

Integración de Redes Neuronales de Grafos y Análisis de Redes Complejas para el Pronóstico de Eventos Sísmicos en Colombia

Integration of GNN with Complex Network Analysis for Seismic
Event Forecasting in Colombia

Estudiante:

Santiago Andrés Giraldo

Código 2004890

giraldo.santiago@correounivalle.edu.co

Director:

Victor Andrés Bucheli Guerrero, Ph.D.

victor.bucheli@correounivalle.edu.co

Escuela de Ingeniería de Sistemas y Computación

Universidad del Valle

Cali, Valle del Cauca

2023

Tesis presentada por
Santiago Andrés Giraldo Alegría
Como requisito parcial para la obtención del título de Magíster en Ingeniería

Victor Andres Bucheli Guerrero Ph.D

Jurado

Jurado

Tabla de Contenidos

1. Resumen	4
2. Introducción	4
2.1. Planteamiento del problema	4
2.2. Justificación	5
2.3. Pregunta de Investigación	6
2.4. Objetivo General	6
2.5. Objetivos Específicos	6
3. Marco Referencial	7
3.1. Estado del arte	7
3.2. Redes neuronales de grafos	14
3.3. Modelo estándar de minería CRISP-DM	15
4. Metodología	16
4.1. Entendimiento de los datos	17
4.1.1. Distribución de Magnitud (Arriba a la izquierda)	19
4.1.2. Distribución de Profundidad (Arriba a la derecha)	20
4.1.3. Distribución de Latitud (Abajo a la izquierda)	20
4.1.4. Distribución de Longitud (Abajo a la derecha)	20
4.2. Conjuntos de datos	20
4.2.1. Conjuntos de datos para redes neuronales LSTM	20
4.2.2. Conjuntos de datos para redes de grafos	22
4.3. Desarrollo y Entrenamiento de Redes Neuronales	22
4.3.1. Redes Neuronales LSTM	23
4.3.2. Redes Neuronales de Grafos	23
4.4. Resultados y Evaluación	24
4.4.1. Arquitectura y Diagrama de Clases	24
4.4.2. Experimentos con Redes Neuronales LSTM	27
4.4.3. Experimentos con Redes Neuronales LSTM con Características de Redes	29
4.4.4. Experimentos con Redes Neuronales de Grafos	30
4.4.4.1. Distancia 30km	30
4.4.4.2. Distancia 50 km	33
4.4.4.3. Distancia 100 km	34
5. Conclusiones	37
6. Referencias	37

1. Resumen

Los terremotos se consideran una catástrofe, no dan tiempo suficiente para reaccionar y tomar precauciones, por lo que generan víctimas y daños en la infraestructura. En los últimos años ha habido un creciente interés investigativo en el pronóstico de terremotos aplicando aprendizaje automático y redes neuronales que han obtenido resultados asombrosos. Este trabajo aborda el problema de pronóstico de terremotos con una aplicación de Redes Neuronales de Memoria a Corto y Largo Plazo (LSTM por sus siglas en inglés) y Redes Neuronales de Grafos (GNN por sus siglas en inglés) a eventos sísmicos históricos de Colombia, obtenidos mediante el Servicio Geológico de los Estados Unidos (USGS por sus siglas en inglés), integrando propiedades de redes complejas tales como centralidad del grado (degree centrality), centralidad de vector propio (eigenvector centrality) y PageRank.

Se analizó el catálogo de eventos sísmicos y se construyeron redes complejas seleccionando una ventana de eventos, teniendo como objetivo encontrar la característica siguiente en el tiempo. Además entrenamos múltiples redes neuronales LSTM y GNN con ajuste de hiper parámetros mediante la librería Ray Tune encontrando una diferencia bastante marcada en los resultados, obteniendo mejores métricas de clasificación con el modelo GNN

1.1. Palabras Clave

Pronóstico de Sismos, Redes Neuronales LSTM, Redes Neuronales de Grafos, Aprendizaje de Máquina, Análisis de Redes Complejas, Ajuste de Hyper Parámetros

2. Introducción

2.1. Planteamiento del problema

Los terremotos ocurren sin previo aviso y no hay suficiente tiempo para tomar medidas preventivas, generando la pérdida de vidas humanas e infraestructura. En los últimos años, de acuerdo a diversos trabajos que se recopilan en el estado del arte [1], [2], [3], [4], [5], [6] se documenta el creciente interés en el uso del aprendizaje de máquina e inteligencia artificial para el pronóstico de estos eventos sísmicos, señalando trabajo a futuro y diversos retos a abordar.

Uno de los enfoques principales corresponde a la aplicación de redes neuronales profundas para encontrar patrones complejos, aunque se señala la dificultad para el pronóstico de características tales como magnitud (para

eventos con magnitud mayor a 5 debido a su escasez) y secuencia de eventos proyectados a largo plazo, se observan buenos resultados utilizando como entrada a la red neuronal la secuencia de diferentes características. Entre estas podemos observar: magnitud, frecuencia u ocurrencia de eventos [7], [8], [9], [10], [11], ocurrencia de réplicas [12], probabilidad de detección y tasa de falsas alarmas [13], variación del campo geomagnético, temperatura del suelo y humedad relativa [14], indicadores sísmicos [15], ubicación [16], Valor b de la ley de Gutenberg-Richter [17], [18], [19], gravedad, georresistividad y nivel del agua [20], valor b, desviación cuadrática media, déficit de magnitud, días transcurridos, magnitud media, tasa de la raíz cuadrada de la energía liberada, tiempo medio entre eventos característicos, coeficiente de variación del tiempo medio [21] y señales electrónicas capturadas en sismógrafos [22], [23], [24], [25], [26], [27].

Además, de acuerdo con Florido et al [2], ante una variedad de métodos de aprendizaje de máquina tales como: Máquinas de Soporte Vectorial, Árboles M5P, Naive Bayes, K-Near Neighbors, Clasificador J48, Random Forest, LPBoost Ensemble, entre otros; se destacan los resultados de las Redes Neuronales Artificiales para el pronóstico de diferentes características (ej. ocurrencia, ubicación, magnitud, etc) con implementaciones en Chile, Japón, India, China, Pakistán, USA o la Península Ibérica, Grecia o Portugal [1], [2].

A pesar de esto no se encontraron trabajos relacionados con aprendizaje de máquina o minería de datos para el pronóstico de sismos en Colombia, por lo tanto su aplicación es una propuesta innovadora.

2.2. Justificación

Las técnicas de inteligencia artificial y aprendizaje de máquina son herramientas poderosas para encontrar patrones ocultos más allá de la intuición humana [6], su aplicación en el pronóstico de sismos es útil para la evaluación de riesgos y prevención temprana [6], [22].

Jiao y Alavi et al [4] expone que los estudios en sismología se realizan por los siguientes objetivos: acciones para reducir a largo plazo desastres generados por terremotos, preparación y ajuste ante desastres, estrategias de respuesta a desastres y planificación de recuperación después del desastre. Aunque limitados o afectados por la cantidad de datos y ocurrencia, la inteligencia artificial provee herramientas para obtener buenos pronósticos y tomar buenas decisiones.

De acuerdo con Satriano et al [28], sistemas de información enfocados a la alerta temprana de eventos sismológicos deben proveer de manera rápida y confiable información sobre distintos parámetros tales como ubicación, tiempo y tamaño para así mejorar la respuesta de emergencia.

Espinosa et al [29] menciona que el sistema de alerta temprana SAS implementado en México ha permitido detectar ondas sísmicas con hasta 60 segundos de anticipación, tiempo que permite ejecutar procedimientos de seguridad para proteger equipos susceptibles a daños tales como plantas de energía, sistemas informáticos y redes de comunicación.

Debido a que en los trabajos explorados aún no hay aplicación de redes neuronales y aprendizaje de máquina en Colombia para el pronóstico de sismos, se destaca su importancia desde el ámbito de la investigación y prevención de riesgos.

Además, de acuerdo a León et al [30], integrar el análisis de redes complejas para estructurar la relación entre eventos sísmicos puede ayudar en la mejora de resultados debido a las interacciones encontradas, por lo que incluir características tales como por ejemplo la centralidad de un nodo dentro una red compleja, puede ayudar a dar importancia a características de eventos sísmológicos y mejorar el pronóstico de uno o más eventos futuros.

2.3. Pregunta de Investigación

De acuerdo a lo expuesto anteriormente en este documento se plantea la siguiente pregunta de investigación: ¿Cómo integrar el análisis de redes complejas al pronóstico de eventos sísmicos mediante el aprendizaje de máquina con redes neuronales profundas?

2.4. Objetivo General

Desarrollar un modelo de redes neuronales profundas que integre el análisis de redes complejas y redes neuronales de grafos para el pronóstico de eventos sísmicos de acuerdo a una serie espacio-temporal en Colombia.

2.5. Objetivos Específicos

Implementar un flujo de trabajo de aprendizaje de máquina inspirado en el estado del arte y el modelo estándar para minería de datos CRISP-DM propuesto por Wirth et al [31]

- 2.5.1. Preparar los datos de eventos sísmicos en Colombia obtenidos desde el Servicio Geológico de Estados Unidos y otras fuentes.
- 2.5.2. Generar los conjuntos de datos de entrenamiento y prueba construyendo las redes complejas de sismos de acuerdo a lo expuesto en León et al [30] incluyendo características de redes complejas: clustering coefficient, betweenness centrality, closeness centrality, pagerank centrality, eigen centrality y degree centrality.

- 2.5.3. Entrenar de forma sistemática modelos de redes neuronales profundas con capas LSTM y redes neuronales de grafos para predecir enlaces, con ajuste de hiperparametros.
- 2.5.4. Evaluar mediante diferentes métricas (Accuracy, Recall, F1 Score) el desempeño de los modelos de redes neuronales entrenados sistemáticamente.

3. Marco Referencial

Las investigaciones en pronóstico de sismos empezaron finalizando el siglo 19, en la revisión de Geller [32] se dividieron las investigaciones en diferentes etapas: antes de 1960, después de 1960, y de 1962 hasta 1997, este trabajo sirvió como punto de partida para para años posteriores y la discusión se centró en factores que permitieran predecir un sismo sin llegar a implementar y/o comparar modelos que intentaran hacerlo.

En la revisión de literatura y el planteamiento del problema se destaca el uso de diferentes enfoques y estructuras de redes neuronales artificiales (ANN por sus siglas en inglés) para clasificación y regresión, tales como la caracterización de señales sísmicas para estimar la ubicación y magnitud de eventos, pronóstico de eventos futuros dejando la relación entre los datos a la interpretación de modelos de aprendizaje de máquina, la construcción de modelos para representar factores que contribuyen a la evolución del sistema en el tiempo, entre otros.

3.1. Estado del arte

A continuación se realizará una revisión rápida al estado del arte desarrollado por Banna et al. [1]

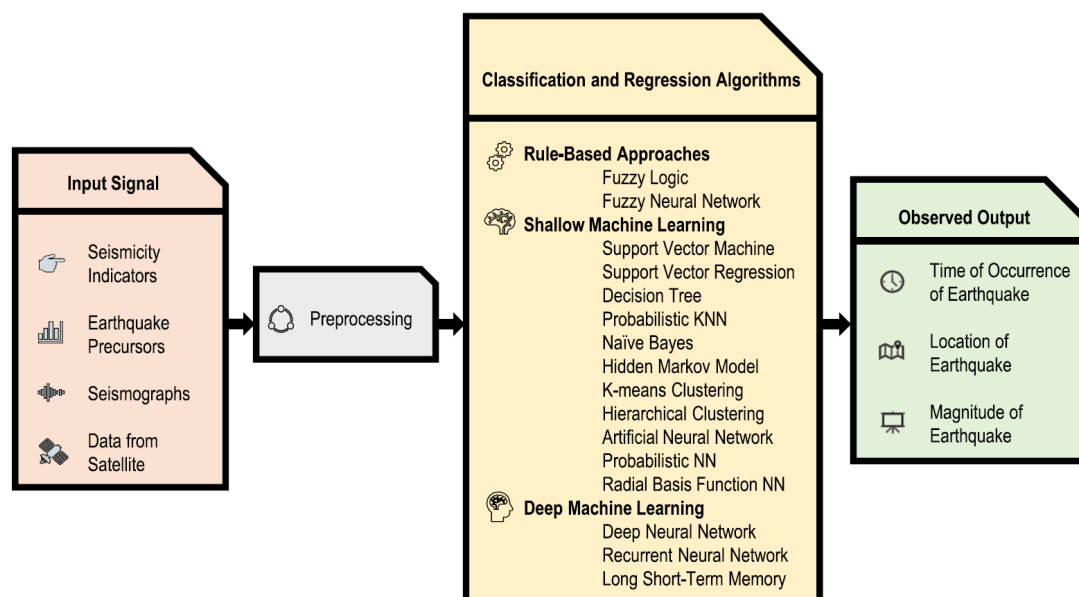


Figura 1. Aproximación general del problema de pronóstico de sismos

De acuerdo a lo discutido en este estado del arte muchas implementaciones usan diferentes tipos de datos de entrada, tales como: señales eléctricas de sismos; tiempo, magnitud y ubicación; sensores y ondas sísmicas; precursoros, entre otros, para encontrar tendencias y así predecir y caracterizar terremotos, o réplicas (ver **Tabla 1**). Aunque se mencionan los precursoros como dato de entrada, ninguno ha sido prueba definitiva para el pronóstico de sismos.

Used Algorithm	Used Features
ANN	dpt., esn., mg., Tile location
	Increments of b-value, Bath's law, OU's law
	Seven seismicity indicators
	OU's law, Dynamic GR's law, seismicity indicators
	Daily maximum mg., SES
	et., oid., heid., hid., it., ed.
	$t_e, \bar{A}_e, E_e, \frac{dA_e}{dt}(dB), \Delta_e, \bar{t}_e, \delta A_e, \rho$
	mg., dbeecf., timing, focal dpt.
	15 channels from ocean bottom pressure gauge
	Seismicity variation rates
Embedded BPNN	Six seismic focus parameters from MapSIS
	Occurrence time, slip rate, epicenter, dpt. b-value, cumulative frequency, energy, etc.
LM with BP	dpt., tos., lo., la., pressure, rc.
	et., oid., heid., hid., it., ed.
LR and BP	Seismic and time varying parameters
NLFA and ANN	b-value, scde., noe., risk, sei., if.
HC and ANN	srga., wbt., La., Lo., dbt., stp.
ANN and SVM	lo., la., dpt.
BP and AdaBoost	Statistical features, Wavelet features
NARX	vgfd., hrh., gt., horizontal component, rainy rate
NDC and NDAP	$t_e, \bar{A}_e, E_e, \frac{dA_e}{dt}(dB), \Delta_e, \bar{t}_e, \delta A_e, \rho$
GMDHNN	Time Series data
MFOFLANN	Six seismicity indicators
GA	Geo-magnetic and solar indices
ANN with GA	b-value, accumulation frequency of eq., energy, etc.

Used Algorithm	Features
DNN	long dipoles L's - I, L' and L, SES First 2.56s (28 samples) of P-waves
LSTM	Spatio temporal parameters Seismicity indicators
RNN, PRNN, RF, LP Boost	$t_e, \bar{A}_e, E_e, \frac{dA_e}{dt} (dB), \Delta_e, \bar{t}_e, \delta A_e, \rho$
RNN	
RNN, BP, RBF	
DNN, GLM, RF, GBM	Seismicity indicators, increments of b-value

Tabla 1. Listado de aproximaciones en aprendizaje de máquina con sus respectivos valores de entrada para el pronóstico de sismos, *referirse a la tabla 4 en [1] para abreviaciones*

La evaluación de estas diferentes aproximaciones se ve documentada con diferentes métricas tales como: valores positivos y negativos de predicción (P1,P0), especificidad (Sp), sensibilidad (Sn), exactitud, tasa de falsa alarma (FAR), R-score, raíz del error cuadrático medio (RMSE), error cuadrático medio (MSE), error relativo (RE), error absoluto medio (MAE), área bajo la curva (AUC), etc. En la **Tabla 2** se puede observar los respectivos resultados y desempeño para implementaciones diversas de ANN, DNN, RNN, etc.

Used Algorithm	Dataset	Evaluation Metrics	Performance
ANN	NCEDC	MAE, MSE	MSE was 0.115351
	CNSS	P_0, P_1, Sn, Sp	Sp was 87.2% and Sn was 46.9%
	Earthquake catalog of SGI		Sn was 87.2% (Alboran sea)
	SES dataset of VAN group		Sn was 80% and Sp was 91.30%
	DEMETER satellite data	MAE	Acc. was 83.56% (Magnitude prediction)
	USGS	Acc.	Acc. was 81.2%
		AVE, maximum error	Acc. was 80% (mid-level eq.)
	Data from JAMSTEC		Average error was 0.5% to 3.5% (eq. energy)
	CENC	Error curve	Successfully predicted aftershocks
		RMSE	Predicted time within 2 months of error
Embedded BPNN		PCC, RMSE, MAE	Reduced the RMSE to 0.41ML
		Error	MSE was between 0.01 and 0.09
LM and BP	EAFS	RE	With LM-BPNN error was 5.6
	DEMETER satellite	Acc.	The highest relative error was 12%
LR and BP	NESDCCSN	RMSE	Acc. was 83.90% (magnitude > 5.9)
NLFA and ANN	NOAA and USGS	R-value	RMSE was ± 0.28 ML
HC and ANN	TSMS	RE	The R-value was 0.525
ANN and SVM	CENC	Acc.	The highest prediction error was 6.25%
BP and AdaBoost	CESDC	Acc., Precision, Recall	Acc. was 73.37%
NARX	TGRC	Norm of error	Acc. was 99.49%
NDC and NDAP	SCEC	Acc., POD, FAR	The highest norm of error was 0.067
GMDHNN	USGS	MAE, RMSE, NMAE	Acc. was 99.4%
MFOFLANN	USGS and SCEC	RMSE	MAE was 0.1734
GA	GIM data by NASA JPL		RMSE was 0.0041
ANN with GA		AE, RE	DTEC value exceeded 54.66% (before 8 days)
BP and BP-GA			Averaged relative errors decreased to 0.4698
BP and PSO		AE	14 eq. will happen till 2044 ($M > 6.5$)
IABC with MLP	SCEC	Acc., MSE, NMSE	The AE was 0.021 to 0.375
Meta-Learning and TL		MSE	Acc. was 99.89%
Ensemble of RBFNN		Acc.	MSE 0.07
		AVE, maximum error	Acc. was 91% (Magnitude)
			Average prediction error was 0.2722
RBFNN	GSS	MAE, RMSE	RMSE was 0.04217
	USGS	RMSE, WIA	RMSE was 0.80
	USGS and NEIC	Acc., P_0 , POD, FAR	Acc. was 96%

Used Algorithm	Dataset	Evaluation Metrics	Performance
DNN	SES of 18th, 19th April 1995	Acc., ROC curve	Acc. was 83% (Anomaly detection)
	KiK-net	Acc.	Acc. was 84.21% (Magnitude prediction)
LSTM	USGS		Acc. was 88.57% with 2D input
RNN, PRNN, RF, LP Boost	CES and USGS	R^2	Achieved R^2 value of -0.252
RNN		Acc., Sn, Sp, P_0, P_1	Acc. was 79% (PRNN and LPBoost)
RNN, BP, RBF	SCEC	POD, FAR, R-score	Predicted location with 15-39 miles error
DNN, GLM, RF, GBM	NCEDC	MAE, RE	POD score was 0.658 (RNN)
			MAE was 0.74 (RF)

Tabla 2. Listado de resultados para los métodos aplicados en diferentes trabajos de investigación con conjuntos de datos variados

El aprendizaje profundo es un tema importante, este método, en general, no necesita ingeniería de características ya que puede generar miles de características internas durante su entrenamiento para relacionar valores de entrada con los valores salida. Se observan resultados muy buenos para este tipo de arquitecturas teniendo en cuenta que muchos de los trabajos han

tenido un enfoque fuerte en la ingeniería de características al momento de generar los datos de entrenamiento.

Para realizar un pronóstico de eventos con los datos disponibles en catálogos nos enfrentamos a un problema de múltiples variables, además de una secuencia temporal donde debemos tener en cuenta las relaciones y dependencias entre estas variables. A continuación se presentan algunos trabajos que abordan el problema de la secuencia temporal en diferentes campos con redes neuronales LSTM, podemos observar varias arquitecturas en las figuras 2, 4 y 5.

Nazaria et al [33], obteniendo datos de early-est [34] el cual es un sistema que ubica y caracteriza terremotos en tiempo real, implementa una red neuronal con 2 capas LSTM que toma como entrada la secuencia de 12 características por minuto durante una ventana de tiempo de 16 minutos por evento para estimar la magnitud final del evento en el último minuto, los datos de entrenamiento se leen desde un archivo en formato de catálogo, obteniendo resultados que se ajustan al pronóstico de la tendencia aunque con muchos altibajos.

Lai et al [35] para encontrar tendencias en el tráfico, producción de energía solar, consumo de energía y tasa de cambios entre países, propone una representación donde es necesario tener una serie de n valores que representan una característica por marca de tiempo y que afectan en el siguiente valor $n+h$ donde h es el horizonte de tiempo, esto junto a una capa convolucional, una capa recurrente con omisión y/o con una capa de atención obtuvo resultados muy buenos para las métricas de error y correlación definidos con valores muy pequeños en el error y altos en correlación para marcas de tiempo cercanas 3,6,12 y 24 respectivamente, estos resultados se pueden observar en la **Figura 3**.

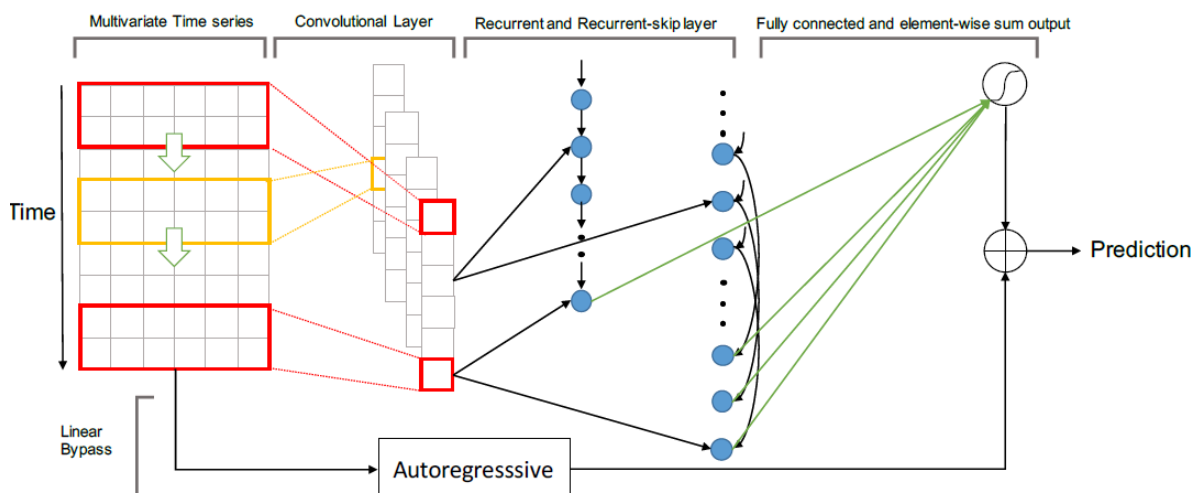


Figura 2. Arquitectura general con datos multivariados en series de tiempo

LST-Skip (17)	RSE	0.1843	0.2559	0.3254	0.4643	0.4777	0.4893	0.4950	0.4973	0.0864	0.0931	0.1007	0.1007	0.0226	0.0280	0.0356	0.0449
	CORR	0.9843	0.9690	0.9467	0.8870	0.8721	0.8690	0.8614	0.8588	0.9283	0.9135	0.9077	0.9119	0.9735	0.9658	0.9511	0.9354
LST-Attn (7)	RSE	0.1816	0.2538	0.3466	0.4403	0.4897	0.4973	0.5173	0.5300	0.0868	0.0953	0.0984	0.1059	0.0276	0.0321	0.0448	0.0590
	CORR	0.9848	0.9696	0.9397	0.8995	0.8704	0.8669	0.8540	0.8429	0.9243	0.9095	0.9030	0.9025	0.9717	0.9656	0.9499	0.9339

Figura 3. Resultados de métricas para red LSTM-Skip y LSTM-Attn

Este trabajo deja en claro una representación para procesar datos y generar modelos de redes neuronales que memoricen patrones para mejorar resultados a problemas multivariados y obtener un pronóstico.

Banna et al [21] propone diferentes modelos utilizando capas LSTM para entrenar con diferentes indicadores sísmicos calculados a partir de datos del Servicio Geológico de Estados Unidos (USGS por sus siglas en inglés), y acotados en un área geográfica. Se calcula el valor-b, la desviación cuadrada promedio, el déficit de magnitud, días transcurridos desde el último sismo de gran magnitud, la magnitud promedio, la tasa de la raíz cuadrada de energía liberada, el tiempo promedio entre eventos característicos y el coeficiente de variación de tiempo promedio. Con esto se busca realizar el pronóstico de la ubicación y la ocurrencia de eventos sísmicos.

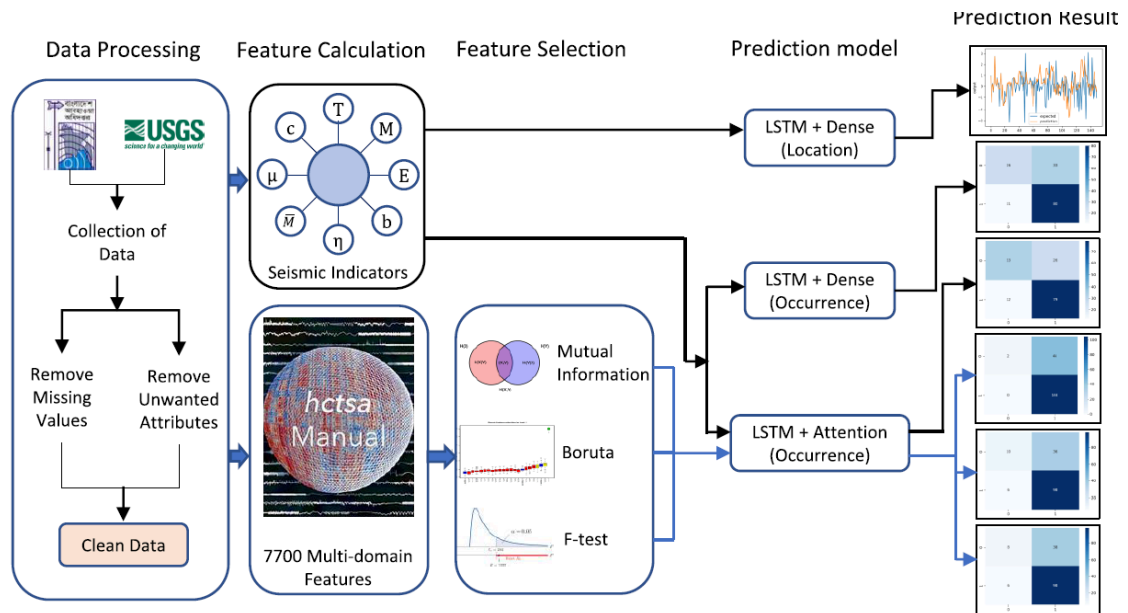


Figura 4. Arquitectura general con procesamiento de datos, cálculo de características, generación de modelos para el pronóstico, resultados en la ocurrencia de eventos (si sucede o no) y en ubicación

En este trabajo se entrenan diferentes arquitecturas de redes neuronales utilizando la librería Keras Tuner [36], la cual está enfocada en la búsqueda de hiperparámetros en diferentes modelos con el objetivo de mejorar su desempeño, encontrando una precisión del 74.67% con los 8 indicadores sísmicos con el mejor modelo, aproximadamente 20% más de los resultados

respecto a otros trabajos mencionados para identificar sismos y un RMSE de 1.24818 para pronóstico de la ubicación presentando cierta dificultad en estimar “outliers”.

Wang et al [37] propone manejar una entrada bidimensional en una red neuronal LSTM para correlacionar secuencias de eventos sísmicos de forma temporal y espacial, donde un vector x_t corresponde a datos de terremotos en diferentes ubicaciones donde cada t hace referencia a una subregión entre M regiones, teniendo así L cantidad de vectores conformando una matriz $M \times L$. Para cada uno de los valores x_t se escoge 1 si ocurrió algún evento en la región o 0 en caso contrario. Obteniendo una precisión aproximada promedio de 82.4% para una cuadrícula de 5×5 subregiones.

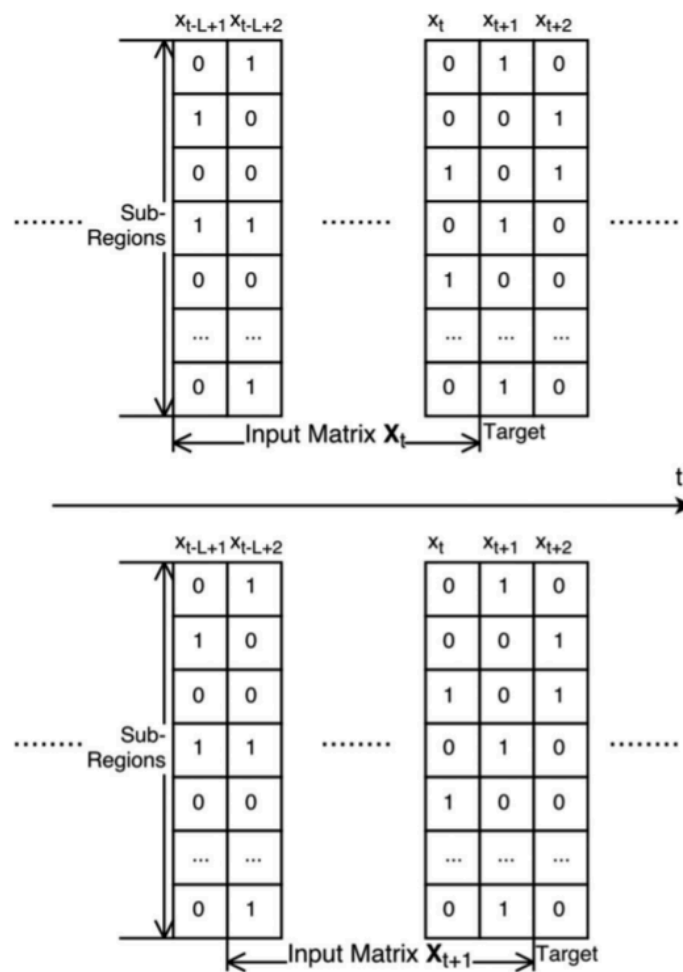


Figura 5. Representación propuesta donde 0 o 1 indica presencia de evento en la subregión y cada columna representa una marca de tiempo

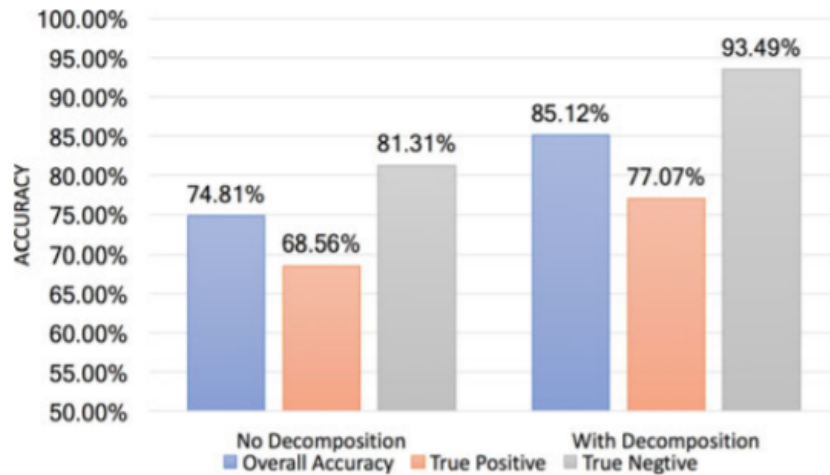


Figura 6. Resultados para una cuadrícula de 3x3 con marcas de tiempo de 1 mes, se observa disminución en el pronóstico de verdaderos positivos

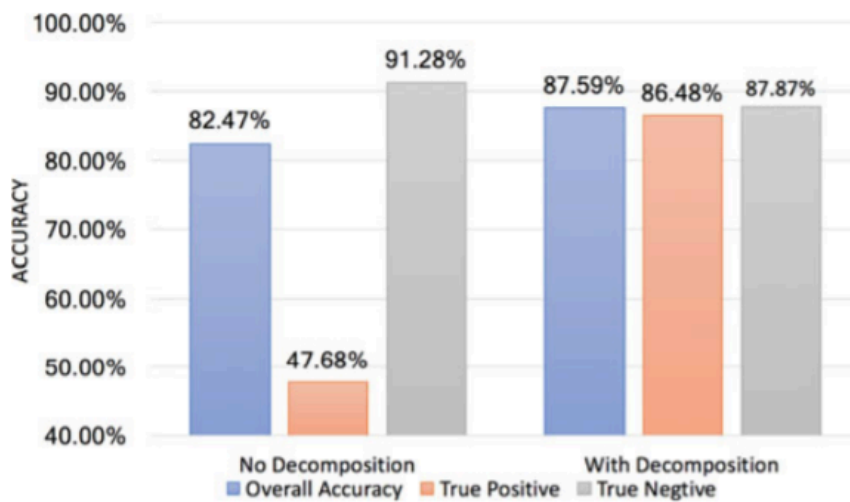


Figura 7. Resultados para una cuadrícula de 5x5 con marcas de tiempo de 1 mes, se observa disminución en el pronóstico de verdaderos positivos

Se observa una precisión promedio del 82.5% para 25 sub regiones, aunque hay una disminución en el pronóstico de verdaderos positivos respecto a 9 subregiones el autor menciona que mediante “descomposición” se puede ayudar al pronóstico agrupando eventos por sub regiones, logrando incrementar el pronóstico de verdaderos positivos a aproximadamente 86.5% para las 25 sub regiones

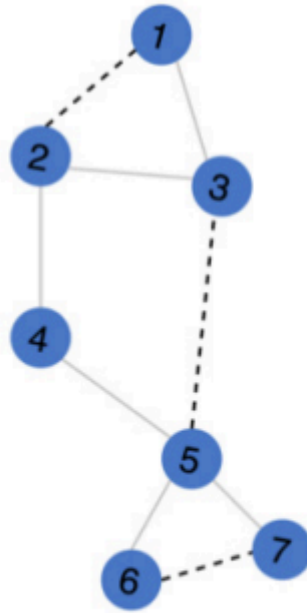
3.2. Redes neuronales de grafos

En los últimos años ha existido un incremento en el interés por representar la estructura de los datos por medio de grafos debido a que se pueden modelar relaciones complejas [38], de acuerdo con Grattarola y Alipi et al [39] las redes neuronales de grafos y el aprendizaje de máquina de grafos se han aplicado en diversos campos de manera exitosa, por ejemplo: interacción de

partículas, sistemas de recomendación, propiedades moleculares, procesamiento de lenguaje natural, etc.

En el trabajo realizado por Chami et al [38] se explica que para grafos, es un desafío definir redes con estructuras ya conocidas, debido a que las conexiones en una red pueden variar significativamente entre diferentes nodos. Tomando como ejemplo una imagen, cada píxel tiene la misma estructura de vecinos (espacio euclídeo), lo cual permite aplicar la misma operación de convolución para cada píxel, sin embargo, en los grafos cada nodo puede tener una estructura diferente de vecinos (espacio no euclídeo). Esto ha llevado a buscar métodos para aprender de esta estructura y reducir la dimensionalidad de la representación del grafo a un vector continuo llamado embedding. Se presenta además una taxonomía sobre estos métodos, clasificándolos entre aprendizaje supervisado y no supervisado, los cuales incluyen el modelo de redes neuronales de grafos propuesto por Scarselli et al [40] que permite procesar directamente grafos cíclicos, acíclicos, dirigidos y no dirigidos, aplicando una función que lleva el grafo G y un nodo n a un espacio euclídeo de m -dimensiones.

Stamille et al [41] menciona que la predicción de enlaces, también conocida como completado de grafos, es un problema común al tratar con grafos. El objetivo es predecir si existen o no aristas para los pares de nodos en un grafo parcialmente observado. Formalmente, dado un grafo $G = (V, E)$, donde V es el conjunto de nodos y $E = O \cup U$ es el conjunto de aristas (observadas y desconocidas), el problema de predicción de enlaces busca estimar U en base a V y O . Este problema también es común cuando se trata con datos de grafos temporales, donde el objetivo es predecir las aristas de un grafo en un momento dado $t+1$ basado en su observación en el momento t .



$$E_o = \{\{1, 3\}, \{2, 3\}, \{2, 4\}, \{4, 5\}, \{5, 6\}, \{5, 7\}\}$$

$$E_u = \{\{1, 2\}, \{3, 5\}, \{6, 7\}\}$$

Figura 8. Grafo con aristas conocidas E_o y aristas desconocidas E_u

3.3. Modelo estándar de minería CRISP-DM

De acuerdo a su nombre en inglés, CRoss Industry Standard Process for Data Mining [31] es un proceso para llevar a cabo proyectos de minería de datos independiente del sector industrial y tecnología usada. Está descrito en términos de una jerarquía, comprendiendo 4 niveles de abstracción: fases, tareas genéricas, tareas específicas e instancias de proceso.

El modelo genérico de referencia que describe el ciclo de vida de un proyecto de minería de datos contiene 6 fases con sus respectivas tareas y salidas. Las fases propuestas son: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelamiento, evaluación y despliegue. La secuencia de estas no es estricta y en la **Figura 8** se puede observar un resumen de las fases y tareas genéricas propuestas para llevarlas a cabo.

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives Background Business Objectives Business Success Criteria Assess Situation Inventory of Resources Requirements, Assumptions, and Constraints Risks and Contingencies Terminology Costs and Benefits Determine Data Mining Goals Data Mining Goals Data Mining Success Criteria Produce Project Plan Project Plan Initial Assessment of Tools and Techniques	Collect Initial Data Initial Data Collection Report Describe Data Data Description Report Explore Data Data Exploration Report Verify Data Quality Data Quality Report	Data Set Data Set Description Select Data Rationale for Inclusion / Exclusion Clean Data Data Cleaning Report Construct Data Derived Attributes Generated Records Integrate Data Merged Data Format Data Reformatted Data	Select Modeling Technique Modeling Technique Modeling Assumptions Generate Test Design Test Design Build Model Parameter Settings Models Model Description Assess Model Model Assessment Revised Parameter Settings	Evaluate Results Assessment of Data Mining Results w.r.t. Business Success Criteria Approved Models Review Process Review of Process Determine Next Steps List of Possible Actions Decision	Plan Deployment Deployment Plan Plan Monitoring and Maintenance Monitoring and Maintenance Plan Produce Final Report Final Report Final Presentation Review Project Experience Documentation

Figura 9. Fases e ítems propuestos por el modelo genérico CRISP-DM

4. Metodología

Esta sección presenta los procesos utilizados en esta investigación para preparar y procesar datos, entrenar modelos de redes neuronales, además de evaluar los resultados. Esta metodología está inspirada en el flujo de trabajo CRISP-DM documentado anteriormente y se presenta de una forma sencilla en las siguientes etapas: Entendimiento de los Datos, Conjunto de Datos, Modelos de Redes Neuronales, Evaluación y Resultados.

Durante el desarrollo de esta investigación:

- Se exploraron artículos científicos cuyo principal objetivo fue el pronóstico de terremotos (probabilidad de ocurrencia, características) enfocados en el ámbito de las ciencias de datos, aprendizaje de máquina y redes neuronales.
- Se exploraron artículos científicos con las bases teóricas de aprendizaje de máquina en grafos y trabajos relacionados con redes neuronales de grafos GNN.
- Se definieron las tecnologías para llevar a cabo el desarrollo, modelamiento, entrenamiento y ajuste de los varios modelos de redes neuronales a explorar.

4.1. Entendimiento de los datos

En esta investigación se descargaron los datos de eventos sísmicos desde el Servicio Geológico de Estados Unidos, el cual tiene una base de datos abierta, accesible mediante un api REST y la cual recopila y valida varias fuentes de eventos sísmicos.

Se seleccionaron eventos entre latitudes $^{\circ}$ -0.132, 9.796, y longitudes -80.343, -72.466. Los datos se almacenaron en un archivo csv, se procesaron en Python utilizando la librería Pandas, filtrando los datos faltantes y convirtiendo tipos de datos respectivamente. Se seleccionaron las siguientes características: latitud, longitud, magnitud y profundidad, también se calculó la diferencia de tiempo entre los eventos.

En las figuras 10 y 11 podemos observar cómo están representados los sismos por profundidad y magnitud en el área seleccionada y en la figura 12, podemos observar la distribución de los datos, principalmente la latitud, longitud, magnitud y profundidad.

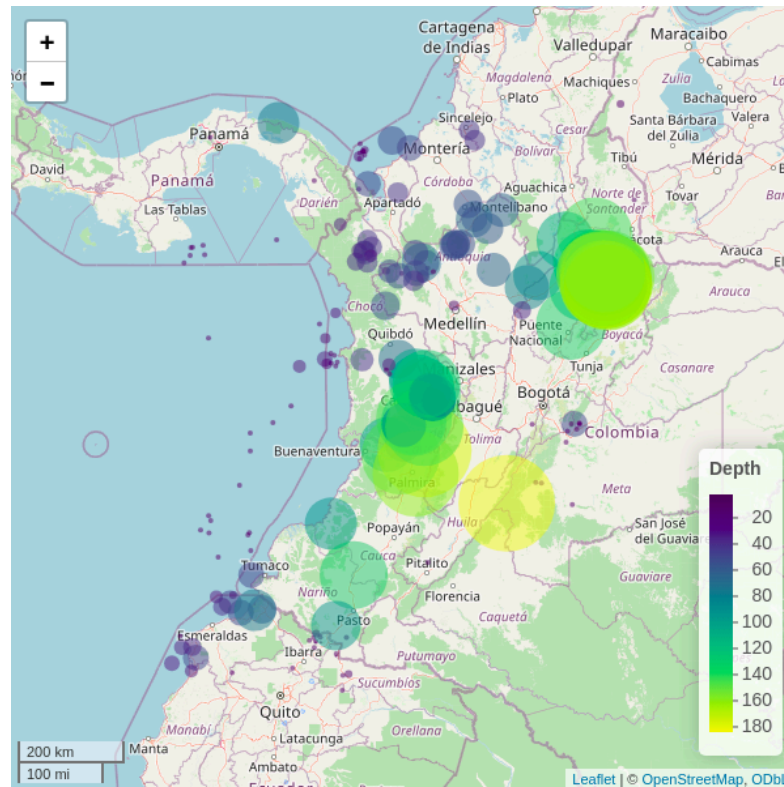


Figura 10. Representación de terremotos por profundidad, se observa relación directa de la profundidad y la ubicación en las cordilleras

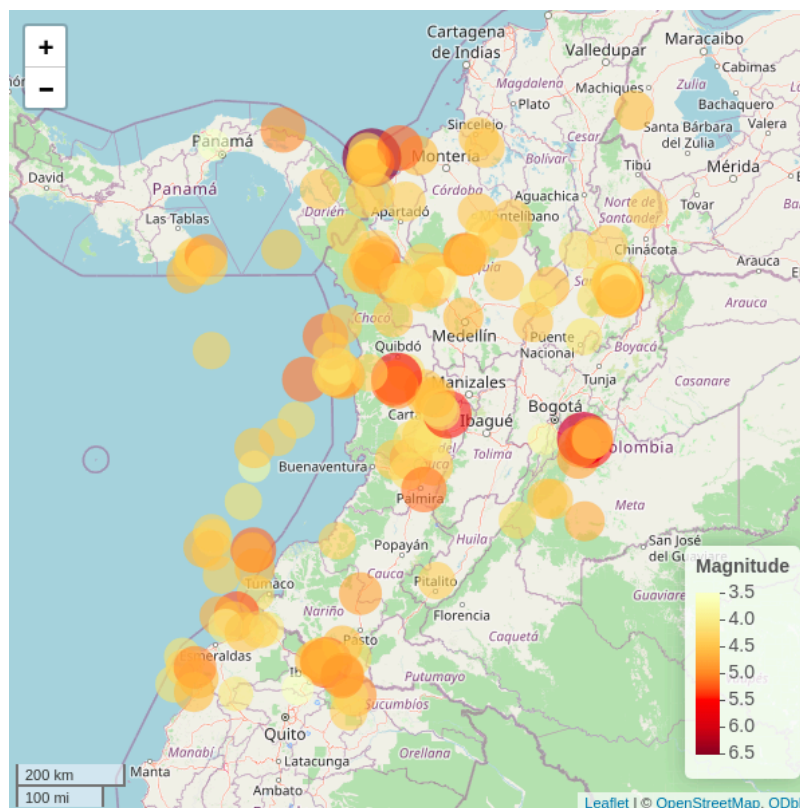


Figura 11. Representación de terremotos por magnitud, se observan bastantes eventos en el rango de 4 a 4.5 en la escala Gutenberg–Richter

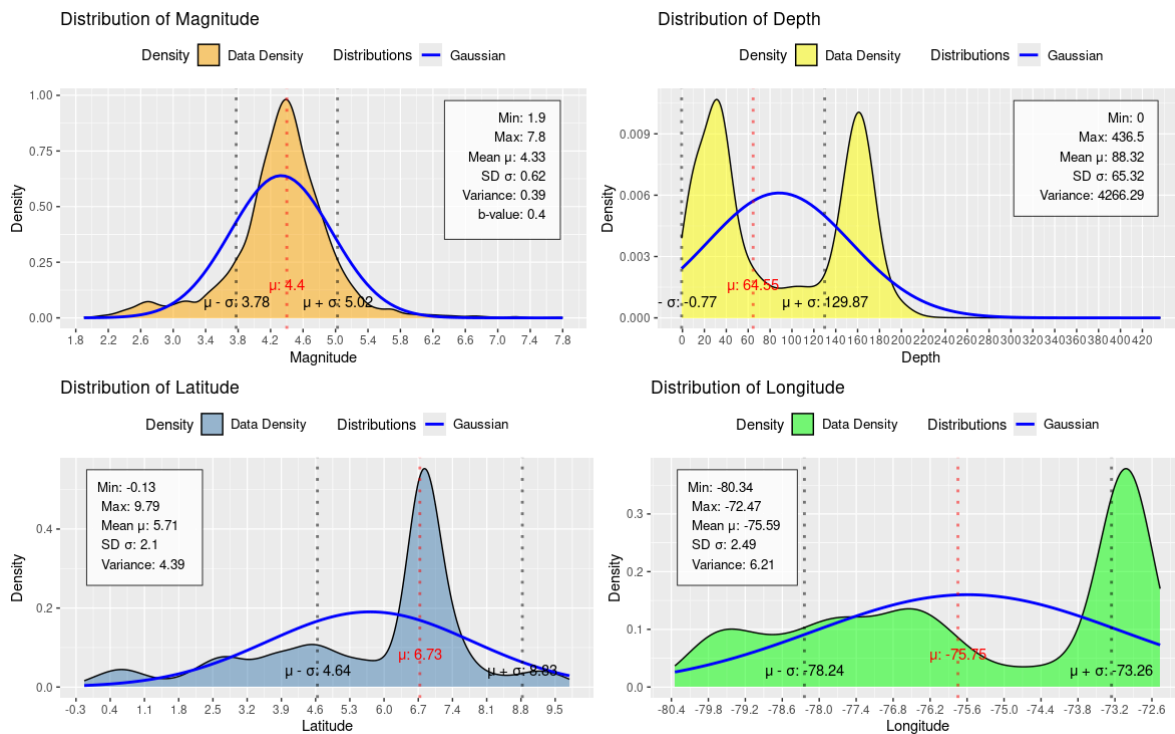


Figura 12. Distribución de las características principales, magnitud, profundidad, latitud y longitud

La **Figura 12** presenta distribuciones de cuatro variables diferentes: Magnitud, Profundidad, Latitud y Longitud. Cada panel muestra la densidad de los datos reales junto con una curva de distribución normal (Gaussiana) ajustada.

Variable	Min	Max	Media (μ)	SD (σ)	Varianza	Otro
Magnitude	1.9	7.8	4.3	0.62	0.39	b-value: 0.4
Depth (m)	254	426.5	88.32	65.32	4266.29	0.02
Latitude	-0.13	9.79	5.71	2.1	4.39	
Longitude	-80.34	-72.47	-75.59	2.49	6.21	

Tabla 1. Estadísticas clave presentadas en la **Figura 12**

4.1.1. Distribución de Magnitud (Arriba a la izquierda)

Se observa que la distribución de la magnitud tiene valores que oscilan entre 3.5 y 6.5, el area sombreada en amarillo muestra la densidad de los datos reales de acuerdo a como aparecen en el catalogo, la distribución tiene un pico cerca de 4.4 donde encontramos la media, además se compara con la distribución normal ajustada representada mediante la curva azul. Podemos interpretar que la distribución está bastante centrada alrededor de su media.

4.1.2. Distribución de Profundidad (Arriba a la derecha)

Se observa que la distribución de la profundidad presenta valores en el rango de los 254 y 246 metros, la distribución presenta dos picos aproximadamente en los 25 y 160 metros, por lo que podemos decir que la mayoría de eventos ocurren a estas distancias y muy pocos ocurren a distancias más profundas como 420 metros.

4.1.3. Distribución de Latitud (Abajo a la izquierda)

Se observa que la distribución de la latitud presenta valores en el rango de -0.13 y 9.79 grados, esta distribución tiene dos picos, uno a los 4.6 grados y el otro a los 6.7 grados en la media, la distribución se encuentra sesgada hacia la derecha y la mayor densidad de datos se encuentran entre los 6 y 8 grados.

4.1.4. Distribución de Longitud (Abajo a la derecha)

Se observa que la distribución de la longitud presenta valores en el rango de -80.33 y -72.56 grados, esta distribución también presenta dos picos, uno a los 76.4 grados y otro a los 73 grados, la distribución tiene un comportamiento bimodal y está sesgada a la derecha

4.2. Conjuntos de datos

Una vez procesamos los datos de catálogo de sismos y realizamos el análisis de las características, procedimos a generar los conjuntos de datos para entrenar las redes neuronales LSTM

4.2.1. Conjuntos de datos para redes neuronales LSTM

Siguiendo lo documentado en el marco referencial, se generaron secuencias de las características obtenidas, teniendo como objetivo encontrar el valor siguiente en la secuencia, en nuestro caso, encontrar la categoría del siguiente evento, en base a la característica seleccionada. Se hizo una división de valores por cuantiles para categorizar cada característica.

Las secuencias se exploraron de acuerdo al parámetro **lookback** durante la exploración de hiper parámetros, generando diferentes matrices de tamaño (S,L) obteniendo un conjunto de datos de tamaño (N,S,L) , en la siguiente figura se expresa la estructura del conjunto de datos para la red lstm.

$$\begin{array}{c}
 W_{t:1 \rightarrow N} \\
 \begin{bmatrix} F_{1,t} & F_{1,t+1} & \cdot & \cdot & F_{1,t+L} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ F_{S,t} & F_{S,t+1} & \cdot & \cdot & F_{S,t+L} \end{bmatrix} \Rightarrow [C_{t+L+1}] \\
 \begin{array}{cc}
 \text{input} & \text{target}
 \end{array}
 \end{array}$$

Figura 13. Conjunto de datos para una red neuronal con capas lstm

En la Figura podemos identificar las variables **F**, **S**, **L**, **W**, **t**, **N** y **C** donde:

- **F_{s,L}** es un serie de valores de una característica
- **S** es el identificador de secuencia que coincide con el número de características del catálogo más las características aumentadas
- **L** es el **lookback** o largo de la secuencia
- **W** es la ventana de eventos con índice de tiempo **t**, que toma valores desde 0 hasta **N**, donde $N = \text{eventos} - \text{lookback} - 1$
- **C** es la clase de la característica objetivo escogida que le sigue a la secuencia anterior

Además se construyeron redes complejas de sismos para cada ventana y se calcularon las características propuestas: degree centrality, clustering, betweenness centrality, closeness centrality, pagerank

En la siguiente figura, podemos observar cómo se crea una red de sismos

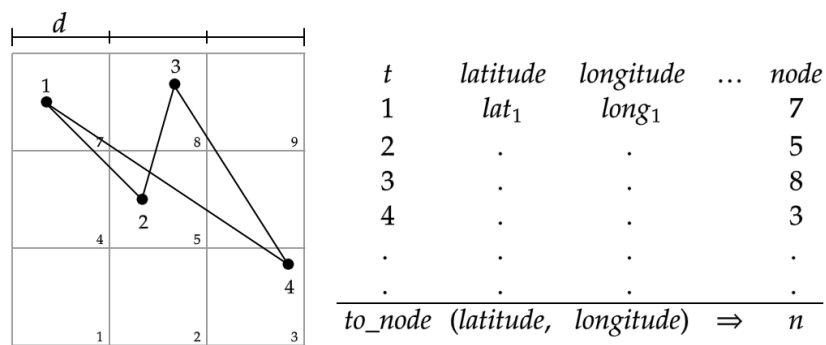


Figura 14. Proceso de creación de redes de sismos, para cada evento en el catálogo de eventos se asigna un nodo de acuerdo a la latitud, longitud y la tamaño **d** del nodo en kilómetros. Los nodos se conectan en orden cronológico **t**

los nodos se calculan desde la latitud y longitud mínima, dividiendo de forma entera las distancias vertical y horizontal en kilómetros por la distancia **d**

4.2.2. Conjuntos de datos para redes de grafos

Siguiendo el proceso documentado por Stamile et al [41] en la predicción de enlaces, después de procesar el catálogo de eventos, se le asignó a cada evento en el catálogo el nodo que le corresponde, se aplicó el proceso descrito en la **Figura 14**, teniendo como entrada diferentes tamaños del nodo, 30 km, 50 km y 100 km y se generó un grafo para cada tamaño de nodo.

Se entrenó un algoritmo Node2Vec [42] con cada grafo para obtener un espacio de vectores, se aplicó un operador Hadamard a los nodos de cada arista, el operador toma una arista y multiplica la representación vectorial de sus nodos en el espacio de Node2Vec, para finalmente generar un vector significativo que identifica la arista en el espacio.

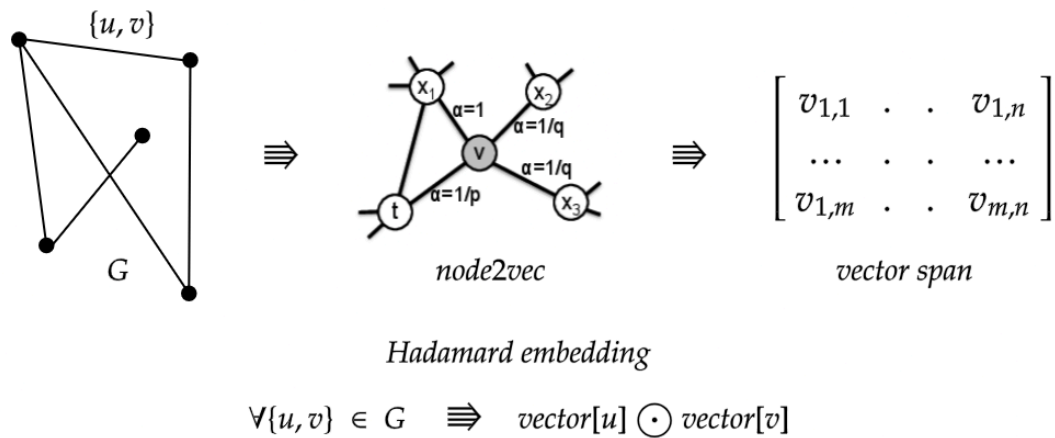


Figura 15.

Además se utilizó un algoritmo llamado EdgeSplitter, para realizar la división de entrenamiento y prueba de las aristas del grafo, este algoritmo genera inicialmente un muestreo aleatorio de aristas conocidas y desconocidas, las cuales se retiran del grafo para el conjunto de prueba y finalmente se repite el mismo proceso para obtener el conjunto de entrenamiento, en cada caso se codifican las aristas conocidas como 1 y desconocidas como 0, en el espacio generado por Node2Vec aplicando Hadamard embedding.

4.3. Desarrollo y Entrenamiento de Redes Neuronales

Para el desarrollo de las redes neuronales se utilizó la librería pytorch, acompañando el entrenamiento con la librería ray tune, la cual es un framework que permite distribuir la búsqueda de hiperparametros, para encontrar el mejor modelo.

4.3.1. Redes Neuronales LSTM

Se implementó un modelo con número de unidades LSTM parametrizable, una capa flatten y una salida lineal

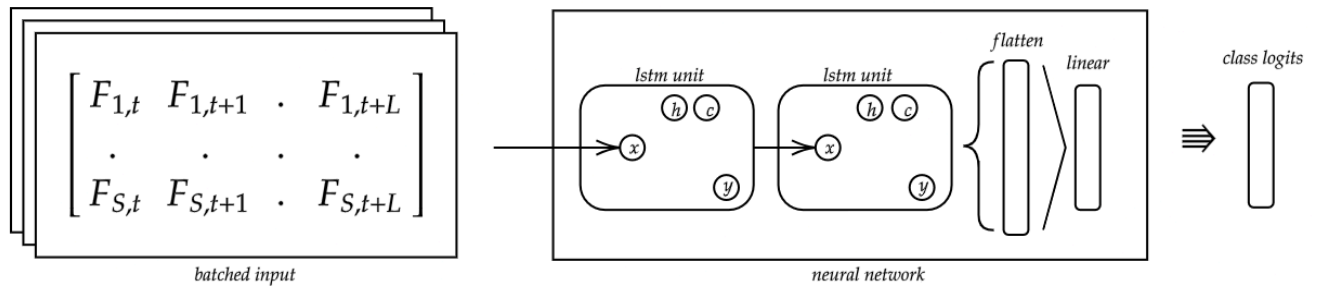


Figura 15. Representación de la red neuronal implementada, la red neuronal toma como entrada grupos de matrices del conjunto de datos

4.3.2. Redes Neuronales de Grafos

Se implementó un modelo de red neuronal simple con 3 capas densas, con 2 capas ocultas con unidades parametrizables

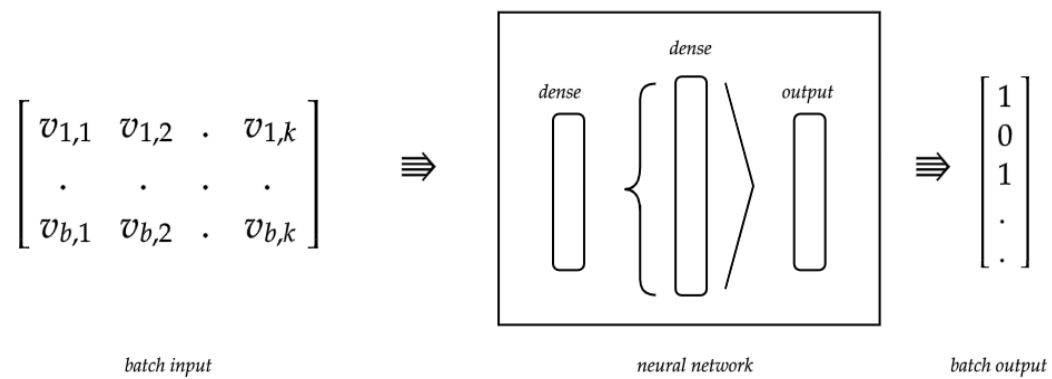


Figura 16. Representación de la red neuronal implementada para entrenar con embedding de grafos

4.4. Resultados y Evaluación

Como resultado de la implementación de la metodología propuesta, se presenta una arquitectura de software enfocada en el procesamiento y análisis de catálogos sismológicos, entrenamiento de modelos de redes neuronales y ajuste de hiperparámetros en búsqueda de las mejores métricas posibles para lograr el objetivo propuesto.

4.4.1. Arquitectura y Diagrama de Clases

La implementación de esta investigación se llevó a cabo en el lenguaje Python versión 3.12, con las dependencias: pytorch, ray tune, Pandas, geopy, Geoandas, Scipy, Scikit-learn, Matplotlib, entre otras. Se llevó a cabo un control de versión del código con git/github, se creó una librería de Python llamada **earthquakes** donde se alojaron sub paquetes con las implementaciones respectivas de entrenamiento y ajuste de hiper parámetros para las redes neuronales con capas LSTM y redes de grafos. Se expuso una interfaz de línea de comandos mediante la librería **click**.

Se implementó un paquete llamado **earthquakes.data**, el cual contiene las siguientes clases, que pueden ser observadas en la **Figura 13**:

- **USGS**: esta clase implementa la conexión con el api del Servicio Geológico de Estados Unidos y permite la descarga de los datos en formato .csv
- **EarthquakeData**: esta clase, recibe los datos descargados, e implementa la limpieza y el procesamiento de estos. Implementa la creación de secuencias de características de eventos, el procesamiento de características categóricas, la normalización de los datos, y la división de entrenamiento y prueba
- **Grid**: esta clase implementa la lógica encargada de representar un evento en un nodo, siguiendo lo propuesto anteriormente por León et al [30], un evento con posición (latitud, longitud), va a estar ubicado en un nodo N, de acuerdo a la distancia entre cada nodo y el área seleccionada

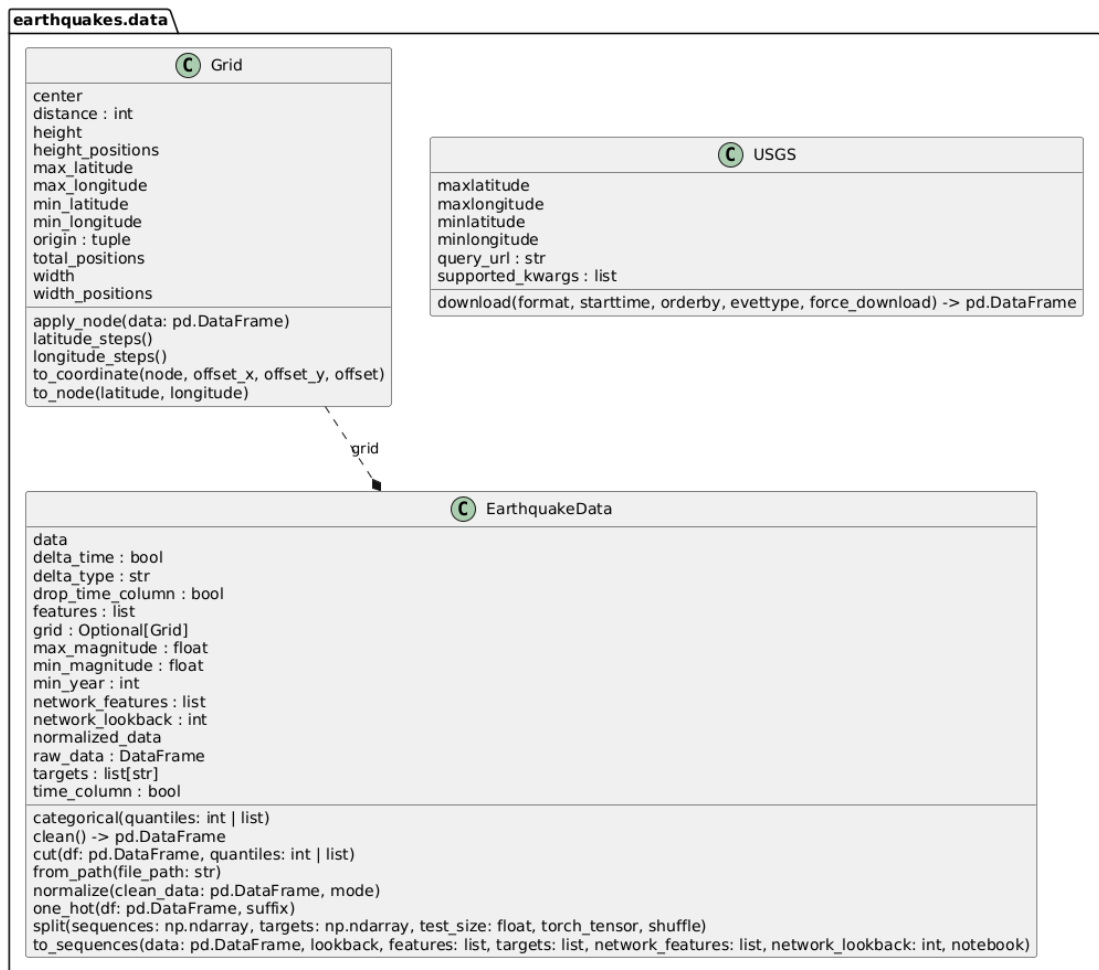


Figura 17. Diagrama de clases del paquete **earthquakes.data**

Se implementó un paquete llamado **earthquakes.lstm**, el cual contiene las siguientes clases, que pueden ser observadas en la **Figura 14**

- **base.BaseTrainable:** se extiende de **ray.tune.Trainable** e implementa la lógica correspondiente para inicializar y entrenar un experimento el cual está conformado por un modelo LSTM y diferentes hiperparámetros.
- **classification.ClassificationTrainable:** se extiende de **ray.train.Trainable** e implementa la lógica necesaria para realizar entrenamiento de clasificación con características categóricas.

loss	mean loss	test loss	lookback	test size	batch size	hidden size	lstm layers	lr	max epochs	accuracy	quantiles
1014.889	0.617	449.927	33	0.176	2	80	4	0.000	60	0.640	2
823.150	0.590	733.873	144	0.282	2	97	2	0.000	25	0.634	2
133.589	0.648	690.542	102	0.268	14	129	8	0.001	19	0.606	2
126.626	0.681	367.758	147	0.139	18	144	7	0.000	58	0.582	2
258.172	0.683	619.781	99	0.231	8	107	8	0.001	49	0.577	2
166.181	0.684	363.903	100	0.136	14	117	9	0.001	46	0.577	2
290.230	0.683	656.259	90	0.244	7	75	9	0.000	34	0.576	2
373.514	0.684	490.382	36	0.180	6	36	9	0.001	50	0.574	2
207.861	0.684	449.364	28	0.164	11	68	9	0.000	58	0.574	2
103.379	0.369	534.021	123	0.141	12	147	3	0.005	25	0.560	2
409.568	0.831	525.436	78	0.117	7	77	4	0.002	45	0.463	3
150.548	0.630	1386.181	54	0.270	12	105	2	0.003	48	0.447	3
38.531	0.214	1464.909	124	0.119	19	33	4	0.001	33	0.430	3
165.997	1.071	1109.243	13	0.263	19	113	2	0.000	53	0.413	3
168.116	1.099	1141.994	42	0.264	19	18	2	0.000	34	0.345	3
1751.982	1.103	753.127	121	0.178	2	70	7	0.004	46	0.344	3
216.893	1.101	861.988	53	0.200	16	32	8	0.002	20	0.343	3
222.096	1.099	477.841	116	0.112	17	52	4	0.001	36	0.343	3
1124.507	1.101	956.388	51	0.221	3	12	9	0.002	13	0.340	3
219.106	1.107	1086.244	37	0.251	15	123	5	0.004	42	0.340	3
405.722	1.071	1245.538	65	0.227	8	129	5	0.000	25	0.411	4
173.115	0.978	848.971	106	0.134	19	101	2	0.001	28	0.397	4
266.301	1.318	790.450	148	0.158	16	76	5	0.000	52	0.387	4
249.980	1.152	902.233	136	0.156	15	22	2	0.008	36	0.382	4
302.347	1.368	854.470	47	0.159	15	32	8	0.004	61	0.340	4
419.391	1.371	763.909	66	0.142	11	139	6	0.001	28	0.340	4
288.732	1.368	1061.811	49	0.197	15	60	9	0.000	40	0.339	4
292.519	1.373	772.410	16	0.143	16	83	9	0.003	68	0.339	4
690.810	1.368	1284.046	17	0.237	6	36	8	0.000	55	0.339	4
1139.644	1.371	949.657	14	0.163	4	59	5	0.002	48	0.282	4

Tabla 3. Resultados de experimentos de redes neuronales LSTM, entre líneas se observa los modelos con mejor accuracy para cada franja de cuantil

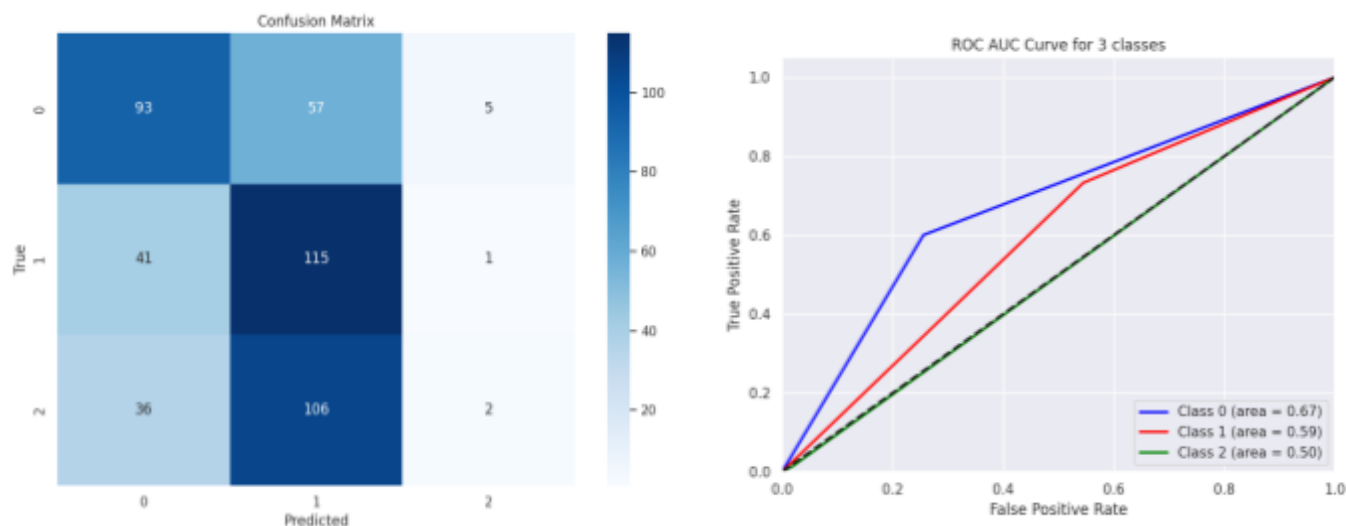


Figura 20. Matriz de confusión del modelo con 46% de accuracy, muestra capacidad del modelo de diferenciar la clase 0 y la clase 1, pero no la clase 2

4.4.3. Experimentos con Redes Neuronales LSTM con Características de Redes

De igual forma que en el punto anterior, se incluyó en cada ventana de eventos el cálculo de un grafo y sus características, variando cada uno el tamaño del nodo con 30, 50 y 100 kilómetros

loss	test loss	lookback	test size	batch size	hidden size	lstm layers	lr	max epochs	accuracy	quantiles	network features	node size
112.576	433.087	82	0.173	18	20	6	0.003	39	0.648	2	betweenness	30
114.802	248.123	104	0.103	19	133	7	0.004	18	0.647	2	clustering	30
513.069	273.118	122	0.110	4	106	2	0.001	58	0.644	2	degree	30
36.951	1394.480	47	0.131	7	76	3	0.000	38	0.643	2	pagerank	30
205.887	686.295	69	0.270	9	109	4	0.001	63	0.641	2	clustering	30
705.396	1115.538	149	0.276	4	19	4	0.002	38	0.473	3	degree	30
1310.758	1208.737	23	0.281	2	131	6	0.000	23	0.465	3	degree	30
312.249	985.935	145	0.239	9	64	2	0.008	18	0.465	3	betweenness	30
404.635	416.466	79	0.102	9	138	9	0.000	55	0.464	3	betweenness	30
366.500	603.875	121	0.153	9	127	5	0.001	19	0.463	3	pagerank	30
269.973	1138.736	105	0.225	14	73	8	0.001	38	0.408	4	clustering	30
527.298	1006.915	75	0.166	6	68	2	0.003	30	0.391	4	clustering	30
212.685	813.492	148	0.161	19	64	6	0.000	67	0.375	4	pagerank	30
584.423	1129.172	33	0.207	6	114	2	0.005	59	0.366	4	clustering	30
349.505	1004.802	52	0.135	9	126	5	0.002	58	0.366	4	betweenness	30
1063.339	456.658	13	0.152	2	36	3	0.007	12	0.651	2	pagerank	50
128.613	602.748	147	0.249	14	69	4	0.000	54	0.647	2	pagerank	50
249.543	417.286	82	0.162	8	12	2	0.001	45	0.640	2	pagerank	50
119.376	708.599	87	0.284	14	37	2	0.007	38	0.639	2	degree	50
98.342	722.712	96	0.290	18	78	2	0.005	53	0.627	2	clustering	50
36.121	2202.229	98	0.153	4	143	3	0.002	22	0.488	3	pagerank	50
402.819	1202.975	36	0.300	7	124	4	0.002	52	0.487	3	closeness	50
406.882	1182.266	139	0.293	7	101	7	0.000	66	0.480	3	pagerank	50
192.634	1022.804	20	0.184	11	80	2	0.004	32	0.464	3	betweenness	50
151.634	834.529	146	0.191	17	73	2	0.000	17	0.461	3	degree	50
728.305	1355.979	27	0.273	5	138	4	0.000	41	0.435	4	betweenness	50
308.907	959.463	137	0.194	13	70	5	0.001	45	0.432	4	closeness	50
350.506	836.819	10	0.165	12	116	3	0.000	39	0.429	4	clustering	50
631.413	1261.671	128	0.250	6	59	6	0.007	13	0.418	4	degree	50
664.843	1094.966	10	0.212	6	117	2	0.000	46	0.409	4	closeness	50
294.545	355.232	49	0.142	7	128	6	0.000	26	0.671	2	betweenness	100
297.019	685.403	18	0.280	6	114	4	0.001	57	0.662	2	closeness	100
720.330	352.685	84	0.145	3	56	6	0.003	39	0.657	2	pagerank	100
126.843	320.770	13	0.126	17	25	6	0.003	58	0.650	2	closeness	100
224.071	724.038	45	0.289	8	83	4	0.000	13	0.645	2	degree	100
254.312	846.157	94	0.202	11	62	4	0.001	11	0.467	3	betweenness	100
191.379	1270.860	140	0.289	12	51	2	0.001	23	0.466	3	clustering	100
183.231	497.836	116	0.106	16	27	5	0.000	23	0.461	3	degree	100
170.810	963.481	62	0.234	19	60	2	0.000	13	0.461	3	pagerank	100
696.927	832.365	53	0.163	6	140	8	0.000	41	0.417	4	pagerank	100
300.884	1110.282	13	0.219	13	35	3	0.004	62	0.403	4	degree	100
231.743	1483.400	128	0.296	16	140	6	0.004	32	0.381	4	closeness	100

Tabla 4. Resultados de experimentos de redes neuronales LSTM con características de redes complejas, se observa que el accuracy es similar para 2 cuantiles con un valor alrededor del 65%.

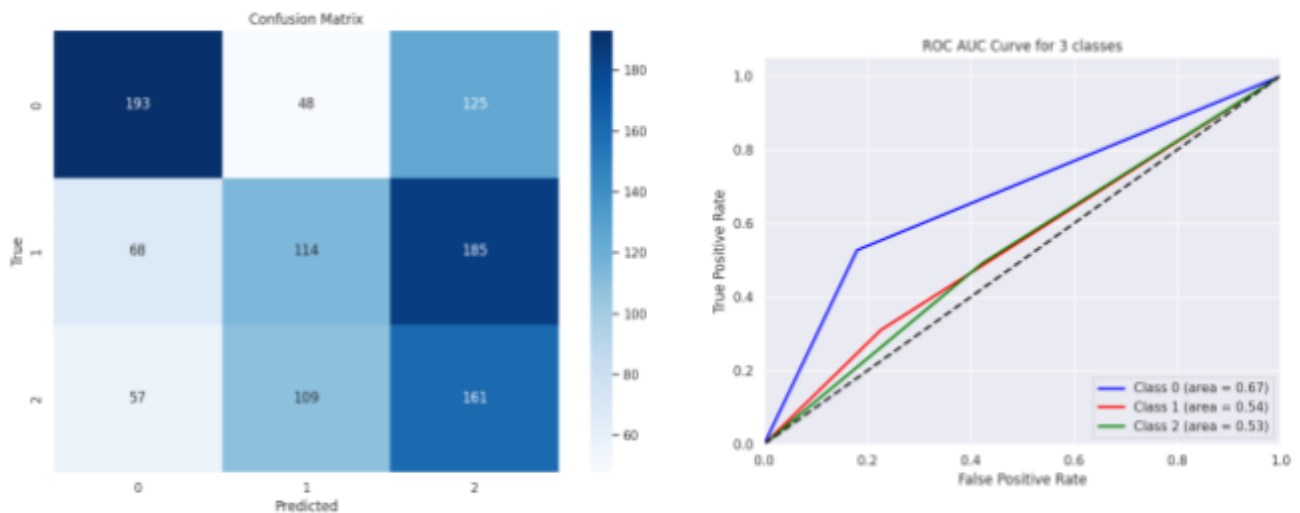


Figura 21. Matriz de confusión del modelo con 47% de accuracy, muestra capacidad del modelo de diferenciar la clase 0 y la clase 1, pero no la clase 2. muy similar al modelo sin características de redes complejas

4.4.4. Experimentos con Redes Neuronales de Grafos

Se llevaron a cabo 3 tipos de experimentos, variando cada uno el tamaño del nodo con 30, 50 y 100 kilómetros y haciendo una división del conjunto de entrenamiento y prueba del 30%

Parameter	Values
Input Dimension	input_dim = 128
Hidden Dimension 1	64, 128, 256
Hidden Dimension 2	32, 64, 128
Learning Rate	tune.loguniform(1e-4, 1e-2)
Batch Size	16, 32, 64

Tabla 5. Espacio de búsqueda de hiperparametros, donde input dim, corresponde al tamaño del embedding

4.4.4.1. Distancia 30km

A continuación se presentan los resultados de experimentos hechos para un tamaño de nodo de 30 km, en la Tabla 3 podemos observar resultados de 10 experimentos ordenados en el orden de mejor accuracy, encontramos un modelo que pudo predecir los enlaces del grafo un 99% de las veces en el conjunto de entrenamiento y 86% de las veces en el conjunto de prueba.

loss	train accuracy	accuracy	input dim	hidden dim1	hidden dim2	learning rate	batch size
0.001769	0.998630	0.861060	128	256	64	0.005702	64
0.010277	0.998434	0.853748	128	256	128	0.000755	64
0.027201	0.990409	0.838665	128	64	128	0.007619	16
0.001787	0.998238	0.837294	128	256	64	0.008100	16
0.404291	0.814054	0.828611	128	128	64	0.001741	64
0.645605	0.659229	0.719835	128	64	64	0.000348	16
0.681024	0.600900	0.684644	128	64	32	0.000223	16
0.686720	0.546878	0.626143	128	64	128	0.000357	64
0.691325	0.511255	0.589580	128	64	32	0.000265	64
0.690199	0.500294	0.516453	128	64	128	0.000300	64

Tabla 6. Experimentos de redes neuronales de grafos y nodo de 30km, se observa el mejor resultado con 99% de accuracy en entrenamiento y 86% en prueba, con unas unidades ocultas de 256 unidades y 64 unidades, una tasa de aprendizaje aproximada del $1e-3$

Para uno de los mejores modelos con 30 km de tamaño de nodo se midieron las siguientes métricas, accuracy en 87.1%, puntaje roc/auc en 87.84%, precisión de 81.55%, puntaje recall de 97%, y puntaje F1 de 88%.

Estas métricas significan que el modelo fue capaz de identificar de forma correcta aproximadamente el 87% de los casos, identificando el 97% de las veces los casos positivos y diferenciando el 88% de las veces entre los casos presentados de acuerdo al puntaje roc/auc.

En la Figura 16 podemos observar la matriz de confusión del modelo, presentando 1070 verdaderos positivos, 852 verdaderos negativos, 242 falsos positivos y 24 falsos negativos, además se presenta un área de curva ROC del 88%.

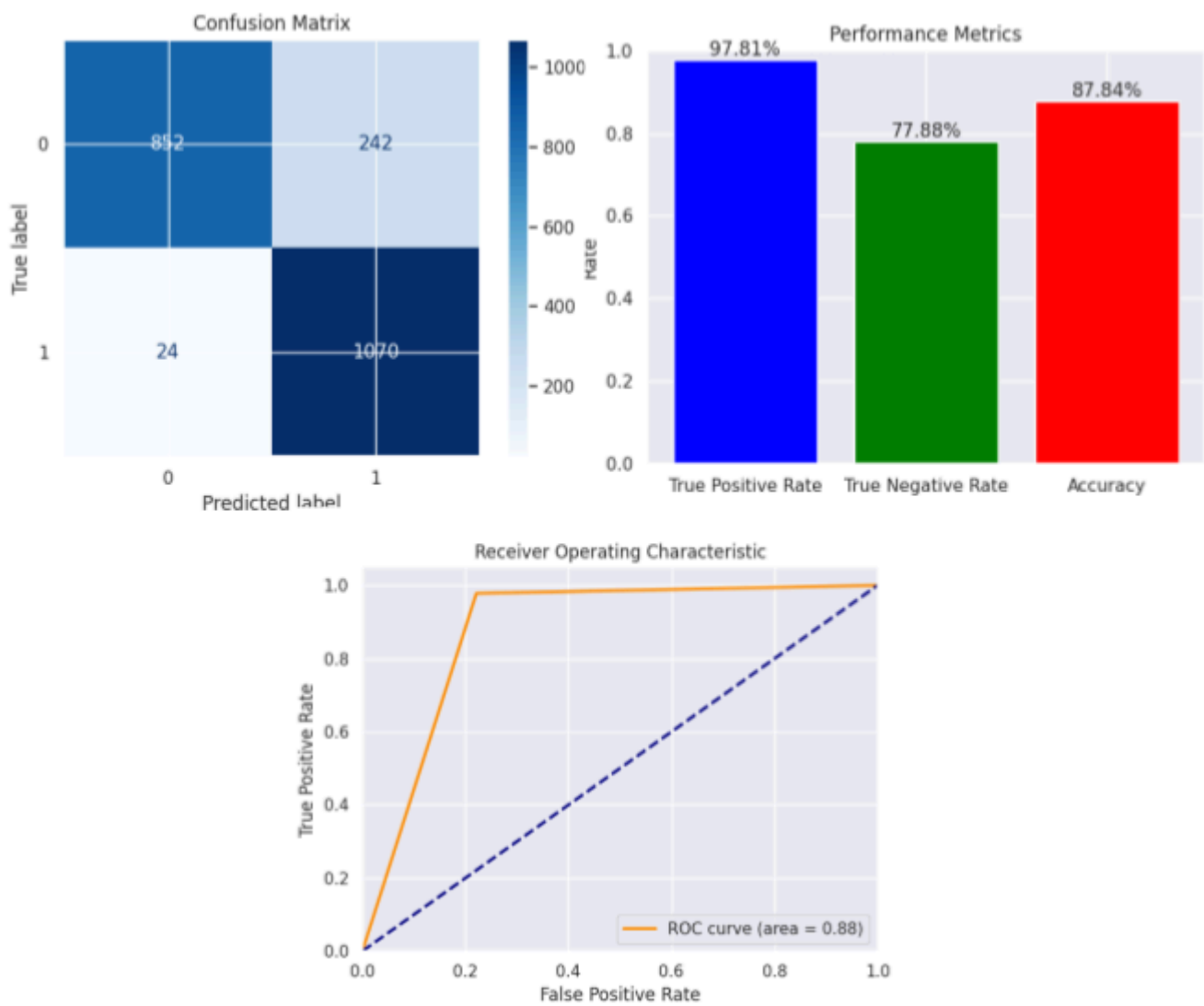


Figura 22. Resultados del mejor modelo con distancia 30km

4.4.4.2. Distancia 50 km

A continuación se presentan los resultados de experimentos hechos para un tamaño de nodo de 50 km, en la Tabla 4 podemos observar resultados de 10 experimentos ordenados en el orden de mejor accuracy, encontramos un modelo que pudo predecir los enlaces del grafo un 98% de las veces en el conjunto de entrenamiento y 74% de las veces en el conjunto de prueba.

Loss	Train Accuracy	Accuracy	Input Dim	Hidden Dim1	Hidden Dim2	Learning Rate	Batch Size
0.032372	0.986559	0.741693	128	256	128	0.004471	16
0.175360	0.938441	0.738558	128	128	64	0.009985	64
0.469090	0.779301	0.735423	128	128	32	0.003753	32
0.496252	0.764247	0.735423	128	128	64	0.001655	32
0.192954	0.933065	0.732915	128	64	128	0.005880	16
0.610217	0.671774	0.710972	128	128	128	0.002931	64
0.690220	0.613172	0.674608	128	64	32	0.000392	32
0.688391	0.613978	0.660815	128	64	128	0.000267	16
0.690271	0.591398	0.660188	128	256	64	0.000264	32
0.693195	0.500000	0.500313	128	64	32	0.000331	64

Tabla 7. Experimentos de redes neuronales de grafos y nodo de 50km, se observa el mejor resultado con 98% de accuracy en entrenamiento y 74% en prueba, con unas unidades ocultas de 256 unidades y 128 unidades, una tasa de aprendizaje aproximada del $1e-3$

Para uno de los mejores modelos con 50 km de tamaño de nodo se midieron las siguientes métricas, accuracy 74.1%, puntaje roc/auc en 80.38%, precisión de 72.61%, puntaje recall de 97.4% y puntaje F1 de 83.2%.

Estas métricas significan que el modelo fue capaz de identificar de forma correcta aproximadamente el 74.1% de los casos, identificando el 97.4% de las veces los casos positivos y diferenciando 80% de las veces entre los casos presentados de acuerdo al puntaje roc/auc.

En la Figura 17, podemos observar la matriz de confusión del modelo, presentando 777 verdaderos positivos, 505 verdaderos negativos, 293 falsos positivos y 20 falsos negativos, además se presenta un área de curva ROC del 80%.

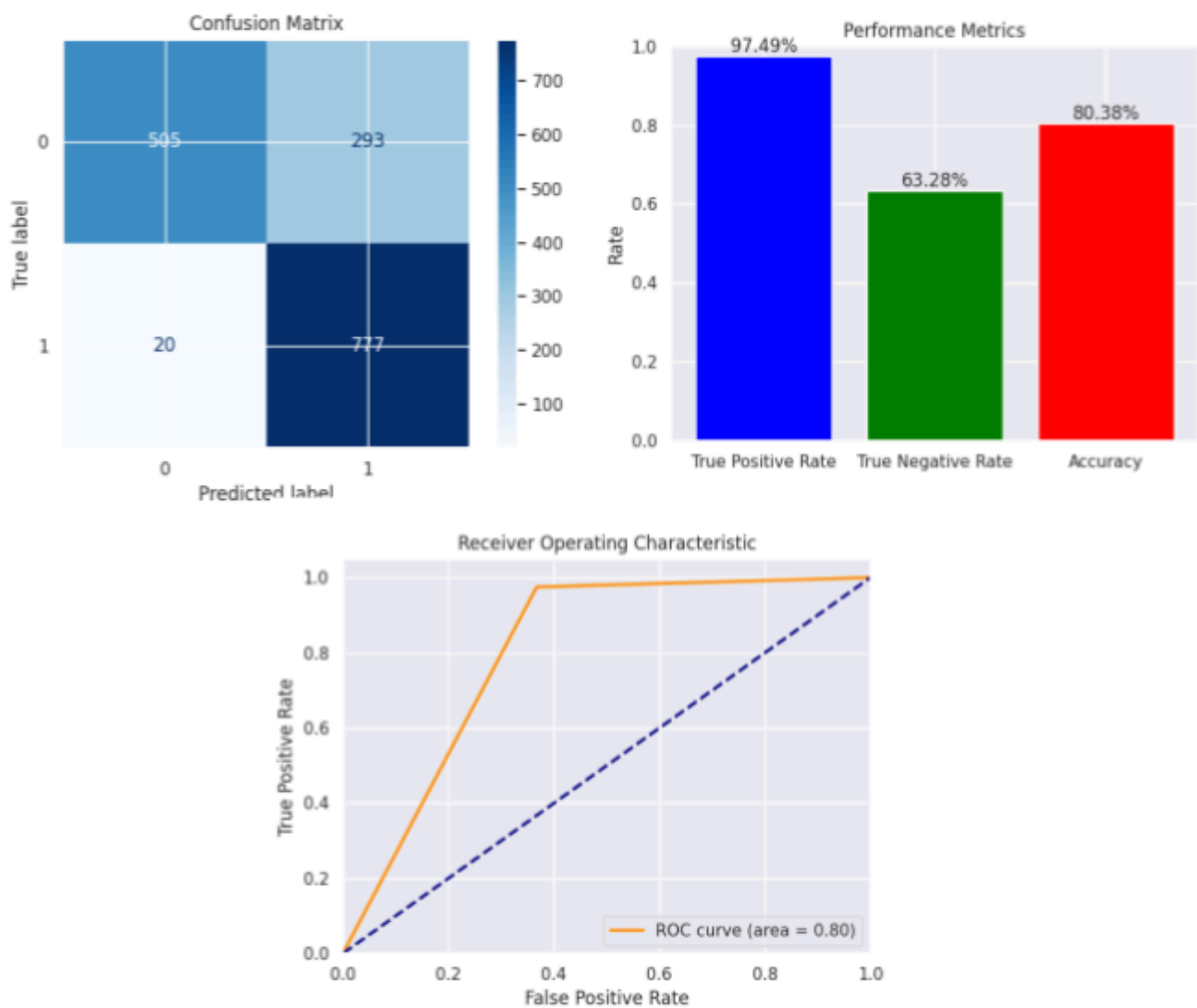


Figura 23. Resultados del mejor modelo con distancia 50km

4.4.4.3. Distancia 100 km

A continuación se presentan los resultados de experimentos hechos para un tamaño de nodo de 100 km. En la Tabla 4 podemos observar resultados de 10 experimentos ordenados según mayor accuracy, encontramos un modelo que pudo predecir los enlaces del grafo un 84% de las veces en el conjunto de entrenamiento y 76% de las veces en el conjunto de prueba.

Loss	Train Accuracy	Accuracy	Input Dim	Hidden Dim1	Hidden Dim2	Learning Rate	Batch Size
0.340570	0.844749	0.766275	128	256	64	0.006214	16
0.346658	0.844749	0.766275	128	128	32	0.005817	16
0.393455	0.817352	0.764141	128	128	32	0.009476	64
0.370865	0.827397	0.753469	128	64	128	0.008232	32
0.396200	0.808676	0.749200	128	64	128	0.006301	16
0.421328	0.799087	0.748132	128	256	32	0.001165	16
0.553075	0.738356	0.718250	128	64	64	0.000126	16
0.692661	0.542466	0.618997	128	128	128	0.000558	32
0.693155	0.499087	0.499466	128	128	128	0.000272	16
0.693512	0.500000	0.499466	128	64	32	0.000371	32

Tabla 8. Experimentos de redes neuronales de grafos y nodo de 100km, se observa el mejor resultado con 84% de accuracy en entrenamiento y 76% en prueba, con unas unidades ocultas de 256 unidades y 64 unidades, una tasa de aprendizaje aproximada del 1e-3

Para uno de los mejores modelos con 100 km de tamaño de nodo se midieron las siguientes métricas, accuracy 76.6%, puntaje roc/auc en 79.38%, precisión de 79.56%, puntaje recall de 79.05%, y puntaje F1 de 79.31%.

Estas métricas significan que el modelo fue capaz de identificar de forma correcta aproximadamente el 76.6% de los casos, identificando el 79.05% de las veces los casos positivos y diferenciando el 79.38% de las veces entre los casos presentados de acuerdo al puntaje roc/auc.

En la Figura 18, podemos observar la matriz de confusión del modelo, presentando 370 verdaderos positivos, 373 verdaderos negativos, 95 falsos positivos y 98 falsos negativos, además se presenta un área de curva ROC del 79%

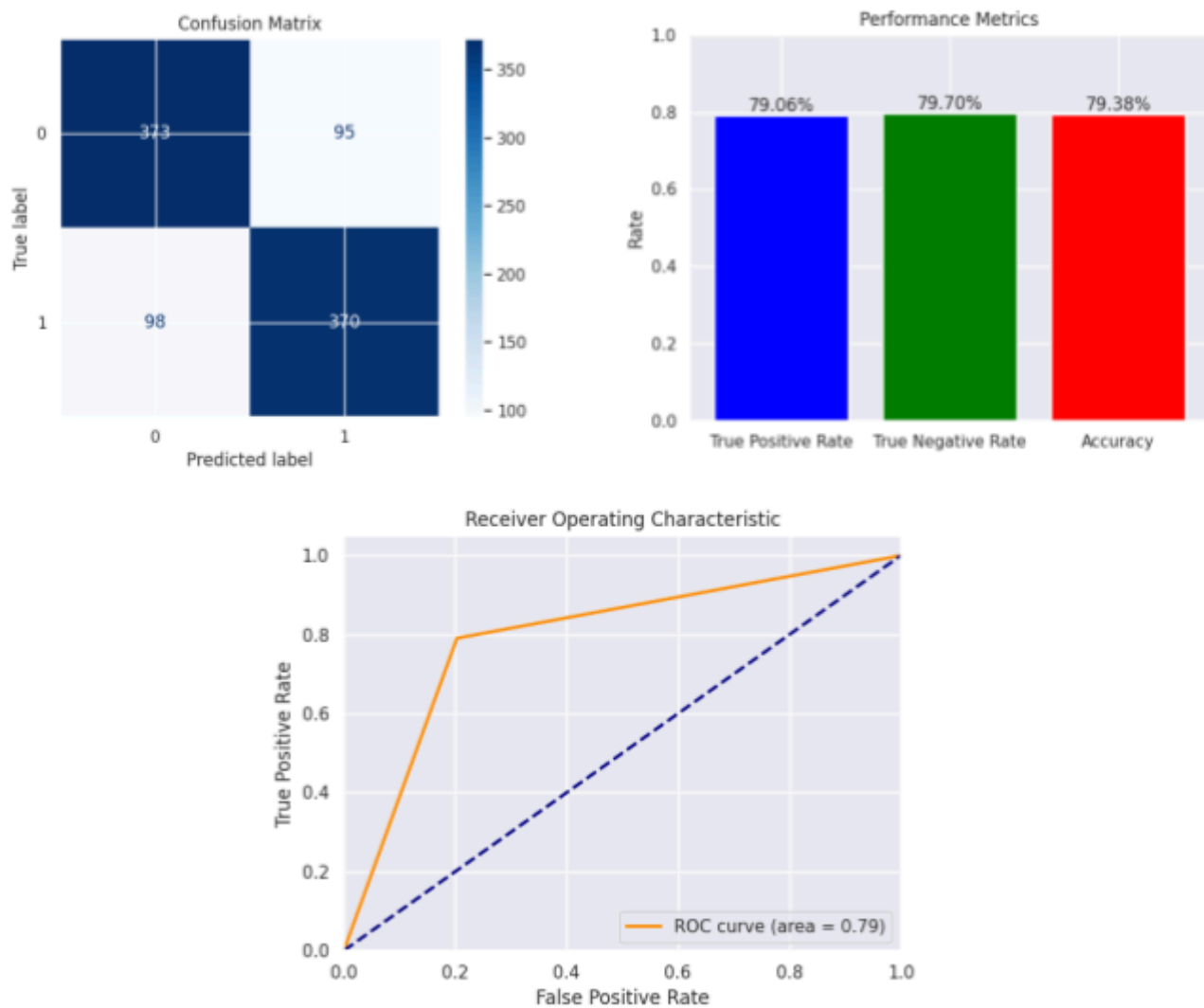


Figura 24. Resultados del mejor modelo con distancia 100 km

5. Conclusiones

- 5.1. Se logró implementar redes neuronales LSTM y de grafos aplicados a los datos de terremotos en Colombia para el área seleccionada.
- 5.2. Se logró documentar un estado del arte que revisa las técnicas, métodos y algoritmos utilizados en otras partes del mundo para abordar el problema de predicción de sismos.
- 5.3. Se obtuvieron resultados similares a los documentados en la literatura cercanos a un 60-80% de precisión en diferentes casos, sin embargo la calidad de predicción de los modelos LSTM se vió impactada significativamente por la falta de datos.
- 5.4. Se observó una mejora mínima de los resultados en las redes neuronales con capas LSTM al momento de incluir características de redes, en ciertos casos, sin embargo el mayor impacto lo tuvo el número de clases, afectando el accuracy con más de 3 clases.
- 5.5. Se observó una mejora de los resultados al incrementar la resolución, es decir, seleccionar un tamaño de nodo de menos kilómetros, en ambos tipos de modelos LSTM/GNN, sin embargo hacer esto requiere más capacidad de cómputo y tiempo.
- 5.6. El trabajo futuro incluye implementar mecanismos de atención o unidades gpt, secuencias de subgrafos y/o servicios dispuestos por OpenAI o terceros como Nixtla TimeGPT-1, de acuerdo a la última tendencia en modelos de inteligencia artificial.

6. Referencias

- [1] M. H. Al Banna *et al.*, «Application of Artificial Intelligence in Predicting Earthquakes: State-of-the-Art and Future Challenges», *IEEE Access*, vol. 8, pp. 192880-192923, 2020, doi: 10.1109/ACCESS.2020.3029859.
- [2] E. Florido, J. L. Aznarte, A. Morales-Esteban, y F. Martínez-Álvarez, «Earthquake magnitude prediction based on artificial neural networks: A survey», *Croatian Operational Research Review*, pp. 159-169, dic. 2016.
- [3] A. Mignan y M. Broccardo, «Neural Network Applications in Earthquake Prediction (1994–2019): Meta-Analytic and Statistical Insights on Their Limitations», *Seismological Research Letters*, vol. 91, n.º 4, pp. 2330-2342, jul. 2020, doi: 10.1785/0220200021.
- [4] P. Jiao y A. H. Alavi, «Artificial intelligence in seismology: Advent, performance and future trends», *Geoscience Frontiers*, vol. 11, n.º 3, pp. 739-744, may 2020, doi: 10.1016/j.gsf.2019.10.004.
- [5] A. Galkina y N. Grafeeva, *Machine Learning Methods for Earthquake Prediction: a Survey*. 2019.
- [6] Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, y P. Gerstoft, «Machine Learning in Seismology: Turning Data into Insights», *Seismological Research Letters*, vol. 90, n.º 1, pp. 3-14, ene. 2019, doi: 10.1785/0220180259.

- [7] Z.-Q. Han y B.-Q. Wang, «The medium- and short-term prediction methods of strong earthquakes based on neural network», *Acta Seismologica Sinica English Edition*, 1997, doi: 10.1007/s11589-997-0054-7.
- [8] I. De Falco, A. Iazzetta, G. Luongo, A. Mazzarella, y E. Tarantino, «The seismicity in the Southern Tyrrhenian area and its neural forecasting», *Pure and Applied Geophysics*, 2000, doi: 10.1007/s000240050003.
- [9] B. Bodri, «A neural-network model for earthquake occurrence», *Journal of Geodynamics*, 2001, doi: 10.1016/S0264-3707(01)00039-4.
- [10] W. Wang *et al.*, «The support vector machine method for time sequence forecasting of strong earthquakes in China's mainland», *Earthquake*, 2005, [En línea]. Disponible en:
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-33646128976&partnerID=40&md5=801c6b3f0c08bcef695da8967685062b>
- [11] S. Sri Lakshmi y R. K. Tiwari, «Model dissection from earthquake time series: A comparative analysis using modern non-linear forecasting and artificial neural network approaches», *Computers and Geosciences*, 2009, doi: 10.1016/j.cageo.2007.11.011.
- [12] F. C. Lin, N. Elhassan, A. Hassan, y A. Yousif, «Forecast of seismic aftershocks using a neural network», en *ICONIP 2002 - Proceedings of the 9th International Conference on Neural Information Processing: Computational Intelligence for the E-Age*, 2002. doi: 10.1109/ICONIP.2002.1198983.
- [13] A. Panakkat y H. Adeli, «Neural network models for earthquake magnitude prediction using multiple seismicity indicators», *International Journal of Neural Systems*, 2007, doi: 10.1142/S0129065707000890.
- [14] A. A. Suratgar, F. Setoudeh, A. H. Salemi, y A. Negarestani, «Magnitude of Earthquake Prediction Using Neural Network», en *2008 Fourth International Conference on Natural Computation*, oct. 2008, pp. 448-452. doi: 10.1109/ICNC.2008.781.
- [15] A. Panakkat y H. Adeli, «Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators», *Computer-Aided Civil and Infrastructure Engineering*, 2009, doi: 10.1111/j.1467-8667.2009.00595.x.
- [16] J. Veri y T. Y. Wah, «Earthquake Prediction Based on the Pattern of Points Seismic Motion», en *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, nov. 2012, pp. 209-212. doi: 10.1109/ACSAT.2012.67.
- [17] A. Morales-Esteban, F. Martínez-Álvarez, y J. Reyes, «Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence», *Tectonophysics*, 2013, doi: 10.1016/j.tecto.2013.02.036.
- [18] J. Reyes, A. Morales-Esteban, y F. Martínez-Álvarez, «Neural networks to predict earthquakes in Chile», *Applied Soft Computing*, vol. 13, n.º 2, pp. 1314-1328, feb. 2013, doi: 10.1016/j.asoc.2012.10.014.
- [19] G. Asencio-Cortés, A. Morales-Esteban, X. Shang, y F. Martínez-Álvarez, «Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure», *Computers and Geosciences*, 2018, doi: 10.1016/j.cageo.2017.10.011.
- [20] Y. Cai, M.-L. Shyu, Y.-X. Tu, Y.-T. Teng, y X.-X. Hu, «Anomaly detection of earthquake precursor data using long short-term memory networks», *Applied Geophysics*, 2019, doi: 10.1007/s11770-019-0774-1.

- [21] M. H. A. Banna *et al.*, «Attention-Based Bi-Directional Long-Short Term Memory Network for Earthquake Prediction», *IEEE Access*, vol. 9, pp. 56589-56603, 2021, doi: 10.1109/ACCESS.2021.3071400.
- [22] M. Böse, F. Wenzel, y M. Erdik, «PreSEIS: A neural network-based approach to earthquake early warning for finite faults», *Bulletin of the Seismological Society of America*, 2008, doi: 10.1785/0120070002.
- [23] S. Lakkos, A. Hadjiprocopis, R. Comley, y P. Smith, «Neural network scheme for earthquake prediction based on the seismic electric signals», en *Neural Networks for Signal Processing - Proceedings of the IEEE Workshop*, 1994. [En línea]. Disponible en:
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-0028699912&partnerID=40&md5=46b2b75c6754a02f06e4362b9f98b247>
- [24] G. A. Rovithakis y F. Vallianatos, «A neural network approach to the identification of Electric Earthquake precursors», *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, 2000, doi: 10.1016/S1464-1895(00)00049-1.
- [25] A. Ifantis y S. Papadimitriou, «Support vector identification of seismic electric signals», *International Journal of Pattern Recognition and Artificial Intelligence*, 2003, doi: 10.1142/S0218001403002484.
- [26] S. Radeva y D. Radev, «Real-time model for earthquake prediction», en *Improvement of Buildings' Structural Quality by New Technologies - Proceedings of the Final Conference of COST Action C12*, 2005. [En línea]. Disponible en:
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84857007188&partnerID=40&md5=92165af4a61858a05156172de7122e29>
- [27] A. Bernardo-Torres y P. Gómez-Gil, «One-step forecasting of seismograms using multi-layer perceptrons», en *2009 6th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2009*, 2009. doi: 10.1109/ICEEE.2009.5393349.
- [28] C. Satriano, Y.-M. Wu, A. Zollo, y H. Kanamori, «Earthquake early warning: Concepts, methods and physical grounds», *Soil Dynamics and Earthquake Engineering*, vol. 31, n.º 2, pp. 106-118, feb. 2011, doi: 10.1016/j.soildyn.2010.07.007.
- [29] J. M. Espinosa-Aranda *et al.*, «Evolution of the Mexican Seismic Alert System (SASMEX)», *Seismological Research Letters*, vol. 80, n.º 5, pp. 694-706, sep. 2009, doi: 10.1785/gssrl.80.5.694.
- [30] D. A. León, J. A. Valdivia, y V. A. Bucheli, «Modeling of Colombian Seismicity as Small-World Networks», *Seismological Research Letters*, vol. 89, n.º 5, pp. 1807-1816, sep. 2018, doi: 10.1785/0220180076.
- [31] R. Wirth y J. Hipp, «CRISP-DM: Towards a standard process model for data mining», *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, ene. 2000, [En línea]. Disponible en:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.198.5133>
- [32] R. J. Geller, «Earthquake prediction: a critical review», *Geophysical Journal International*, vol. 131, n.º 3, pp. 425-450, dic. 1997, doi: 10.1111/j.1365-246X.1997.tb06588.x.
- [33] M. Nazaria, «Predicting the Accuracy of Early-est Earthquake Magnitude Estimates with an LSTM Neural Network: A Preliminary Analysis», *arXiv:2104.05712 [physics]*, abr. 2021, Accedido: 12 de mayo de 2021. [En

- línea]. Disponible en: <http://arxiv.org/abs/2104.05712>
- [34] «Early-est - EArthquake Rapid Location sYstem with EStimation of Tsunamigenesis». Accedido: 27 de octubre de 2021. [En línea]. Disponible en: <http://early-est.rm.ingv.it/warning.html>
 - [35] G. Lai, W.-C. Chang, Y. Yang, y H. Liu, «Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks», *arXiv:1703.07015 [cs]*, abr. 2018, Accedido: 17 de mayo de 2021. [En línea]. Disponible en: <http://arxiv.org/abs/1703.07015>
 - [36] T. O'Malley *et al.*, «Keras Tuner». 2019. [En línea]. Disponible en: <https://github.com/keras-team/keras-tuner>
 - [37] Q. Wang, Y. Guo, L. Yu, y P. Li, «Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach», *IEEE Transactions on Emerging Topics in Computing*, vol. 8, n.º 1, pp. 148-158, 2020, doi: 10.1109/TETC.2017.2699169.
 - [38] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, y K. Murphy, «Machine Learning on Graphs: A Model and Comprehensive Taxonomy», *arXiv:2005.03675 [cs, stat]*, ene. 2021, Accedido: 22 de enero de 2022. [En línea]. Disponible en: <http://arxiv.org/abs/2005.03675>
 - [39] D. Grattarola y C. Alippi, «Graph Neural Networks in TensorFlow and Keras with Spektral», *arXiv:2006.12138 [cs, stat]*, jun. 2020, Accedido: 12 de noviembre de 2020. [En línea]. Disponible en: <http://arxiv.org/abs/2006.12138>
 - [40] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, y G. Monfardini, «The Graph Neural Network Model», *IEEE Transactions on Neural Networks*, vol. 20, n.º 1, pp. 61-80, ene. 2009, doi: 10.1109/TNN.2008.2005605.
 - [41] C. Stamile, A. Marzullo, y E. Deusebio, *Graph Machine Learning: Take Graph Data to the Next Level by Applying Machine Learning Techniques and Algorithms*. Birmingham: Packt Publishing, Limited, 2021.
 - [42] A. Grover y J. Leskovec, «node2vec: Scalable Feature Learning for Networks», en *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, ago. 2016, pp. 855-864. doi: 10.1145/2939672.2939754.