

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Aprendizaje de un espacio de características de manera
automática usando Autoencoders para señales sísmicas del
Volcán Cotopaxi**

Jenny Belén Pupiales Alulema

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como
requisito para la obtención del título de
Ingeniera en Ciencias de la Computación

Quito, 22 de diciembre del 2022

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Aprendizaje de un espacio de características de manera automática usando

Autoencoders para señales sísmicas del Volcán Cotopaxi

Jenny Belén Pupiales Alulema

Nombre Profesor, Título académico Felipe Grijalva Arévalo, PhD

Quito, 22 de diciembre del 2022

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y Apellidos: Jenny Belén Pupiales Alulema

Código: 00213412

Cédula de identidad: 1725615841

Lugar y Fecha: Quito, 22 de diciembre del 2022

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

RESUMEN

Ecuador forma parte de una de las regiones más volcánicas y sísmicas del mundo, es por esto la gran necesidad de evaluar y predecir actividad anómala que ayude a evitar situaciones catastróficas. Una forma es realizar múltiples evaluaciones a datos recolectados por equipo técnico, sin embargo, mucha de esta información proporcionada no se encuentra etiquetada y realizar este proceso manualmente a una gran cantidad de información resulta ineficiente y costoso. Debido a esto se aplicó un método de aprendizaje no supervisado como es el Autoencoder (red neuronal) para aprender y obtener un nuevo espacio de característica a partir de una base de datos sin etiquetas. Finalmente, los resultados de la eficiencia del nuevo espacio de características entrenadas fueron comparados con otros modelos de clasificación los cuales mostraron una clasificación poco favorable con valores de AUC entre 0.5 y 0.6, posiblemente causado por la fase de preparación de la base de datos, ya que el canal de fase presenta un comportamiento caótico y una alternativa mejorar el proceso de clasificación pueden ser el uso de redes neuronales secuenciales o preentrenadas.

Palabras clave: Aprendizaje automático, no supervisado, Autoencoder, sismos volcánicos, volcán, Cotopaxi VII

ABSTRACT

Ecuador is part of one of the most volcanic and seismic regions in the world, therefore there is a great need to evaluate and predict anomalous activity that helps prevent catastrophic situations. One way is making multiple evaluations on data collected by professional instruments, but much of this collected information is not labeled and to make this process manually on a huge amount of information is inefficient and expensive. Due to this, unsupervised learning method was applied, in this case we used an Autoencoder (neural network) to learn and obtain a new feature space from a database without labels. Finally, the efficiency results of the new trained feature space were compared with other classification models, which showed an unfavorable classification with AUC values between 0.5 and 0.6, possibly caused by the database preparation, because the phase channel presents a chaotic behavior and an option to become better the classification process, can be the use of sequential or pretrained neural net- works.

Key words: Machine learning, unsupervised, autoencoder, volcanic earthquakes, volcano, Cotopaxi

TABLA DE CONTENIDO

INTRODUCCIÓN	11
DESARROLLO DEL TEMA	13
Objetivo General.....	13
Objetivos Específicos	13
Alcance	14
Marco teórico.....	15
Sismicidad volcánica.....	15
Volcán Cotopaxi.....	16
Redes Neuronales	17
Aprendizaje Supervisado vs no supervisado	18
Autoencoders.....	19
Transformada de Fourier de tiempo corto	20
Metodología	21
Preparación del Dataset	21
Diseño de la fase no supervisada (Autoencoder)	22
Evaluación del Autoencoder entrenado.....	23
Diseño de la fase supervisada.....	23

Configuración Experimental	26
Resultados y discusión	27
CONCLUSIONES Y RECOMENDACIONES	36
REFERENCIAS BIBLIOGRÁFICAS.....	37
ANEXOS.....	41
Anexo A. Reconstrucción de espectrogramas de varios tipos de señales.	41
Anexo B. Métricas del proceso de clasificación en Python para el Volcán Cotopaxi. 42	
Anexo C. Métricas del proceso de clasificación en Python para el Volcán Llama. ...	43

ÍNDICE DE TABLAS

Tabla 1. Diferencias entre el aprendizaje supervisado y no supervisado en Machine Learning	
.....	19

ÍNDICE DE FIGURAS

Figura 1. Metodología del proyecto.	14
Figura 2. Ubicación geográfica de las estaciones sísmicas del volcán Cotopaxi.....	17
Figura 3. Esquema de funcionamiento de un Autoencoder convolucional.....	20
Figura 4. Ejemplo del comportamiento de las señales de la BD1. } (a) Señal sísmica sin preprocesamiento. (b) Señal sísmica luego de aplicar sftp.	28
Figura 5. Arquitectura Encoder.....	29
Figura 6. Arquitectura Decoder.....	30
Figura 7. Plot Loss vs epoch del entrenamiento del Autoencoder.	31
Figura 8. Reconstrucción de espectrograma de señales sísmicas.....	32
Figura 9. Matrices de confusión resultantes en Matlab volcán Cotopaxi, (b) volcán Llaima.	33
Figura 10. Curvas ROC resultantes en Matlab } (a) volcán Cotopaxi, (b) volcán Llaima.	33
Figura 11. TSNE resultantes en Matlab (a) volcán Cotopaxi, (b) volcán Llaima.....	34
Figura 12. Resultados de clasificación en Python volcán Cotopaxi. (a) Matriz de confusión volcán Cotopaxi, (b) Gráfico curva de ROC.....	35
Figura 13. Resultados de clasificación en Python volcán Llaima. (a) Matriz de confusión volcán Llaima, (b) Gráfico curva de ROC	35

INTRODUCCIÓN

Las erupciones volcánicas son fenómenos que ocasionan un gran impacto ambiental, económico y social por el alto nivel de desastre que ocasionan, especialmente cuando estas elevaciones están ubicadas cerca de zonas pobladas. Ecuador es un país con un gran número de volcanes activos, debido a esto se realizan constantes monitoreos por parte del Instituto Geofísico de la Escuela Politécnica Nacional, para evaluar el comportamiento de estas estructuras, especialmente el comportamiento sísmico del volcán Cotopaxi, considerado como uno de los más peligrosos a nivel mundial. El Cotopaxi se encuentra ubicado a en la cordillera de los andes a 35 km de la ciudad de Latacunga, provincia de Cotopaxi y a 45 km de la ciudad de Quito capital del Ecuador (*COTOPAXI - Instituto Geofísico - EPN, s. f.*), sus señales sísmicas son monitoreadas mediante con manómetros que detectan señales sísmicas de largo periodo, señales tectónicas, temblores volcánicos, explosiones, terremotos, temblores espasmódicos, etc. (Cortés Moreno, 2016) Sin embargo, la clasificación de los datos de las señales obtenidas, hoy en día son realizadas de forma manual lo cual requiere la intervención de varios recursos humanos especialistas en el tema para etiquetar de datos. Algunos autores han realizado procesos de clasificación sísmica utilizando modelos de aprendizaje supervisado o semi supervisado, sin embargo, este método puede presentar algunas desventajas, como la ineficacia del entrenamiento del modelo para la evaluación de datos de diferentes fuentes. El resultado de este tipo de modelos supervisados varía mucho dependiendo de los datos de entrada, ya que forzosamente se requiere del uso de una base de datos etiquetada. Por otro lado, los modelos de aprendizaje no supervisado como los Autoencoders no requieren de una base de datos etiquetada y pueden revelar estructuras ocultas que ayuden al mejor entendimiento de

los datos procesados (Sathya, Abraham, 2013). Esto ayudará a entender de mejor manera la dinámica de un volcán, para realizar una clasificación más eficaz a diferencia de la clasificación manual que se realiza actualmente.

DESARROLLO DEL TEMA

Objetivo General

Implementar el aprendizaje de un espacio de características de forma automática para señales sísmicas del Volcán Cotopaxi utilizando Autoencoders.

Objetivos Específicos

- Estudiar los conceptos básicos de sismicidad volcánica y aprendizaje automático.
- Investigar sobre la aplicación de Autoencoders en aprendizaje no supervisado para la construcción de un nuevo espacio de características.
- Codificar el método de aprendizaje no supervisado con Autoencoders.
- Codificar el método de clasificación mediante aprendizaje supervisado.
- Evaluar cuantitativamente la eficiencia de clasificación de datos sísmicos sobre el nuevo espacio de características.

Alcance

Este documento muestra el flujo del desarrollo de este proyecto para el diseño y entrenamiento de una red neuronal, en este caso se utiliza Autoencoders para el entrenamiento de la red neuronal de forma no supervisada, una vez obtenido el modelo entrenado se procede al diseño de una nueva red neuronal supervisada que permita la clasificación de datos etiquetados y en base a esto medir la efectividad de la clasificación del modelo, este proceso se lo puede observar en la figura 1.



Figura 1. Metodología del proyecto.

Marco teórico

Sismicidad volcánica

Existe una estrecha relación entre los sismos y la actividad volcánica y es que al momento en que se producen movimientos sísmicos las placas tectónicas llegan a converger y fracturarse formando diversas grietas, las cuales se convierten en salidas eficientes para el magma, el cual está compuesto por roca fundida, compuestos volátiles y sólidos. Al momento en que el magma desciende puede provocar varios tipos de procesos. [7]

- Aumento de presión en la roca encajante
- Eventos sísmicos
- Deformación de la superficie terrestre
- Cambios fuertes de temperatura
- Emisión de gases magmáticos
- Cambios en el patrón de circulación de los fluidos subterráneos
- Cambios en los campos eléctrico y magnético a escala local

La sismicidad volcánica está relacionada generalmente con los rompimientos de rocas. Por tanto, este tipo de sismos permite obtener diversa información acerca de los volcanes al momento sus erupciones, por otro lado, se tiene en cuenta que ningún volcán es similar a otro, es decir las características de sismicidad volcánica son propias de cada uno, por tanto, al pretender obtener parámetros de modelos estándar para diagnosticar otro volcán, será una actividad complicada de establecer. A pesar de esto conforme el paso del tiempo se consiguió obtener información útil gracias a la detección y registro de las señales sísmicas, se logró identificar, clasificar y analizar cuantitativamente los datos los cuales posteriormente se clasificaron como parámetros estándares todo gracias al estudio continuo de las actividades eruptivas.

Volcán Cotopaxi

El aumento de zonas urbanas y rurales, así como la fructificación industrial tanto pública como privada en Latacunga, provincia de Cotopaxi, presenta considerable vulnerabilidad asociada con la actividad del volcán Cotopaxi. Al Cotopaxi se lo cataloga como uno de los volcanes más peligrosos a nivel mundial por diversos factores, los más importantes son: frecuentes erupciones y estilo eruptivo.

Se localiza a 35 km al Noreste de Latacunga y 45 km al Sureste de Quito, tiene una altura de 5897 metros sobre el nivel del mar, con un diámetro basal de aproximadamente 20 km, es un volcán de tipo estrato volcán compuesto, es decir es cónico y de altura considerable (*COTOPAXI - Instituto Geofísico - EPN, s. f.*)

Sus periodos eruptivos más significantes son 1532-1534, 1742-1744, 1766-1768, 1853-1854 y 1877-1880. La última erupción fue registrada en el 2015. El volcán Cotopaxi está en constante monitoreo, tanto visual como instrumental. El monitoreo visual consiste en la observación sistemática permanente la cual no requiere de instrumentación. Por otro lado, el monitoreo instrumental se basa en el uso de equipos especiales los cuales están diseñados para tener una alta precisión y comunicación robusta la cual permite obtener datos a tiempo real. (Zapata Yugsi, 2022).

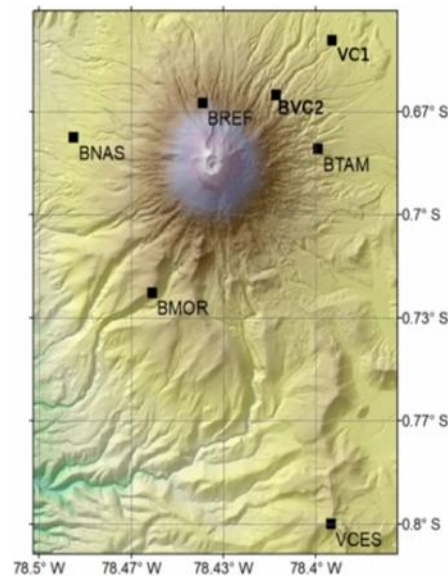


Figura 2. Ubicación geográfica de las estaciones sísmicas del volcán Cotopaxi

Redes Neuronales

Se basa en la extracción del funcionamiento de la red neuronal de un ser humano, comenzando por un modelo básico, mientras con el paso del tiempo se la va estructurando con diversas conexiones mediante varios nodos los cuales representan la función específica de salida o función de activación. La conexión entre 2 nodos representa un peso lo cual equivale a la memoria de la red neuronal y reflejan la fuerza de conexión entre células. (Shahid, Rappon, & Berta, 2019)

Por lo general la red suele representar algún tipo de algoritmo o puede llegar a ser una expresión de estrategia lógica. Por otro lado, una unidad de procesamiento de neuronas puede dividirse en 3 categorías de unidades: de entrada, de salida y ocultas y las mismas representan varios tipos de objetos que tengan patrones significativos de abstracción. (DeVore, Hanin, & Petrova, 2021)

La red neuronal artificial es un procesamiento de información adaptativo del cerebro sin software, y su esencia es transformar la red y mantener la dinámica de distribución paralela de funciones de imitación y procesamiento de información en diferentes niveles. La red neuronal también investiga muchos campos de la neurociencia,

la ciencia de las ideas, la inteligencia artificial, informática y otros campos interdisciplinarios.

Aprendizaje Supervisado vs no supervisado

El aprendizaje supervisado y el no supervisado son dos modelos distintos de aprendizaje automatizado. Sin embargo, para la aplicación de uno u otro debemos tomar en cuenta los datos a disposición, además del trabajo que se requiere. En el artículo Aprendizaje supervisado y no supervisado de la Universidad Europea, nos explican la importancia del aprendizaje automatizado. UE indica que: “El aprendizaje automático ya forma una parte importante dentro de las organizaciones y los servicios modernos hoy en día. Ya sea en las redes sociales, la atención médica o en el sector financiero, los modelos de aprendizaje automático se utilizan de diversas formas” . (*Aprendizaje supervisado y no supervisado / Blog UE*)

Se define aprendizaje supervisado porque este necesita ser inspeccionado por personas. Esto se debe a que los datos que tenemos no tienen etiqueta, por lo que necesita acción humana que rotule con precisión. Obviamente, esto requiere de un proceso largo para que el humano logre etiquetar un gran volumen de datos. Este modelo datos de entrada y otros etiquetados en la salida, con la finalidad de las combine y pueda predecir. Sharma en uno de sus escritos sobre el tema dice: “La técnica es comúnmente utilizada para deducir una función a partir de los datos de entrenamiento, que consisten en partes de objetos dentro de los cuales uno de los componentes son los datos de entrada y el componente restante son los resultados deseados”. (Sharma, 2020)

El aprendizaje no supervisado, por otro lado, no necesita intervención humana. Por lo tanto, no se requiere etiquetar, ni procesamiento previo de datos. Aunque, necesita que el usuario entregue ciertos parámetros necesarios para el modelo, pero después es capaz de procesar grandes volúmenes de datos de forma eficaz y de manera autónoma.

Sharma expone que: “Si no se cuenta con un conjunto de entrenamiento, es decir, no existe comprensión acerca de las etiquetas correspondientes a los patrones, se pueden entonces clasificar objetos a través de un proceso previo de análisis de los datos conocido como clasificación no supervisada, aprendizaje no supervisado, técnicas de agrupamiento o clustering, que permite conocer la estructura de los datos”. (Sharma, 2020)

Aprendizaje supervisado	Aprendizaje no supervisado
Datos con etiqueta.	Datos sin etiqueta.
Suele usarse para clasificar o hacer predicciones.	Sirve para encontrar las relaciones en un grupo de datos.
Utiliza más recursos computacionales.	Necesita menos recursos computacionales.

Tabla 1. Diferencias entre el aprendizaje supervisado y no supervisado en Machine Learning.

Autoencoders

Generalmente, las redes neuronales son usadas en entornos supervisados, es decir, que para cada etiquetado que se realice, el modelo aprende la relación entre la entrada y las etiquetas. Umberto Michelucci, en su paper *An Introduction to Autoencoders* nos indica que: “Un Autoencoder es un tipo de algoritmo con el principal propósito de aprender una representación "informativa" de los datos que se puede usar para diferentes aplicaciones aprendiendo a reconstruir un conjunto de observaciones de entrada lo suficientemente bien”. (Michelucci, 2022)

Los Autoencoders tienen una organización y arquitectura base. Los componentes básicos son los siguientes:

- Codificador: función Encoder (redes neuronales), se puede representar como como una función matemática que dependa de x como:

$$h_i = g(x_i) \tag{1}$$

- Latent feature representation: tensor de números reales es:

$$h_i \in R^q \quad (2)$$

- Decodificador: función decoder (redes neuronales) es:

$$\tilde{X}_i = f(h_i) = f(g(x_i)) \quad (3)$$

Donde $\tilde{X}_i \in R^n$, lo que significa que entrenar al Encoder es encontrar g y f , que satisfagan

$$\arg \min_{f, g} [\Delta(X_i, f(g(X_i)))] \quad (4)$$

A continuación, en la figura 3, podemos ver de forma gráfica el flujo y la estructura que sigue un Autoencoder.

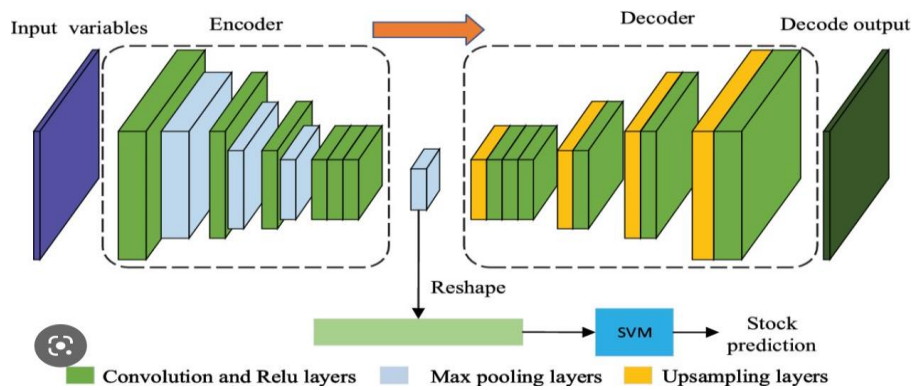


Figura 3. Esquema de funcionamiento de un Autoencoder convolucional.

Transformada de Fourier de tiempo corto

Es un tipo de Transformada de Fourier que tiene como objetivo determinar la frecuencia sinusoidal y el contenido de fase en cada sección de una señal que varía con el tiempo. El cálculo de este proceso se basa en segmentar una señal en segmentos más pequeñas para calcular la transformada de Fourier en cada segmento por separado y así generar un espectro de Fourier para cada uno y generar una representación bidimensional de la señal.

A continuación, se muestra la fórmula matemática que describe este proceso de transformación de la señal. (Cerón, 2019)

$$STFTX(T)(\omega, \tau) \equiv X(\omega, \tau) = \int_{-\infty}^{\infty} x(t)\omega(t - \tau)e^{i\omega t} dt \quad (5)$$

Metodología

Este proyecto plantea la clasificación de datos sísmicos del volcán Cotopaxi en 2 etapas elementales. La primera etapa utiliza un conjunto de datos no etiquetado, el cual será utilizado como entrada para el entrenamiento de un modelo de aprendizaje no supervisado llamado Autoencoder. La ventaja del uso de este modelo está en su estructura de 2 capas, donde la primera capa realiza codificación (reducción de la dimensionalidad de datos) y la segunda un proceso de decodificación (expansión al tamaño original de la entrada) (Goikoetxea Pardo, 2018) (Paco. M, Paco. V, Loaiza, 2019).

Posteriormente una vez entrenado el modelo, se procederá a la etapa de aprendizaje supervisado. Aquí se utilizará un conjunto de datos más liviano y etiquetado, para evaluar la eficacia del nuevo espacio de características usando un clasificador mediante la métrica de exactitud (Ríos Carrillo, 2022).

Preparación del Dataset

Para este proyecto se utilizó una base de datos del Instituto Geofísico de la Escuela Politécnica Nacional, en la primera etapa de análisis se consideró un Dataset compuesto por señales sísmicas sin etiqueta, denominado BD1, las cuales fueron obtenidas de diferentes estaciones como: BNAS, BREF, BTAM y BVC2 del volcán Cotopaxi como se observa en la figura 2. Estas estaciones registran diferentes tipos de señales sísmicas volcánicas como LP, VT, REG, TR, HYB, EXPL, ICE, TRESP, VLP y TRBA. Cada una de estas señales están representadas por un vector numérico los cuales fueron utilizado para el entrenamiento del Autoencoder, es importante tomar en cuenta que la gran cantidad de volumen de datos en esta primera etapa permite que el proceso de entrenamiento sea más eficiente, esta etapa fue desarrollada en base a la norma ISO/IEC TR 24028:2020 Data pre-processing and modelling. (ISO/IEC TR 24028:2020)

En la segunda etapa de entrenamiento con el modelo supervisado se utilizó un Dataset etiquetado de menor volumen, denominado BD2, utilizando únicamente las señales sísmicas LP y VT, las etiquetas de estos datos fueron verificados por los especialistas del Instituto Geofísico.

Para el inicio del entrenamiento neuronal no supervisado, se realizó un preprocesamiento de la BD1 utilizando Matlab, la longitud de cada vector es reajustada a un tamaño fijo $(8192 \times 1)(w \times h)$, para la normalización de datos. Luego de preparada la BD1, se realizó un análisis de frecuencia a las señales, debido a que cada vector contiene datos no estacionarios en el tiempo, los cuales no permiten mostrar una clara interpretación de la información de los eventos sísmicos, el proceso de filtrado se realizó utilizando la función Transformada de Fourier de tiempo corto, concepto detallado en la sección 5.6.

Una vez preparada la BD1 se procedió con el cálculo de la magnitud y fase de cada señal para generar los datos numéricos necesarios para el Autoencoder. (Badescu, 2017)

Diseño de la fase no supervisada (Autoencoder)

Esta etapa, utiliza la BD1 preprocesada con un tamaño de 21201 señales, la arquitectura del modelo implementado es un Autoencoder convolucional, el diseño del Encoder y Decoder utilizado está conformado por capas ocultas y filtros de convolución los cuales realizan el cálculo del valor de un píxel en base a la función de ponderación de sus vecinos, el número de filtros estará definido como primer parámetro dentro de la función Conv2D, luego se definen las dimensiones por las cuales se va a aumentar o disminuir el volumen de datos (Cheng, Sun, Takeuchi, & Katto, 2018), el procesamiento de esta etapa se desarrolló en base a las normas ISO/IEC 22989:2022 Machine Learning Concepts (ISO/IEC TR 24372:2021) y ISO/IEC 24028:2022 Modelling. (ISO/IEC TR

24028:2020)

El modelo del Autoencoder realiza una evaluación no supervisada donde se toma como métrica principal de evaluación de eficiencia el valor de pérdida logarítmica durante la fase de entrenamiento, esta métrica aumenta a medida que la probabilidad predicha se aleja de la etiqueta real, es por esto que en el aprendizaje automático se busca minimizar el valor de pérdida. (Rim, Millán, Planes, Bringa, & Moyano, 2020)

Evaluación del Autoencoder entrenado

Para la evaluación cualitativa del modelo entrenado, se realizó un proceso de reconstrucción de imágenes 2D luego de pasar por el modelo entrenado. Este proceso se denomina procesamiento digital de imágenes, donde cada imagen se define como una función bidimensional $f(x_1, x_2)$, donde x representa las coordenadas espaciales y f la intensidad de la imagen, este proceso de conversión de una imagen de analógica a digital se realiza mediante un muestreo digitalizando las coordenadas y la intensidad de la matriz de números-imagen ($M \times N$). (Lundh, 2001)

Diseño de la fase supervisada

Esta etapa utiliza la BD2 más pequeña con datos etiquetados, donde se utilizarán distintos tipos de señales sísmicas. Este análisis se realizó mediante la aplicación de distintos modelos de clasificación como:

- **Modelo de análisis de Discriminante Lineal optimizable**

Utiliza cálculos estadísticos basados en el teorema de Bayes, para calcular la probabilidad de que una observación pertenezca a un conjunto de datos o variable cualitativa dado un valor de predicción determinada, este proceso se describe matemáticamente por la siguiente ecuación. (Cortez, 2021)

$$\log(p(x|\omega_j)p(\omega_j)) > \log(p(x|\omega_k)p(\omega_k)) \quad (6)$$

Donde $k = 1, \dots, C; k \neq j$

- **Modelo de Árbol de decisión optimizable**

Este tipo de aprendizaje automático mediante árbol de decisión pertenece a un procedimiento de aproximaciones de valores discretos de una función. Por lo que, la función a la se referencia, es en este caso, representada como un árbol de decisión. (C.-F. YU & 余致鋒, 2018) Tenemos una distribución probabilística:

$$P = \{p_1, p_2, \dots, p_n\} \quad (7)$$

Entendemos como entropía de la distribución:

$$H_b(p_1, p_2, \dots, p_n) = \sum_{i=1}^{i=n} p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p} \quad (8)$$

- **Modelo de Naive Bayes optimizable**

Se puede definir como un procedimiento de probabilidades que tiene como origen el teorema de Bayes. En pocas palabras, se dice que se puede aprender cuando realizamos aproximaciones, es decir, cada aproximación nos acercamos a la verdad, a medida que obtenemos más información. (Wester, 2021)

$$P(a|b) = \frac{p(a|b)p(a)}{p(b)} \quad (9)$$

- **Modelo de SVM optimizable**

Es modelo de aprendizaje de tipo supervisado, este se lo utiliza generalmente para soluciones que requiera regresión y clasificación.

Generalmente se usa la función de base radial o Gaussiana. (Łakomska, 2014)

$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right) \quad (10)$$

- **Modelo de KNN optimizable**

Es un tipo de clasificador, con aprendizaje supervisado no paramétrico. En

este caso, usa la proximidad que tiene k para poder realizar las clasificaciones y predicciones. Para poder usar este tipo de clasificador se parte de la premisa de que es posible encontrar otros puntos similares, que se encuentren cerca unos de otros. (Sateesh Kumar. Dr.S.Sameen Fatima, 2020) Para esto, se usan métricas de distancia, en este caso la más usada es la euclidiana:

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (11)$$

- **Modelo de conjunto optimizable**

Es un tipo de clasificador de tipo predictivo que está hecho por una mixtura de otros varios modelos de clasificación, esto hace que la capacidad de predecir aumente. (Abello, 2019)

- **Modelo de red neuronal optimizable**

Son modelos de redes neuronales, que simulan la forma en la que trabaja el cerebro al momento de procesar información. Estas son redes neuronales predictivas. Estas pueden ser reajustadas a las capas interconectadas, y además, cambiar ciertas funciones de activación. (Wang, 2020) Encontramos a la función de propagación:

$$p_j(t) = \sum_i \sigma_i(t) w_{ij} \quad (12)$$

Por otro lado, también se realizó la construcción de una red neuronal, tradicional para clasificación de multiclases con validación cruzada en referencia a la norma ISO/IEC 24028:2022, lo que permitió realizar una división del set de datos en k subconjuntos donde los subconjuntos $k-1$ se utilizaron para entrenamiento y el restante para validación, repitiendo este proceso k veces para realizar una evaluación eficiente del modelo. Los

datos evaluados en este modelo fueron para señales sísmicas del volcán Cotopaxi y señales sísmicas del volcán Llaima en Chile.

Configuración Experimental

Para la preparación de la BD1 y la BD2, se registraron 21202 señales sísmicas de diferentes tipos, estas señales fueron cargadas al software MATLAB & Simulink-MathWorks. Cada vector que representa la señal sísmica fue redimensionado a un tamaño específico 1×8292 , para luego procesar las señales bajo la función $\text{sft}()$ (Transformada de Fourier a tiempo corto), con la transformada de Fourier de cada señal se realizó el cálculo de la magnitud $M = 20 \log(|s|)$ y fase $F = \text{unwrap}(\text{angle}(s))$ de cada señal para construir la imagen ($M \times F \times 2$) como entrada para el proceso de entrenamiento con el Autoencoder.

Durante el proceso de entrenamiento del Autoencoder se utilizó la arquitectura mostrada en las figuras 5 y 6, este modelo está compuesto por 4 capas convolucionales, cada capa utiliza el método $\text{Conv2D}(M, N)$ el cual crea una matriz de convolución de kernel de tamaño ($M \times N$) para generar el número de filtros de convolución de cada capa para el posterior entrenamiento, de la misma forma se aplicó un subsampling para cada capa de convolución $\text{MaxPooling2D}()$ para la fase del Encoder y $\text{UpSampling2D}()$ para la fase del Decoder, lo cual permite modificar el tamaño de la entrada para la siguiente etapa de convolución y se utilizaron las funciones de activación LeakyReLU, en base a la experimentación recomendadas por la literatura en base a la norma ISO/IEC 24028:2022 Modelling. (ISO/IEC TR 24028:2020)

El proceso de entrenamiento del Autoencoder se realizó bajo las siguientes condiciones:

- Implementación de un checkpoint para guardar los mejores pesos durante el entrenamiento del modelo, tomando como punto de monitoreo la métrica

val_loss.

- Optimizador de tipo Adam.
- Valor de pérdida calculado en base al error logarítmico medio cuadrático.
- 100 épocas de entrenamiento.

Para la fase de Clasificación se evaluaron las predicciones de la BD2 generadas con el modelo Encoder entrenado a partir de Autoencoder, para esto se utilizó la herramienta Classification Learner de MATLAB & Simulink-MathWorks, las predicciones se evaluaron con los modelos de clasificación descritos en la sección de diseño de la fase supervisada.

De la misma forma se realizó el diseño de un Clasificador en Python, la arquitectura de este clasificador está compuesta por la primera fase del Encoder entrenado, mas 2 capas Densas para pasar por un proceso de entrenamiento, el cual se realiza bajo las siguientes condiciones:

- Implementación de un callback Checkpoint para guardar los mejores pesos durante el entrenamiento del modelo, tomando como punto de monitoreo la métrica val_loss.
- Implementación de un callback EarlyStop para terminar el proceso de entrenamiento cuando la métrica monitoreada (val_categorical_accuracy) deje de mejorar y empiece a estabilizarse.
- Optimizador de tipo Adam.
- Valor de pérdida calculado en base al error logarítmico medio cuadrático.

Cada plot de evaluación obtenido fue estructurado en base a la norma ISO/IEC TR 24028:2020 Testing and evaluation. (ISO/IEC TR 24028:2020)

Resultados y discusión

En la primera etapa el preprocesamiento de la base de datos permitió representar las señales en escalas de tiempo frecuencia (*Short-time Fourier transform - MATLAB stft*, MathWorks) para obtener señales más comprensibles, como se observa en la figura 4 (b), para cada señal se realizó el cálculo de la magnitud y fase para representar cada dato como un vector de imágenes con 2 canales ($W \times H \times C$).

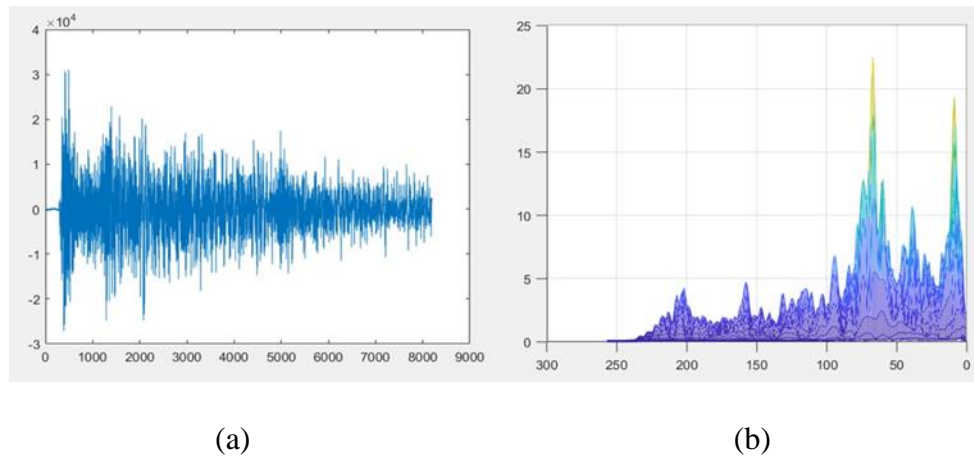


Figura 4. Ejemplo del comportamiento de las señales de la BDI.} (a) Señal sísmica sin preprocesamiento. (b) Señal sísmica luego de aplicar stft.

Esto permitió que se pueda guardar el vector de datos de cada señal en distintos archivos tipo Numpy para entrenar el Autoencoder.

Para la fase no supervisada se realizó el diseño de la arquitectura del Autoencoder, en la siguiente figura se muestra cada fase de la red neuronal, el Encoder donde el tamaño de la entrada definida es de $257 \times 221 \times 2$, el cuello de botella donde se definió la capa comprimida de la entrada con un tamaño de 100 y finalmente el Decoder donde se definieron varias capas para reconstruir los datos y generar la salida del mismo tamaño de la entrada $257 \times 221 \times 2$.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 257, 221, 32)	608
batch_normalization (Batch Normalization)	(None, 257, 221, 32)	128
leaky_re_lu (LeakyReLU)	(None, 257, 221, 32)	0
max_pooling2d (MaxPooling2D)	(None, 129, 111, 32)	0
conv2d_1 (Conv2D)	(None, 129, 111, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 129, 111, 32)	128
leaky_re_lu_1 (LeakyReLU)	(None, 129, 111, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 65, 56, 32)	0
conv2d_2 (Conv2D)	(None, 65, 56, 32)	9248
batch_normalization_2 (Batch Normalization)	(None, 65, 56, 32)	128
leaky_re_lu_2 (LeakyReLU)	(None, 65, 56, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 33, 28, 32)	0
conv2d_3 (Conv2D)	(None, 33, 28, 1)	289
batch_normalization_3 (Batch Normalization)	(None, 33, 28, 1)	4
leaky_re_lu_3 (LeakyReLU)	(None, 33, 28, 1)	0
max_pooling2d_3 (MaxPooling2D)	(None, 17, 14, 1)	0
flatten (Flatten)	(None, 238)	0
dense (Dense)	(None, 100)	23900

Figura 5. Arquitectura Encoder

dense_1 (Dense)	(None, 100)	10100
reshape (Reshape)	(None, 25, 4, 1)	0
conv2d_4 (Conv2D)	(None, 25, 4, 1)	10
batch_normalization_4 (Batch Normalization)	(None, 25, 4, 1)	4
leaky_re_lu_6 (LeakyReLU)	(None, 25, 4, 1)	0
up_sampling2d (UpSampling2D)	(None, 25, 28, 1)	0
conv2d_5 (Conv2D)	(None, 25, 28, 32)	320
batch_normalization_5 (Batch Normalization)	(None, 25, 28, 32)	128
leaky_re_lu_7 (LeakyReLU)	(None, 25, 28, 32)	0
up_sampling2d_1 (UpSampling2D)	(None, 75, 56, 32)	0
conv2d_6 (Conv2D)	(None, 75, 56, 32)	9248
batch_normalization_6 (Batch Normalization)	(None, 75, 56, 32)	128
leaky_re_lu_8 (LeakyReLU)	(None, 75, 56, 32)	0
up_sampling2d_2 (UpSampling2D)	(None, 150, 112, 32)	0
conv2d_7 (Conv2D)	(None, 150, 112, 32)	9248
batch_normalization_7 (Batch Normalization)	(None, 150, 112, 32)	128
leaky_re_lu_9 (LeakyReLU)	(None, 150, 112, 32)	0
up_sampling2d_3 (UpSampling2D)	(None, 300, 224, 32)	0
conv2d_8 (Conv2D)	(None, 300, 224, 2)	578
batch_normalization_8 (Batch Normalization)	(None, 300, 224, 2)	8
leaky_re_lu_10 (LeakyReLU)	(None, 300, 224, 2)	0
cropping2d (Cropping2D)	(None, 257, 221, 2)	0

Total params:	73,581	
Trainable params:	0	
Non-trainable params:	73,581	

Figura # 6. Arquitectura Decoder

En el proceso de entrenamiento del Autoencoder se obtuvo la gráfica de época vs pérdida, la cual se observa en la figura 7. En este plot se puede apreciar que la pérdida cuadrática logarítmica calculada durante el proceso de entrenamiento disminuye significativamente durante las primeras épocas de entrenamiento, a medida que aumenta el número de épocas esta tiende a llegar a una fase estable donde no hay variación en la pérdida de entrenamiento y validación llegando hasta un punto mínimo entre 0.09 y 0.08, sin la presencia de comportamientos anómalos como overfitting.

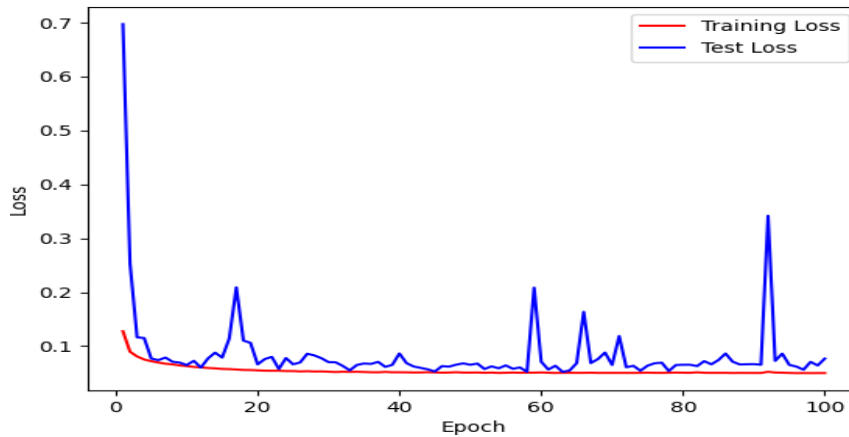


Figura 7. Plot Loss vs epoch del entrenamiento del Autoencoder.

Durante la fase de entrenamiento del Autoencoder los modelos entrenados fueron guardados en diferentes archivos .h5, que posteriormente fueron utilizados para realizar una evaluación cualitativa de la eficacia del modelo entrenado, a continuación, se muestran 3 ejemplos de reconstrucción de distintos tipos de señales, se puede observar en el lado izquierdo la imagen de la señal original y el en lado derecho la imagen reconstruida luego de pasar por el Autoencoder. La figura 8 muestra que las imágenes mantienen una ligera similitud, lo que quiere decir que el modelo entrenado está funcionando favorablemente extrayendo las características más importantes de las señales para luego poder predecir la forma de la imagen aprendida, más ejemplos pueden observarse en el Anexo A.

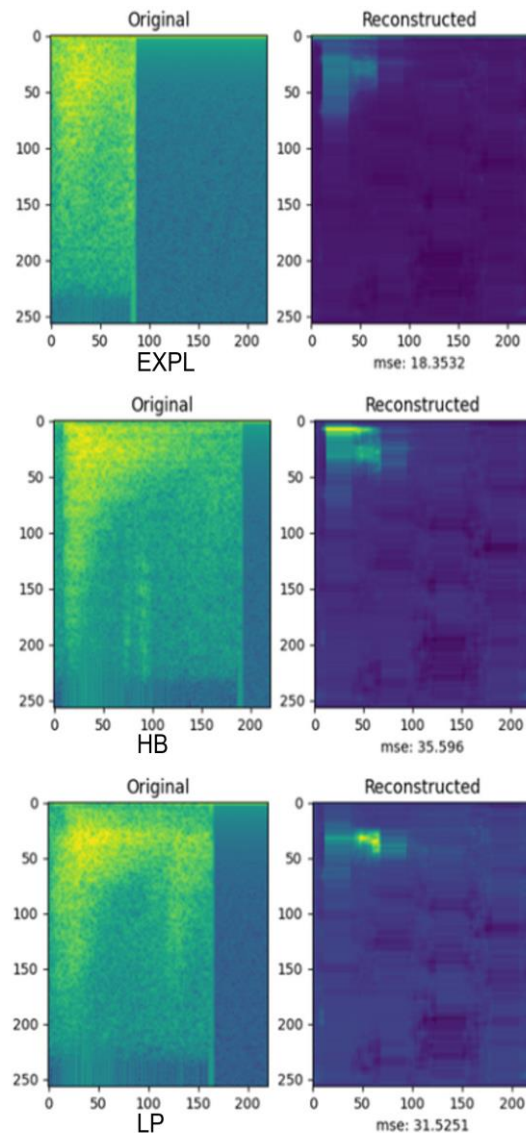


Figura 8. Reconstrucción de espectrograma de señales sísmicas.

Para la fase de clasificación en Matlab el modelo con mejores resultados fue Optimizable Discriminant, tanto para la base de datos del volcán Cotopaxi como para el volcán Llaima, se realizó el cálculo de la matriz de confusión y el plot de la curva ROC con las predicciones obtenidas a partir del Encoder entrenado.

En la figura 9(a) se observa que la clasificación se concentra más únicamente en una sola clase, en el caso de los datos del volcán Cotopaxi la clasificación se centra en la clase LP, mientras que para los datos del volcán Llaima la clasificación se centra en la clase TC como se observa en la figura 9(b).

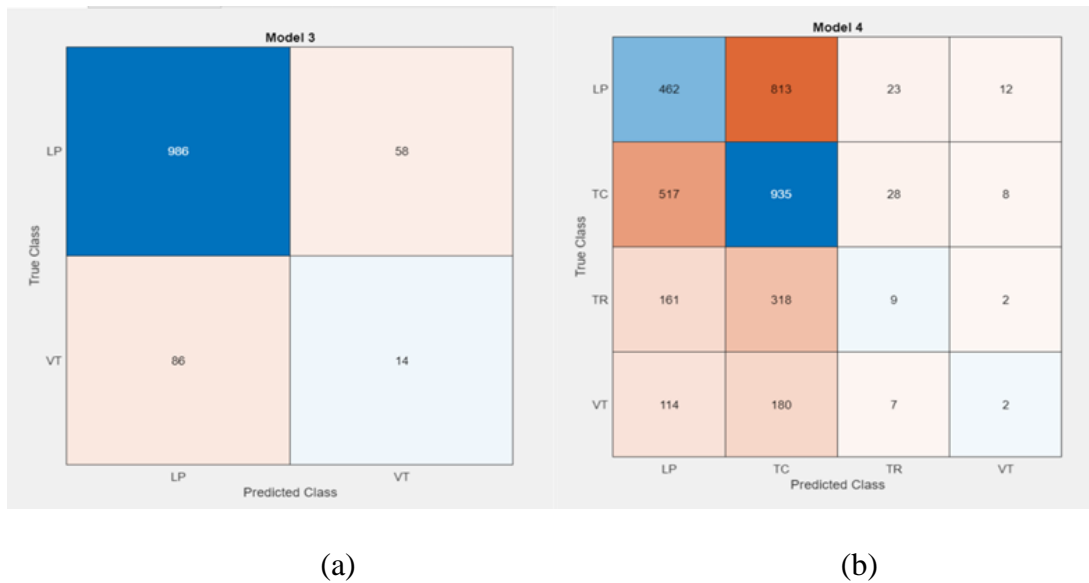


Figura 9. Matrices de confusión resultantes en Matlab volcán Cotopaxi, (b) volcán Llaima.

Las curvas ROC obtenidas para cada análisis reflejan una baja eficiencia al momento de evaluar el modelo, los valores de AUC variaron entre 0.5 y 0.6 como se puede observar en la figura 10, lo cual representa bajos niveles de clasificación de las clases y en consecuencia la clasificación centrada en una única clase como se pudo observar anteriormente en la matriz de confusión.

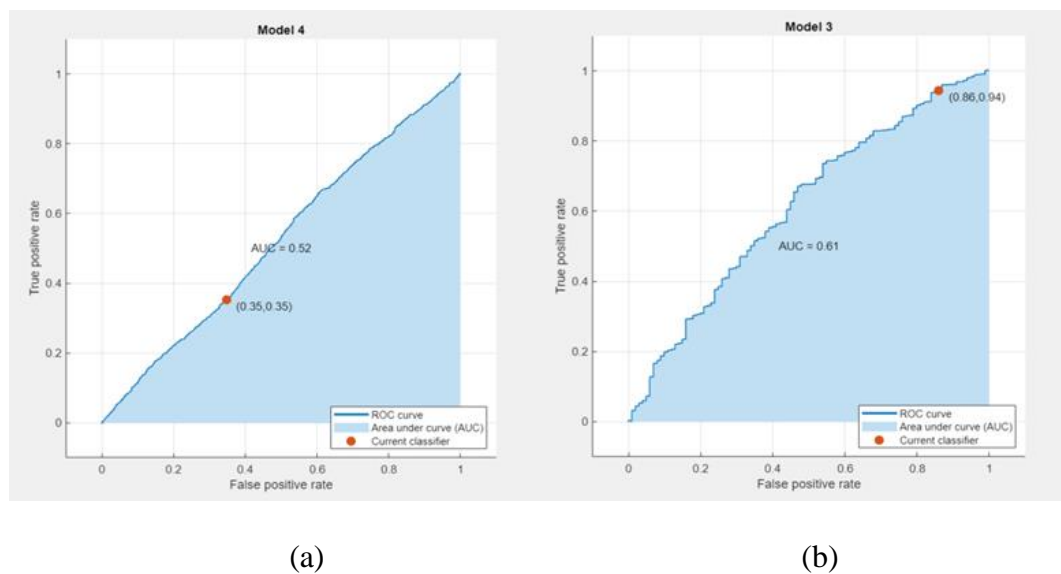


Figura 10. Curvas ROC resultantes en Matlab (a) volcán Cotopaxi, (b) volcán Llaima.

A continuación, se muestran los gráficos TSNE generados en Matlab, con los datos de cada base de datos, estos gráficos muestran que efectivamente el proceso de clasificación no será muy eficiente debido a que la distribución de objetos similares no presenta una separación significativa dentro del espacio de análisis generado como se puede ver en la figura 11.

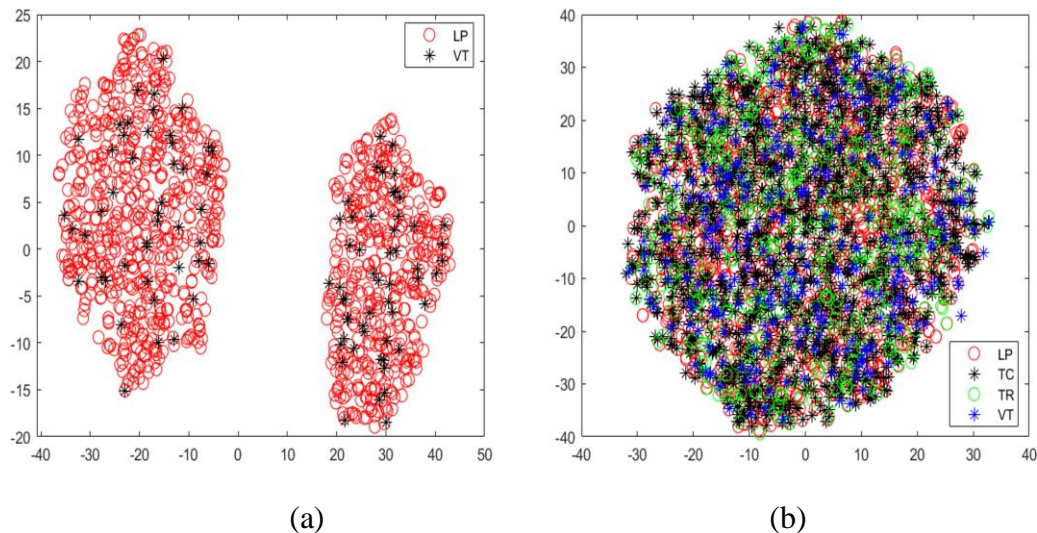


Figura 11. TSNE resultantes en Matlab (a) volcán Cotopaxi, (b) volcán Llaima.

El clasificador diseñado en Python mostró un comportamiento menos favorable en el proceso de clasificación que los modelos probados en Matlab, ya que la matriz de confusión para los dos tipos de base de datos realizan únicamente clasificación para un tipo de clase y estos resultados también se ven reflejados en los valores de AUC representados en la curva ROC mostradas en las figuras 12 y 13. De la misma forma se evaluó la eficiencia del entrenamiento de este modelo con el cálculo de diferentes métricas, la tabla de evaluación de las métricas para cada base de datos se muestra en los Anexos 2 y 3, donde se observan mejores resultados para la base de datos del volcán Cotopaxi debido a que el proceso de entrenamiento del Autoencoder se realizó únicamente con señales de este volcán.

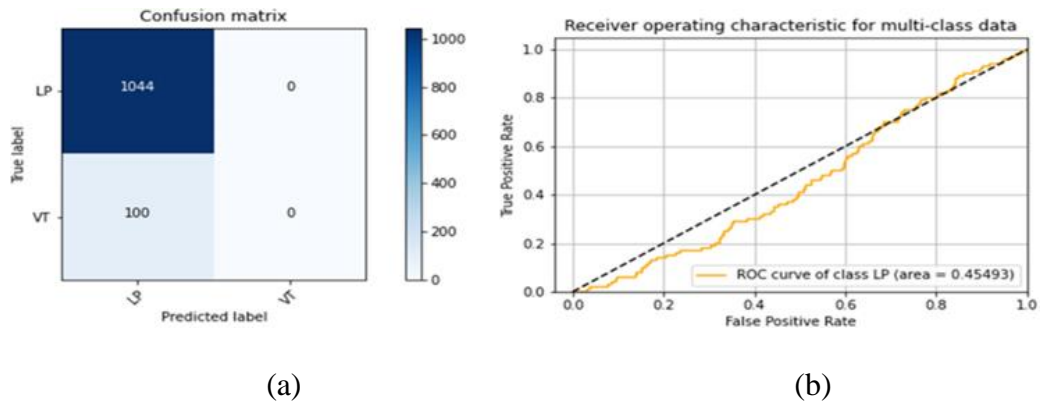


Figura 12. Resultados de clasificación en Python volcán Cotopaxi. (a) Matriz de confusión volcán Cotopaxi, (b) Gráfico curva de ROC

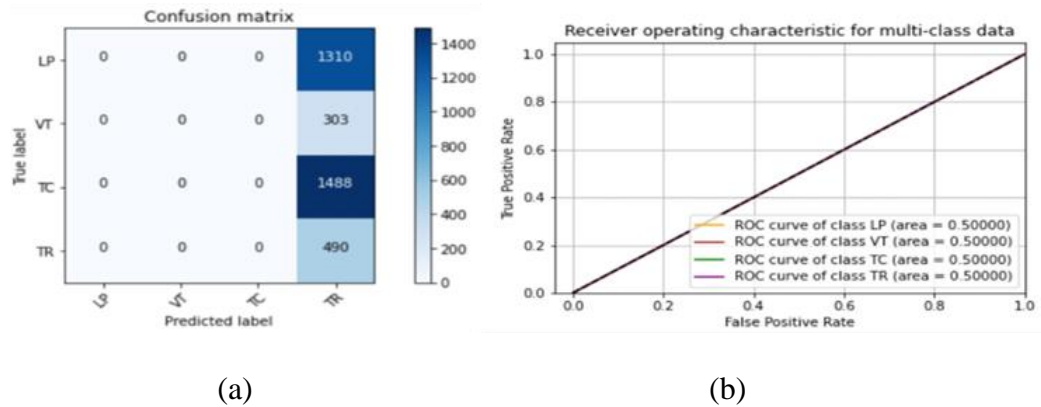


Figura 13. Resultados de clasificación en Python volcán Llaima. (a) Matriz de confusión volcán Llaima, (b) Gráfico curva de ROC

CONCLUSIONES Y RECOMENDACIONES

Se realizó el diseño de un Autoencoder con una arquitectura que consta de 4 capas convolucionales y 1 capa densa de aprendizaje, donde el proceso de entrenamiento se hizo desde cero únicamente con señales del volcán Cotopaxi. El proceso de evaluación del Autoencoder a primera vista se consideró eficiente, ya que presentó un rango de pérdida logarítmica mínima entre 0.08 y 0.1. Sin embargo, durante el proceso de clasificación probado con los diferentes modelos tanto en Matlab como en Python fue poco eficiente, en base al valor de las métricas de AUC obtenidas, a pesar de que se observó que la inclusión de bloques convolucionales dentro de la arquitectura de diseño del Autoencoder ayudó a que el proceso de clasificación mejore.

Se recomienda verificar el preprocesamiento del Dataset ya que la estructura de la fase de cada señal de la base de datos muestra comportamiento caótico, por la inclusión de ruido al momento de normalizar la base de datos para que cada vector tome el mismo tamaño definido y afecta a los modelos diseñados, también se recomienda el uso de redes neuronales recursivas debido a que este tipo de arquitecturas procesa de mejor manera datos en serie de tiempo o el uso de una arquitectura de Autoencoder preentrenada para mejorar el proceso de entrenamiento del Autoencoder y obtener mejores resultados.

REFERENCIAS BIBLIOGRÁFICAS

- COTOPAXI - Instituto Geofísico - EPN.* (s. f.). <https://www.igepn.edu.ec/cotopaxi>
- Ríos Carrillo, D. (2022). *Implementación de un método de agrupación de señales sísmicas generadas por el volcán Cotopaxi basado en aprendizaje automático no supervisado utilizando el modelo de mezcla gaussiana.* Escuela Politécnica Nacional. Quito.
- Goikoetxea Pardo, D. (2018). *Entrenamiento Evolutivo de Autoencoders.* Tesis de Máster. Universidad Politécnica de Madrid. Madrid. https://oa.upm.es/51661/1/TFM_DARIO_GOIKOETXEA_PARDO.pdf
- Sathya, R. Abraham, A. (2013). *Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification.* International Journal of Advanced Research in Artificial Intelligence. vol 2. n.º 2. p 80. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.5274&rep=rep1&type=pdf#page=41>
- Cortés Moreno, G. (2016). *Reconocimiento de señales sismo-volcánicas mediante canales específicos basados en modelos ocultos de Markov.* Universidad de Granada. Granada. <https://digibug.ugr.es/handle/10481/42050>
- Paco, M. Paco, V. Loaiza, A. (2019). *A Feature Extraction Method Based on Convolutional Autoencoder for Plant Leaves Classification.* IEEE Colombian Conference on Applications in Computational Intelligence. p 6. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8781985>
- Cabrera, A. Rosero, E. (2019). *Caracterización de señales volcánicas para la clasificación supervisada de eventos sísmicos.* Universidad de Nariño. Pasto. https://www.diegopeluffo.com/Theses/2019_DegreeThesis_Volcano_classification.pdf

COTOPAXI - Instituto Geofísico. Home - Instituto Geofísico - EPN.

<https://www.igepn.edu.ec/cotopaxi>.

COTOPAXI - Red de Monitoreo - Instituto Geofísico. Home - Instituto Geofísico - EPN.

<https://www.igepn.edu.ec/cotopaxi-red-de-monitoreo>.

Zapata Yugsi, J. (2022). *Generación de señales volcánicas artificiales de tipo lp (long-period) y vt (volcano-tectonic) a partir de una base de datos del volcán cotopaxi usando la técnica de bootstrapping*. Escuela Politécnica Nacional. Quito.

<http://bibdigital.epn.edu.ec/handle/15000/23276>

Shahid, N., Rappon, T., & Berta, W. (2019). *Applications of artificial neural networks in health care organizational decision-making: A scoping review*. PloS one, 14(2),

e0212356. <https://doi.org/10.1007/s11277-017-5224-x>

DeVore, R. Hanin, B. and Petrova, G. (2021). *Neural network approximation*. Acta Numerica, vol. 30, pp. 327–444.

Aprendizaje supervisado y no supervisado | Blog UE. Universidad Europea.

<https://universidadeuropea.com/blog/aprendizaje-supervisado-no-supervisado/>

Sharma, R. (2020). *Study of Supervised Learning and Unsupervised Learning*.

International Journal for Research in Applied Science and Engineering Technology. vol 8, n.º 6. pp 588–593. <https://doi.org/10.22214/ijraset.2020.6095>

Michelucci, U. (2022). *An Introduction to Autoencoders*. Cornell University – arXiv. vol 1. n.º 1. art n.º 1.

Cerón, J. (2019). *La transformada de Fourier de tiempo corto y el impacto de sus hiperparámetros en la calidad de las predicciones*. Universidad de los Andes.

Bogotá. <https://repositorio.uniandes.edu.co/bitstream/handle/1992/45660/u827848.pdf?sequence=1&isAllowed=y#page=30&zoom=100,144,429>

Short-time Fourier transform - MATLAB stft. MathWorks - Makers of MATLAB and

Simulink - MATLAB & Simulink.

<https://www.mathworks.com/help/signal/ref/stft.html>

Badescu, D. (2017). *Speech Enhancement using Deep Learning*, Universidad Politécnicade Cataluña. Cataluña.

<https://upcommons.upc.edu/bitstream/handle/2117/100596/Speech%20Enhancement%20using%20Deep%20Learning.pdf?sequence=1&isAllowed=y>

Schmidt-Hieber, J. (2020). *Nonparametric regression using deep neural networks with ReLU activation function*. *Annals of Statistics*. vol 48. n.º 4. pp 1875–1897.
<https://doi.org/10.1214/19-aos1875>

Cheng, Z. Sun, H. Takeuchi, M. & Katto, J. (2018). *Deep Convolutional AutoEncoder-based Lossy Image Compression*. 2018 Picture Coding Symposium (PCS). pp. 253-257, doi: 10.1109/PCS.2018.8456308.

Bobulski, J. Kubanek, M. (2019). *Waste Classification System Using Image Processing and Convolutional Neural Networks*. *Advances in Computational Intelligence. Lecture Notes in Computer Science*. vol 11507. Springer. Cham.
https://doi.org/10.1007/978-3-030-20518-8_30

Rim, D. N. Millán, E. N. Planes, M. B. Bringa, E. M. y Moyano, L. G. (2020). *Análisis de clústeres para simulaciones de mecánica granular mediante algoritmos de aprendizaje automático*. *Entre ciencia e ingeniería*. Vol, 14. n.º 28. pp 81–86.
<https://doi.org/10.31908/19098367.2058>

Lundh, F. (2001). *Python standard library*. Beijing: O'Reilly.

Haik, Y. (2002). *Engineering Design Process*. Thomson-Engineering.

Cortez, O. (2021). *Modelos de visión por computadora para la determinación de características superficiales y categorías de calidad del Pallar (Phaseolus lunatus L.) Iqueño seco*. UNIVERSIDAD PRIVADA ANTENOR ORREGO.

http://repositorio.upao.edu.pe/bitstream/20.500.12759/7848/1/REP_EVER.CORTEZ_MODELOS.DE.VISION.POR.COMPUTADORA.pdf

C.-F. YU & 余致鋒. (2018). *Application of Decision Tree C5.0 to Fund Decision*, 學位論文. <http://ndltd.ncl.edu.tw/handle/y98nsm>

Wester, P. (2021). *Anomaly-based intrusion detection using Tree Augmented Naive Bayes Classifier*, Student thesis, KTH, Skolan för elektroteknik och datavetenskap (EECS). <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-295754>

Łakomska, J. (2014) *Klasyfikatory SVM*. <https://ruj.uj.edu.pl/xmlui/handle/item/200875>

Sateesh Kumar, R. Dr.S.Sameen Fatima (2020). *Heart Disease Prediction Using Extended KNN(E-KNN)*. International Journal of Advanced Trends in Computer Science and Engineering. vol. 9. n.º 5. pp. 8799–8803. <https://doi.org/10.30534/ijatcse/2020/272952020>

Abello, A. J. A. (2019). *Power Measurement Model Optimization using MATLAB*. International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, n.º 3, pp. 538–543. <https://doi.org/10.30534/ijatcse/2019/31832019>

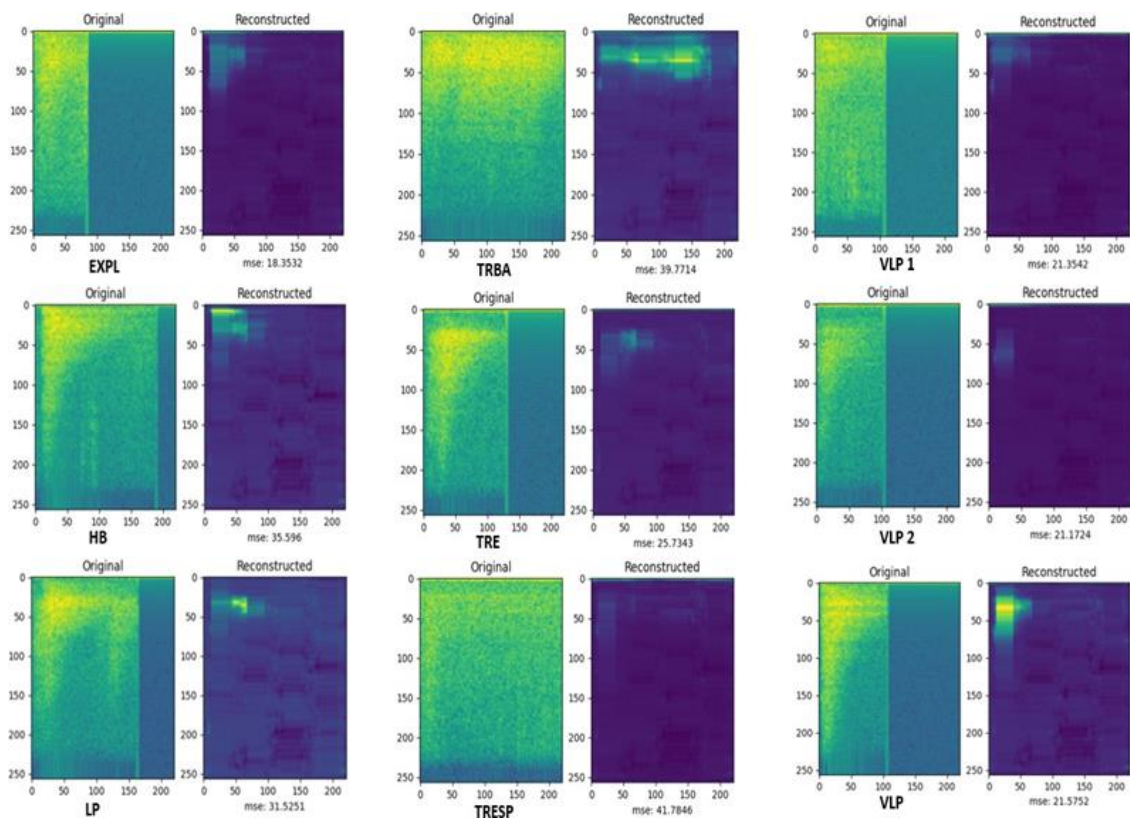
Wang, L. (2020). *Realization of Linear Neural Network Based on MATLAB Neural Network Toolbox*, Computer System Networking and Telecommunications, vol. 3, n.º 1. <https://doi.org/10.18063/csnt.v3i1.1189>

ISO/IEC TR 24028:2020 *Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence*.

ISO/IEC TR 24372:2021 *Information technology — Artificial intelligence (AI) — Overview of computational approaches for AI systems*.

ANEXOS

Anexo A. Reconstrucción de espectrogramas de varios tipos de señales.



Anexo B. Métricas del proceso de clasificación en Python para el Volcán
Cotopaxi.

Fold	Epoch	Random State	AUC mean	AUC std	ACC mean	ACC std	Pre mean	Pre std	Recall mean	Recall std	F1 mean	F1 std	Loss mean	Loss std	Val_Loss mean	Val_Loss std
1	range(1, 10)	1	0,914	0,007	0,907	0,017	0,907	0,017	0,907	0,017	0,907	0,017	0,332	0,048	0,314	0,009
2	range(1, 15)	1	0,918	0,003	0,912	0,001	0,912	0,001	0,912	0,001	0,912	0,001	0,305	0,002	0,302	0,018
3	range(1, 14)	1	0,922	0,002	0,912	0,001	0,912	0,001	0,912	0,001	0,912	0,001	0,300	0,002	0,290	0,020
4	range(1, 9)	1	0,920	0,002	0,912	0,001	0,912	0,001	0,912	0,001	0,912	0,001	0,300	0,003	0,305	0,010
5	range(1, 12)	1	0,916	0,002	0,913	0,001	0,913	0,001	0,913	0,001	0,913	0,001	0,301	0,002	0,303	0,009

Anexo C. Métricas del proceso de clasificación en Python para el Volcán

Llama.

Fold	Epoch	Random State	AUC mean	AUC std	ACC mean	ACC std	Pre mean	Pre std	Recall mean	Recall std	F1 mean	F1 std	Loss mean	Loss std	Val Loss mean	Val Loss std
1	range(1, 13)	1	0,424	0,000	0,137	0,000	0,000	0,000	0,000	0,000	nan	nan	13917285680770800	0,005	13926930745442700	0,062
2	range(1, 9)	1	0,424	0,000	0,136	0,000	0,000	0,000	0,000	0,000	nan	nan	1392308497428890	0,006	13901099920272800	0,032
3	range(1, 10)	1	0,424	0,000	0,136	0,001	0,000	0,000	0,000	0,000	nan	nan	1392925230662020	0,009	13906481213039800	0,053
4	range(1, 19)	1	0,424	0,000	0,136	0,000	0,000	0,000	0,000	0,000	nan	nan	13920046965281100	0,007	13919397089216400	0,048
5	range(1, 11)	1	0,424	0,000	0,136	0,000	0,000	0,000	0,000	0,000	nan	nan	13921593189239500	0,007	1394565725326530	0,061