de embedding donde cada fila corresponde a una palabra en el vocabulario del tokenizador, y los valores son los vectores de palabras GloVe. Si una palabra no está presente en el archivo GloVe, su fila permanece como un vector de ceros.

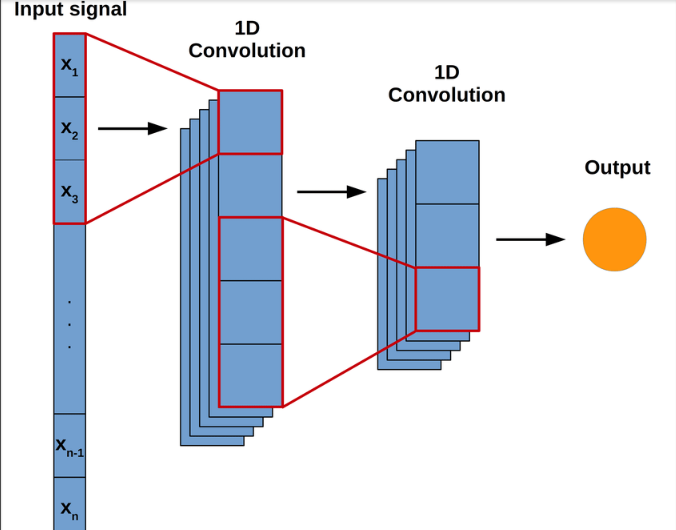
***Modelos***

A continuación, se presentan los modelos que se implementaron en el proyecto, el objetivo es probar diferentes técnicas y evaluar cual funciona mejor con los datos de los tuits.

* ***1D CNN***

Recientemente, se han propuesto modelos CNN unidimensionales para tareas de predicción relacionadas con series temporales [21], como el monitoreo instantáneo de la salud, la detección de fallas del motor y la clasificación de cultivos multitemporales con resultados alentadores.

En este proyecto, se utiliza una red convolucional de una dimensión (1D CNN) con capas de convolución, max pooling, flatten y capas densas.

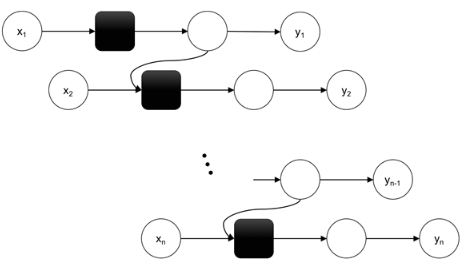


*Ilustración 4. Modelo 1D CNN*

* ***RNN***

Una red neuronal recurrente (RNN) consta de un estado ocultos h y un estado de salida opcional, los cuales operan con secuencias de longitud variable [22]. Los RNN pueden aprender distribuciones de probabilidad en una secuencia entrenada para predecir un símbolo en secuencia.

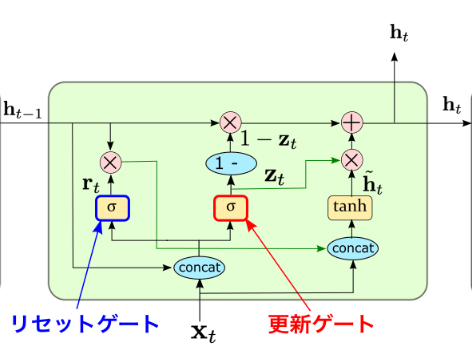
Para este proyecto, se utiliza una red neuronal recurrente simple bidireccional (BiRNN) con capas de embedding, bidireccional SimpleRNN, capa de pooling global, normalización por lotes, dropout y capas densas.



*Ilustración 5. Modelo RNN*

* ***BiGRU***

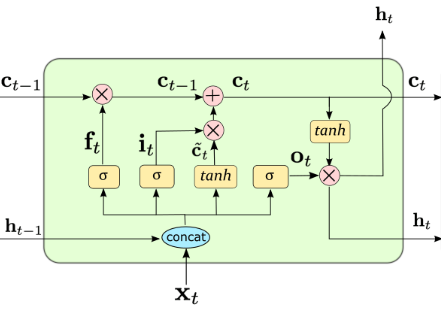
El modelo BiGRU es una versión simplificada del modelo LSTM [23]. BiGRU utiliza la diferencia entre la estructura de puerta y LSTM para combinar la puerta de entrada y la puerta de olvido en la puerta de actualización de LSTM. Tiene sólo dos estructuras de puerta, una puerta de reinicio y una puerta de actualización. El efecto de la última capa de salida oculta sobre la capa oculta actual está controlado por el puerto de actualización. Cuanto mayor sea el puerto de actualización, mayor será la influencia de la salida de la capa oculta anterior sobre la capa oculta actual. Es Similar al RNN, pero con capas de GRU en lugar de SimpleRNN.



*Ilustración 6. Modelo BiGRU*

* ***BiLSTM***

El modelo BiLSTM [24] trata todas las entradas por igual. En las tareas de análisis de sentimientos, la polaridad emocional de un texto depende en gran medida de palabras con un mensaje emocional. Primero, los vectores de palabras ponderados se utilizan como entrada para el modelo BiLSTM y la salida del modelo BiLSTM se utiliza como representación del texto del informe. Es similar a BiGRU, pero con capas de LSTM en lugar de GRU.



*Ilustración 7. Modelo BiLSTM*

***Entrenamiento del modelo***

* Se utiliza la función training (model, model\_name) para entrenar cada modelo.
* Se configuran callbacks como ModelCheckpoint, ReduceLROnPlateau, y EarlyStopping para optimizar el entrenamiento y prevenir el sobreajuste.

***Evaluación del modelo***

Finalmente, se procedió a la evaluación de los modelos utilizando un conjunto de datos de prueba independiente. La métrica de precisión de la clasificación fue empleada para evaluar la eficacia del modelo. Adicionalmente, se crearon matrices de confusión para proporcionar una representación visual del desempeño del modelo en cada categoría de análisis de desastres en redes sociales.

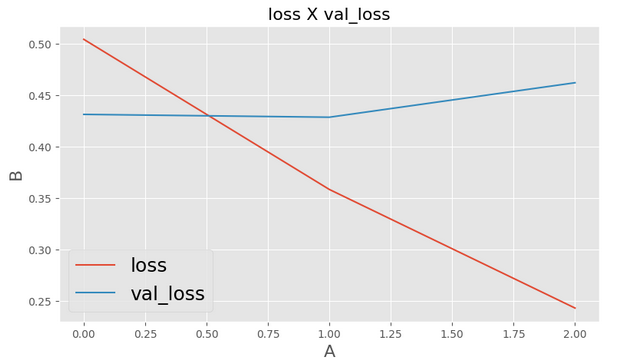
* Se evalúan métricas clave (precisión, recall, F1 score) en el conjunto de prueba.
* Se genera un gráfico de la pérdida y la precisión durante el entrenamiento con la función plot(history, [['loss', 'val\_loss'], ['accuracy', 'val\_accuracy']])

# **RESULTADOS Y DISCUSIÓN**

Para la evaluación correcta de la capacidad de cada modelo se estableció de dos partes, la primera se presenta en gráficos de precisión y perdida realizados durante el entrenamiento [25]. Donde el eje horizontal incorpora el número de épocas que el entrenamiento utiliza, mientras que el eje vertical muestra la precisión y la pérdida correspondiente a cada gráfico [26]; dichos gráficos son para medir el porcentaje de predicciones correctas en base a todas las predicciones del modelo según su precisión y evaluar la calidad de las predicciones, además de ajustar los pesos del modelo para minimizarla teniendo en cuenta la pérdida para diferenciar errores [27].

Como segunda parte esta la matriz de confusión y la curva ROC que se encarga de evaluar el rendimiento del modelo entrenado [28]. La matriz de confusión permite medir en base a los resultados medirlo por un programa en paralelo con los resultados reales, dado que analiza diversas observaciones que se emparejan para dar como el desempeño de algoritmo [29]. La curvo de ROC representa la sensibilidad y especificidad para todos los posibles valores del corte, dando una imagen general de la presión de la prueba [30].

1. ***Modelo CNN***
   1. ***Gráfica de perdidas***
      * **Eje x:** Los Epoch o épocas se representan numéricamente, con una secuencia que comienza en 0.0 y se extiende hasta 2.0 Los intervalos en el eje x están marcados en incrementos de 0.25.
      * **Eje y:** La trayectoria de la pérdida inicia en un valor aproximado de 0.5 y exhibe una tendencia decreciente conforme crece el número de epoch.



*Ilustración 8: Gráfica de perdidas del modelo CNN*

La evolución de la pérdida a lo largo de las épocas es un indicador crucial del rendimiento del modelo. En este caso, se observa una disminución constante en la pérdida en el conjunto de entrenamiento, lo cual es positivo [31]. La pérdida en el conjunto de validación también muestra una tendencia a la baja en las primeras dos épocas. Sin embargo, en la tercera época, la pérdida en el conjunto de validación aumenta ligeramente, indicando que podría haber un problema de sobreajuste [32].

* 1. ***Gráfica de la precisión***
     + **Eje x:** Se muestran diferentes niveles de precisión, desde 0 (indicando una precisión perfecta) hasta 2.0. Estos valores representan los resultados de diferentes iteraciones del modelo.
     + **Eje y:** Los números que van desde 0.76 hasta 0.90 en incrementos de 0.02 representan el número de épocas relacionados con el entrenamiento del modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

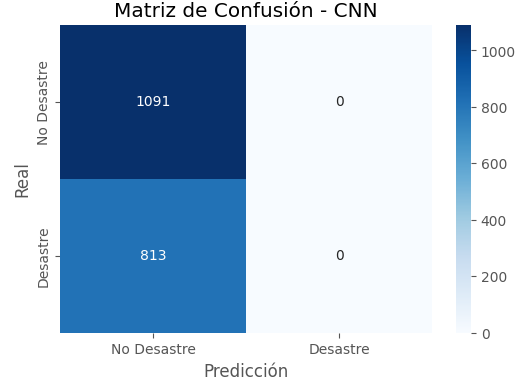
*Ilustración 9: Gráfica de precisión del modelo CNN*

La métrica de precisión (accuracy) en el conjunto de entrenamiento muestra que el modelo está aprendiendo bien los datos de entrenamiento, alcanzando una precisión del 90.54% en la tercera época. [33] La precisión en el conjunto de validación también es razonablemente alta (81.20%), pero la ligera subida en la pérdida podría requerir una revisión más detallada al momento del análisis del modelo [34].

* 1. ***Matriz de Confusión***

La Ilustración 8 presenta una matriz de confusión, correspondiente a un modelo de red convolucional diseñado para clasificar los tweets acerca si habla de un desastre y un no desastre.

* **Eje Horizontal (Clases Predichas):** Las clases predichas por el modelo están dispuestas horizontalmente.
* **Eje Vertical (Clases Verdaderas):** Las clases verdaderas o reales están dispuestas verticalmente.



*Ilustración 10: Matriz de confusión de CNN*

Teniendo en cuenta en la imagen, se establece que el modelo parece ser muy eficiente en clasificar instancias como "No Desastre", ya que no tiene falsos positivos (0)y un gran número de verdaderos negativos (1091). Sin embargo, el modelo no ha logrado identificar correctamente ninguna instancia de la clase "Desastre", ya que los falsos negativos (813) son elevados y no hay verdaderos positivos (0) [35].

* 1. ***Curva ROC***

Esta representación gráfica permite la comparación directa del rendimiento entre varios modelos, observando la forma de sus curvas ROC [36]. Un posicionamiento cercano al vértice superior izquierdo indica un mejor rendimiento del modelo en términos de sensibilidad y especificidad [36] [37]. La métrica numérica Área Bajo la Curva (AUC) complementa esta evaluación, proporcionando una medida cuantitativa del desempeño general del modelo [38].

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 11:* *Curva de ROC del modelo CNN*

Como resultado la ilustración, se obtiene que el modelo tiene un buen rendimiento en la identificación de instancias positivas [39], minimizando los falsos negativos. Esto sugiere una alta sensibilidad del modelo, pero también es crucial considerar el equilibrio con la especificidad. Es decir, que al igual se puede visualizar la precisión del modelo.

1. ***Modelo RNN***
   1. ***Gráfica de perdidas***
      * **Eje x:** Los Epoch o épocas se representan numéricamente, con una secuencia que comienza en 0.0 y se extiende hasta 5.0 Los intervalos en el eje x están marcados en incrementos de 1.
      * **Eje y:** La trayectoria de la pérdida inicia en un valor aproximado de 0.15 y exhibe una tendencia decreciente conforme crece el número de epoch.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 12: Gráfica de pérdidas del modelo CNN*

En la primera época, la pérdida en el conjunto de entrenamiento es de 0.8326, disminuyendo a lo largo de las épocas y alcanzando 0.4719 en la sexta época. En el conjunto de validación, la pérdida disminuye significativamente de 0.64574 en la primera época a 0.43466 en la quinta época. La mejora en la pérdida indica que el modelo está ajustando sus pesos para minimizar la diferencia entre las predicciones y los valores reales.

* 1. ***Gráfica de la precisión***
     + **Eje x:** Se muestran diferentes niveles de precisión, desde 0 (indicando una precisión perfecta) hasta .0. Estos valores representan los resultados de diferentes iteraciones del modelo.
     + Gráfico, Gráfico de líneas

       Descripción generada automáticamente**Eje y:** Los números que van desde menos de 0.6 hasta 0.80 en incrementos de 0.05 representan el número de épocas relacionados con el entrenamiento del modelo.

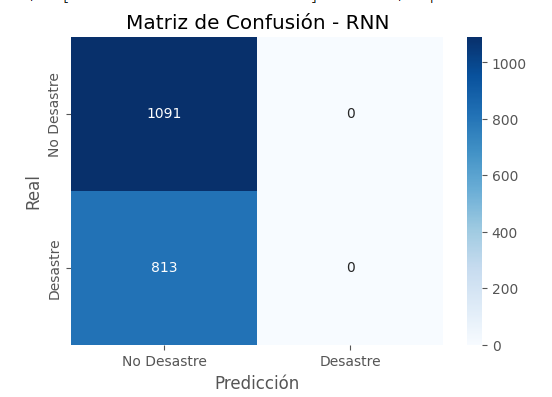
1. *Ilustración 13: Gráfica de precisión del modelo RNN*

La precisión en el conjunto de entrenamiento aumenta de 56.38% en la primera época a 79.37% en la sexta época. En el conjunto de validación, la precisión mejora constantemente, alcanzando un máximo de 82.09% en la quinta época.

* 1. ***Matriz de Confusión***

La Ilustración presenta una matriz de confusión, correspondiente a un modelo de red convolucional diseñado para clasificar los tweets acerca si habla de un desastre y un no desastre.

* **Eje Horizontal (Clases Predichas):** Las clases predichas por el modelo están dispuestas horizontalmente.
* **Eje Vertical (Clases Verdaderas):** Las clases verdaderas o reales están dispuestas verticalmente.

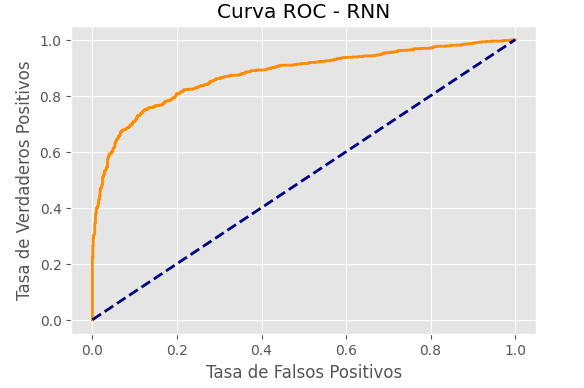


*Ilustración 14: Curva de ROC del modelo RNN*

Teniendo en cuenta en la imagen, se establece que el modelo parece ser muy eficiente en clasificar instancias como "No Desastre", ya que no tiene falsos positivos (0) y un gran número de verdaderos negativos (1091). Sin embargo, el modelo no ha logrado identificar correctamente ninguna instancia de la clase "Desastre", ya que los falsos negativos (813) son elevados y no hay verdaderos positivos (0).

* 1. ***Curva ROC***

Esta representación gráfica es la siguiente:



*Ilustración 15: Curva de ROC del modelo RNN*

Como resultado la ilustración, se obtiene que el modelo tiene un buen rendimiento en la identificación de instancias positivas minimizando los falsos negativos. Esto sugiere una alta sensibilidad del modelo, pero también es crucial considerar el equilibrio con la especificidad. Es decir, que al igual se puede visualizar la precisión del modelo.

1. ***Modelo BiGRU***
   1. ***Gráfica de perdidas***
      * **Eje x:** Los Epoch o épocas se representan numéricamente, con una secuencia que comienza en 0.0 y se extiende hasta 4.0 Los intervalos en el eje x están marcados en incrementos de 0.25.
      * **Eje y:** La trayectoria de la pérdida inicia en un valor aproximado de 0.4 y exhibe una tendencia decreciente conforme crece el número de epoch.



*Ilustración 16:*

*Grafica del perdido de BiGRU*

La evolución de la pérdida a lo largo de las cinco épocas del modelo BiGRU revela una disminución constante en el conjunto de entrenamiento, señalando una adaptación efectiva a los datos de entrenamiento y una mejora continua en la capacidad predictiva [23]. En el conjunto de validación, se observa una tendencia descendente en las primeras cuatro épocas, indicando una buena generalización a datos no vistos. Sin embargo, en la quinta época, la pérdida en el conjunto de validación experimenta un leve aumento, sugiriendo la posibilidad de un ligero sobreajuste que podría abordarse mediante técnicas de regularización, como la optimización de la tasa de dropout o la reducción de la complejidad del modelo.

* 1. ***Gráfica de la precisión***
     + **Eje x:** Se muestran diferentes niveles de precisión, desde 0 (indicando una precisión perfecta) hasta 4.0. Estos valores representan los resultados de diferentes iteraciones del modelo.
     + **Eje y:** Los números que van desde 0.65 hasta 0.85 en incrementos de 0.05 representan el número de épocas relacionados con el entrenamiento del modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

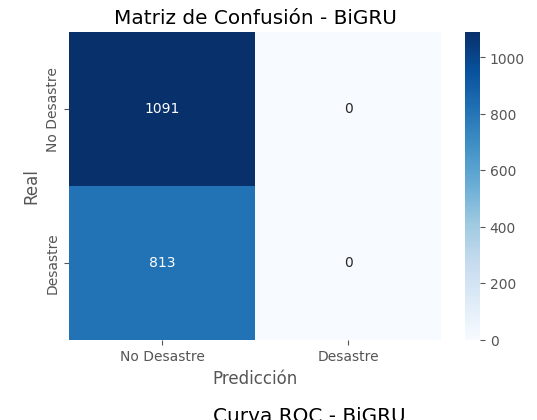
*Ilustración 17: Grafica del perdida de BiGRU*

La métrica de precisión (accuracy) en el conjunto de entrenamiento muestra que el modelo está aprendiendo bien los datos de entrenamiento, alcanzando una precisión del 90.54% en la tercera época. La precisión en el conjunto de validación también es razonablemente alta (81.20%), pero la ligera subida en la pérdida podría requerir una revisión más detallada al momento del análisis del modelo.

* 1. ***Matriz de Confusión***

La Ilustración 8 es similar a las matrices de confusión anteriormente evaluado de las anteriores modelos y el análisis es el mismo.

* **Eje Horizontal (Clases Predichas):** Las clases predichas por el modelo están dispuestas horizontalmente.
* **Eje Vertical (Clases Verdaderas):** Las clases verdaderas o reales están dispuestas verticalmente.



*Ilustración 18: Matriz de confusión del modelo BiGRU*

* 1. ***Curva ROC***

Esta representación gráfica permite la comparación directa del rendimiento entre varios modelos, observando la forma de sus curvas ROC. Un posicionamiento cercano al vértice superior izquierdo indica un mejor rendimiento del modelo en términos de sensibilidad y especificidad.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 19:* *Cuerva de BiGRU*

Como resultado la ilustración, se obtiene que el modelo tiene un buen rendimiento en la identificación de instancias positivas, minimizando los falsos negativos. Esto sugiere una alta sensibilidad del modelo, pero también es crucial considerar el equilibrio con la especificidad. Es decir, que al igual se puede visualizar la precisión del modelo.

1. ***Modelo*** ***BiTLM***
   1. ***Gráfica de perdidas***
      * **Eje x:** Los Epoch o épocas se representan numéricamente, con una secuencia que comienza en 0.0 y se extiende hasta 4 .0 Los intervalos en el eje x están marcados en incrementos de 0.25.
      * **Eje y:** La trayectoria de la pérdida inicia en un valor aproximado de 0.35 y exhibe una tendencia decreciente conforme crece el número de epoch.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 20: Grafica del perdido de BiTLM*

La evolución de la pérdida en el conjunto de entrenamiento muestra una disminución constante a lo largo de las cinco épocas, indicando una eficiente convergencia del modelo. Comienza con una pérdida inicial de 0.7475 en la primera época y alcanza su punto más bajo en la última época, situándose en 0.3819. Esta tendencia descendente sugiere que el modelo está aprendiendo de manera efectiva las representaciones de los datos de entrenamiento. Sin embargo, es importante destacar que la pérdida en el conjunto de validación, aunque inicialmente decreciente, experimenta un ligero aumento en la última época, señalando la posibilidad de sobreajuste.

1. ***Gráfica de la precisión***
   * + **Eje x:** Se muestran diferentes niveles de precisión, desde 0 hasta .0. Estos valores representan los resultados de diferentes iteraciones del modelo.
     + **Eje y:** Los números que van desde 0.675 hasta 0.850 en incrementos de 0.025 representan el número de épocas relacionados con el entrenamiento del modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 21 Grafica del precisión de BiGRU*

En cuanto a la precisión, se observa un aumento constante tanto en el conjunto de entrenamiento como en el de validación durante el proceso de entrenamiento. La precisión en entrenamiento inicia en 63.27% en la primera época y experimenta un crecimiento constante hasta alcanzar un máximo de 83.99% en la quinta época. Similarmente, la precisión en el conjunto de validación presenta una tendencia ascendente, iniciando en 80.04% y alcanzando su punto más alto en 83.46% en la cuarta época. Aunque la precisión es alta, la pequeña discrepancia entre las precisiones de entrenamiento y validación en las últimas épocas sugiere la necesidad de abordar el sobreajuste para mejorar la generalización del modelo en datos no vistos.

* 1. ***Matriz de Confusión***

La Ilustración 21 es similar a las matrices de confusión de los dos primeros [modelos](#_RESULTADOS_Y_DISCUSIÓN).

* **Eje Horizontal (Clases Predichas):** Las clases predichas por el modelo están dispuestas horizontalmente.
* **Eje Vertical (Clases Verdaderas):** Las clases verdaderas o reales están dispuestas verticalmente.

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

*Ilustración 21: Matriz de confusión del modelo BiLSTM*

* 1. ***Curva ROC***

En la representación gráfica permite la comparación directa del rendimiento entre varios modelos, observando la forma de sus curvas ROC. Un posicionamiento cercano al vértice superior izquierdo indica un mejor rendimiento del modelo en términos de sensibilidad y especificidad [30].

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Ilustración 22: Curva de ROC de BiGRU*

Como resultado la ilustración, se obtiene que el modelo tiene un buen rendimiento en la identificación de instancias positivas, minimizando los falsos negativos. Esto sugiere una alta sensibilidad del modelo, pero también es crucial considerar el equilibrio con la especificidad. Es decir, que al igual se puede visualizar la precisión del modelo.

# **CONCLUSIONES**

En conclusión, la exploración y evaluación de varios modelos de procesamiento del lenguaje natural (NLP) para la clasificación de eventos desastrosos revela un rendimiento prometedor en la identificación de patrones y características asociadas con la presencia de desastres.

Tras el estudio cada modelo, se destaca el mejor modelo depende de consideraciones específicas y métricas de evaluación prioritarias. Los resultados de entrenamiento y evaluación de los modelos, tanto el BiGRU (Sequential\_40) como el CNN (Sequential\_37) se destacan con precisiones superiores al 80% en el conjunto de validación, indicando un rendimiento competitivo en la clasificación de desastres [12].

Ambos modelos enfrentan desafíos de sobreajuste, resaltando la necesidad de técnicas de regularización. La matriz de confusión y métricas específicas revelan que la identificación de la clase "No Desastre" es más robusta que la de "Desastre", señalando áreas de mejora en la detección de eventos catastróficos. En términos de eficiencia, el modelo CNN destaca por su menor tamaño y tiempo de entrenamiento. Sin embargo, la elección final entre ambos modelos dependerá de los objetivos particulares, prioridades de métricas y consideraciones de eficiencia computacional, proporcionando así un enfoque personalizado para abordar las necesidades específicas de la tarea de clasificación de desastres.

# **REFERENCIAS**

|  |  |
| --- | --- |
| [1] | H. Herrera, «LAS REDES SOCIALES: UNA NUEVA HERRAMIENTA DE DIFUSIÓN SOCIAL NETWORKS: A NEW DIFFUSION TOOL,» Universidad de Costa Rica, San José, Costa Rica, 2012. |
| [2] | E. Çano y M. Morisio, *Word Embeddings for Sentiment Analysis: A Comprehensive Empirical Survey,* Computation and Language (cs.CL), 2019. |
| [3] | M. Azizjon, A. Jumabek y K. Wooseong, «1D CNN based network intrusion detection with normalization on imbalanced data,» de *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, Fukuoka, 218-224, p. doi: 10.1109/ICAIIC48513.2020.9064976.. |
| [4] | Y. JJingxiu y M. Shepperd, *The impact of using biased performance metrics on software defect prediction research,* 2021: Information and Software Technology, doi: https://doi.org/10.1016/j.infsof.2021.106664. |
| [5] | H. Mazhar, I. Khurum, G. Rozaida y S. Zunaira, «Karachi Stock Exchange Price Prediction using Machine,» *EAI Endorsed Transactions on Creative Technologies,* vol. 4, nº 12, pp. 17-23, 2021. |
| [6] | Khan, Ghazanfar, Azam, Karami, Alyoubi y Alfakeeh, *Stock market prediction using machine learning classifiers and social media, news,* 10.1007/s12652-020-01839-w: J. Ambient Intell. Humaniz. Comput., 2020. |
| [7] | M. Vargas, C. Dos Anjos, G. Bichara y A. Evsukoff, *Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles,* DOI:10.1109/IJCNN.2018.8489208: Proc. Int. Jt. Conf. Neural Networks, 2018. |
| [8] | C. Eunsuk, H. Chulwoo y F. Park, *Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representations, and Case Studies,* DOI:10.1016/j.eswa.2017.04.030: Expert Systems with Applications, 2017. |
| [9] | M. Luciana, R. Norton y M. Ana, «Forecasting stock market index daily direction: A Bayesian Network approach,» *Expert Systems with Applications,* vol. 105, nº https://doi.org/10.1016/j.eswa.2018.03.039, pp. 11-22, 2018. |
| [10] | M. Adil y H. Mhamed, *Stock Market Prediction Using LSTM Recurrent Neural Network,* doi: 10.1016/j.procs.2020.03.049.: Creative Commons, 2020. |
| [11] | Kaggle, «Kaggle : Your Home for Data Science,» Website, 2021. [En línea]. |
| [12] | Google, «Colaboratory,» [En línea]. Available: ¿Qué es Colaboratory?. |
| [13] | V. Mirjalili y S. Raschka, Python Machine Learning, Marcombo, 2020. |
| [14] | N. Rusk, «Deep learning,» *NATURE METHODS,* vol. 13, nº 1, p. 35, 2016. |
| [15] | M. Taboada, Sentiment Analysis: An Overview from Linguistics, doi 10.1146/annurev-linguistics-011415-040518: British Columbia V5A 1S6, 2016. |
| [16] | L. Li, T. Goh y D. Jin, How textual quality of online reviews affect classification performance: A case of deep learning sentiment analysis, Neural Comput, 2018, p. 1. |
| [17] | K. O’Shea y R. Nash, «An Introduction to Convolutional Neural Networks,» Neural and Evolutionary Computing , doi https://arxiv.org/abs/1511.08458, 2015. |
| [18] | H. Sutskever, J. Martens y G. Hinton, «Generating Text with Recurrent Neural Networks,» University of Toronto, Toronto, 2011. |
| [19] | A. Aipe, E. Asif, K. Sadao y Mukuntha, *Deep Learning Approach towards Multi-label Classification of Crisis Related Tweets,* WiPe Paper, 2018. |
| [20] | D. S. Team, «Data Science,» 24 abril 2020. [En línea]. [Último acceso: 16 01 2024]. |
| [21] | S. Cavall y M. Amoretti, «CNN-based multivariate data analysis for bitcoin trend prediction,» *Applied Soft Computing,* vol. 101, p. https://doi.org/10.1016/j.asoc.2020.107065, 2021. |
| [22] | K. Cho, D. Bahdanau y Y. Bengio, «Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,» Doha, Qatar, Association for Computational Linguistics, 2014, pp. 1724-1734. |
| [23] | D. She y J. Minping, «A BiGRU method for remaining useful life prediction of machinery,» *Elsevier,* vol. 167, p. doi: https://doi.org/10.1016/j.measurement.2020.108277, 2021. |
| [24] | X. Guixian, M. Yueting y Y. Ziheng, «Sentiment Analysis of Comment Texts Based on BiLSTM,» *IEEE,* vol. 7, 2006. |
| [25] | L. J. Vázquez Díaz, «Generacion y comparativa de modelos anal ´ ´ıticos AutoML con el uso de datos no tabulados mediante NLP contra modelos de Data Science,» pp. 1-13, 2021. |
| [26] | A. Zapata García, «Análisis de textos mediante técnicas NLP para la categorización de usuarios,» 2021. |
| [27] | F. Castro Caicedo y P. Tojoa Gómez, Máquina de entrenamiento comprimido basada en Extreme Learning Machine −MEC-ELM, . Universidad del Cauca., Ed., Colombia: Universidad del Cauca., 2022, pp. 15-130. |
| [28] | J. M. Cotelo, «Explorando Twitter mediante la integraci´on de informaci´on estructurada y no estructurada,» pp. 2-9, 2 2015. |
| [29] | J. Manuel y S. Muñoz, «Investigación Análisis de Calidad Cartográfica mediante el estudio de la Matriz de Confusión Quality Cartographic analysis by studying Confusion Matrix,» 2016. |
| [30] | N. Christos T, B. Leonidas E y G. Constantine A, «ROC Analysis for Classification and Prediction in Practice,» *CRC Press,* pp. 01-234, 3 2023. |
| [31] | V. Raina y S. Krishnamurthy, «Natural Language Processing,» *Building an Effective Data Science Practice,* pp. 63-73, 2022. |
| [32] | R. Montiel González, M. A. Bolaños González, A. Macedo Cruz, A. Rodríguez González y A. López ´Pérez, «Clasificación de uso del suelo y vegetación con redes neuronales convolucionales,» *Revista Mexicana de Ciencias Forestales,* vol. 13, nº 74, pp. 97-119, 11 2022. |
| [33] | K. FONDEVILA R, «USO DE REDES NEURONALES CONVOLUCIONALES EN PYTHON PARA LECTURA DE PRUEBAS RÁPIDAS DE COVID19,» 2022. |
| [34] | E. Alférez B, «Servicio Web para la clasificación de cáncer de piel usando redes neuronales profundas,» pp. 1-40, 2018. |
| [35] | W. Atencia, J. Bustillo y J. Rambal, «Analizador de tweets asociados a la política y polarización Colombiana,» pp. 1-21, 11 2020. |
| [36] | H. Ayala-Yaguara, G. Valenzuela Sabogal y A. Espinosa García, «Obtención de un modelo de minería de datos aplicado a la deserción universitaria del programa de Ingeniería de Sistemas de la Universidad de Cundinamarca,» pp. 134-150, 2019. |
| [37] | F. Izco, «Base de datos corporativa de personas,» 11 2018. |
| [38] | C. L. Inca, A. M. Paredes y Á. P. [. Mena, «Eficiencia de modelos de predicción de COVID-19 usando curvas ROC y matriz de confusión,» vol. 8, pp. 1-1460, 4 2022. |
| [39] | T. Weerasooriya, N. Perera y S. Liyanage, «A method to extract essential keywords from a tweet using NLP tools,» *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer),* pp. 29-34, 9 2016. |
| [40] | G. Rojas Espinoza y O. Ortiz Iribarren, «Identificación del Cilindro Nudoso en imágenes TC de trozas podadas de Pinus Radiata utilizando el Clasificador de Máxima Verosimilitud,» *Maderas. Ciencia y tecnología,* vol. 11, nº 2, pp. 1-26, 2009. |
| [41] | J. N. Mandrekar, «Receiver Operating Characteristic Curve in Diagnostic Test Assessment,» *Journal of Thoracic Oncology,* vol. 5, nº 9, pp. 1315-1316, 9 2010. |
| [42] | J. Cerda y L. Cifuentes, «Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos,» *Revista chilena de infectología,* vol. 29, nº 2, pp. 138-141, 4 2012. |