UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

LOCALIZACIÓN DE EPICENTROS CON DEEP LEARNING

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

ANDRÉS ANTONIO PINTO SALINAS

PROFESOR GUÍA:
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISIÓN:
JORGE WUTH SEPÚLVEDA
FRANCISCO RIVERA SERRANO

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: **ANDRÉS ANTONIO PINTO SALINAS**
FECHA: 2022
PROF. GUÍA: NÉSTOR BECERRA YOMA

LOCALIZACIÓN DE EPICENTROS CON DEEP LEARNING

La localización de epicentros es un problema que ha sido resuelto de distintas maneras y se han ido refinando a lo largo de los años pero, en particular, la localización de epicentros de origen volcánico tiene un trato especial debido a la compleja estructura geológica interna de los volcanes como a lo débiles que son los sismos. Esto desencadena en no tener una solución rápida y efectiva a este problema. Por estos motivos los métodos de localización en vulcanología son costosos en tiempo y personal.

El presente trabajo tiene como objetivo el diseño e implementación de un sistema basado en *Deep Learning* que sea capaz de estimar epicentros provenientes del volcán Chillán, utilizando tan solo la información contenida en las ondas sísmicas captadas por 3 sensores ubicados en las cercanías del volcán.

Con este sistema se logró estimar con error menor a un kilómetro un 51,51 % de los eventos probados, llegando a un error promedio de 1,42 kilómetros.

En dedicatoria a mi padre y mi madre:
Les agradeceré durante toda mi vida lo que me han otorgado.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación y Antecedentes

Desde el origen de la humanidad los eventos sísmicos han formado parte de su vida. Dependiendo de su magnitud y de la zona en que estos se originen pueden llegar a ocasionar catástrofes. Se han logrado identificar distintas fallas en la corteza terrestre lo que permite determinar sectores potenciales donde ocurrirán los sismos. Estas zonas pueden cubrir grandes extensiones de terreno como, por ejemplo, las fallas de San Ramon, Atacama y Pichilemu donde pueden llegar hasta los 1000 [km] de longitud [1], [2], [3]. Debido a esto, aunque se conozca la ubicación de la falla no es posible determinar donde sucederán los sismos con adecuada precisión y es obligatorio recurrir al trabajo de expertos junto con sismógrafos para lograr localizar los epicentros una vez se originan los sismos.

Localizar el epicentro de los eventos sísmicos es una tarea de suma importancia, pues, una vez se determina este punto y su magnitud, las autoridades pertinentes pueden comenzar a tomar acción, ya sea para la evacuación de civiles o para guardar registro de este. Para el caso de sismos con origen volcánico, estos tienen un rol importante para estudiar la actividad volcánica [4], pues, proveen información útil que viaja largas distancias, permitiendo un monitoreo en casi tiempo real de la actividad volcánica.

En este trabajo se utilizaron sismos de origen volcánico ocurridos en el volcán Chillan y se debe destacar su complejidad frente a los sismos de origen tectónico. Según Keiiti Aki: "La estructura dentro de un volcán, es probablemente, el tema más complejo que los sismólogos han encontrado en el planeta Tierra" [5]. Lo antes mencionado se traduce en que las técnicas para identificar estos epicentros deben ser más sofisticadas que las usadas para eventos originados por el choque de placas, en particular, se requiere de expertos que cataloguen cada una de las señales tomadas por los sismógrafos para luego ingresarlas a un modelo de propagación que se encarga de localizar el epicentro[6].

En base a los tres párrafos anteriores, surge la motivación de implementar un sistema que sea capaz de localizar epicentros de origen volcánico, el cual debe ser rápido, preciso y no deba necesitar de expertos para su funcionamiento. Por esto se estima conveniente utilizar una estrategia *End to End* , la cual no requiera del modelo de propagación ni las señales catalogadas.

Debido a los excelentes resultados obtenidos en [7], [8] y [9] donde se catalogaron señales

sísmicas utilizando *Deep Learning*, se plantea que, el sistema localizador de epicentros se puede basar en esta tecnología para su implementación.

1.2. Descripción del problema

La problemática principal consiste en estimar epicentros de origen volcánico en las cercanías del volcán Chillan, utilizando tan solo las ondas sísmicas captadas por los sensores ubicados en el perímetros de este.

Se cuenta con 3 bases de datos con la información de eventos sísmicos. La primera contiene 953 eventos transcurridos entre diciembre del 2015 a septiembre del 2017, la segunda, 229 eventos correspondientes al año 2019 y, la tercera, 572 eventos ocurridos en el 2020. En estas 3 base de datos se almacenó la siguiente información:

- Localización del epicentro.
- Fecha en que se captó el evento con precisión de centésima de segundo.
- Coordenadas de los sensores.
- Ondas captadas por los sensores.
- Tasa de muestreo de los sensores.

La información más importante, y con la que se entrenará el modelo, corresponde a la localización de los epicentros y las señales de los sensores. Esto debido a que, todas las ondas almacenadas son del tipo VT (Volcano-Tectónico), lo cual significa que es posible extraer la distancia del sensor al epicentro. Luego, utilizando un mínimo de 3 sensores es posible triangular cada uno de los epicentros.

Las bases de datos no se encuentran libres de errores o dificultades agregadas y las más importantes se detallan a continuación:

- No se tienen los mismos sensores para las 3 bases de datos; la primera cuenta con 6, en la segunda, se elimina un sensor contenido en la primera y se agregan 2 nuevos, en la tercera, se agregan 2 sensores más.
- Solo se asegura que un mínimo de 3 sensores captaron las ondas sísmicas. Debido a la baja intensidad de los sismos, como por temas de calibración y mantenimiento, no todos los sensores se encuentran activos para todos los eventos.
- La primera base de datos contiene las ondas captadas en los 3 ejes, en cambio, las dos últimas bases tan solo contienen al eje Z.
- Existen epicentros en que se detalla la misma localización. Estos se omitirán de la base de datos, pues, se determinó que fueron erróneamente agregados.
- Existen eventos en los que no se detalló el epicentro.

Para afrontar estos puntos se trabaja con tan solo 3 sensores los cuales se encuentran en las 3 bases de datos. Los eventos que se utilizaron en el modelo de *Deep Learning* tienen los 3 sensores activados; tan solo se utilizará el eje Z. Luego de estos cambios, el total de eventos

contenidos en las bases de datos pasa de 1754 a 1532. Para aumentar la base de datos se propusieron técnicas de Data Augmentation

Para que el sistema sea capaz de estimar los epicentros, este requiere extraer la información de las ondas P y las ondas S contenidas en las ondas sísmicas. En particular, el momento en que llegan estas y cómo se relacionan entre ellas. Un ejemplo sacado de la base de datos se muestra en la figura 1.1 en donde se puede observar que las ondas P son las primeras en llegar y, además, tienen un bajo nivel de energía. Luego llega la onda S con una magnitud considerablemente mayor.

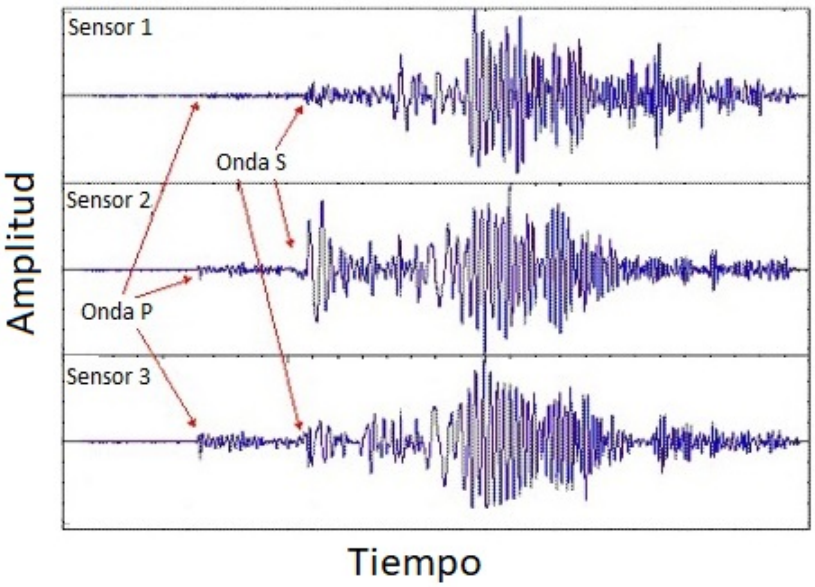

Figura 1.1: Señal sísmica de la base de datos capturada por 3 sensores simultáneamente. Fuente: Adaptada de [10]

Claramente, la energía es un factor clave para determinar la llegada de estas ondas, por ende, un desafío que se debe afrontar es el preprocesamiento que se realiza a las ondas. Esto con el fin de que el sistema de *Deep Learning* sea capaz de extraer la información necesaria para triangular. El preprocesamiento se encarga de extraer las características que ingresan a la red neuronal.

1.3. Objetivos

1.3.1. Objetivo General

Crear un sistema en base a *Deep Learning* que sea capaz de estimar, con la mayor precisión posible, los epicentros generados por actividad volcánica en las cercanías del volcán Chillán. Para esto, se busca minimizar el error de estimación del sistema con respecto al verdadero epicentro.

1.3.2. Objetivos Específicos

Para lograr el objetivo general de este trabajo se plantean los siguientes objetivos específicos:

- Proponer arquitecturas de *Deep Learning* y métodos de procesamiento de ondas que tengan la capacidad de resolver el problema planteado. Para esto, se debe realizar un análisis del estado del arte para los métodos de localización de epicentros de sismos volcánicos como de estructuras de redes neuronales que puedan ser capaces de realizar dicha tarea.
- Preprocesar correctamente la base de datos para que el sistema de *Deep Learning* utilizado capte la información necesaria para localizar epicentros. Esto se logra al demostrar que el sistema es aplicable en un conjunto de test obteniendo un error promedio inferior a los 10 [km].
- Aplicar técnicas de aumento de data para agrandar la base de datos y así disminuir el error promedio de estimación en comparación a métodos que no utilizaron esta técnica en el entrenamiento del sistema.
- Elegir y entrenar distintas arquitecturas de *Deep Learning*, para ir reduciendo cada vez más el error obtenido, planteando como meta un error promedio máximo de 1 [km].

1.4. Estructura de la Memoria

Este trabajo se ha separado por capítulos. En el capitulo 2 se muestra el marco teórico y el estado del arte. Estos con el fin de estudiar y entender los métodos clásicos de localización de epicentros para así extraer ideas y propuestas que mejoren los resultados del sistema. También se muestra la estructura de *Deep Learning* escogida y las ventajas que presenta esta frente a otras para abordar este problema en particular. En el capitulo 3 se da una descripción formal del problema, detallando la estructura de la base de datos, los ajustes que se realizaron con esta y la forma en que se construyeron los sistemas de *Deep Learning*. El capitulo 4 corresponde a la presentación de los resultados obtenidos con los modelos de redes neuronales, como también al análisis y discusión de estos. Finalmente, en el capitulo 5, se tienen las conclusiones, en donde se destacan las mayores dificultades del problema, cuál fue el mejor resultado obtenido y propuestas para el futuro del proyecto.

Capítulo 2

Marco Teórico y Estado del Arte

2.1. Marco teórico

2.1.1. Eventos sísmicos

La mayor parte de los eventos sísmicos son generados por el desplazamiento de placas tectónicas. Estos se propagan a través de ondas en todas direcciones desde su origen. En particular, y para interés de este trabajo, se propagan ondas P del tipo longitudinal y ondas S del tipo transversal. Las ondas P tienen una mayor velocidad que las S y, además, poseen un nivel energético menor. Dada esta característica y las diferencias de velocidad, al lograr identificarlas en una señal sísmica captada por un sismógrafo, es posible calcular la distancia del sismógrafo al origen del sismo.

También se pueden producir movimientos sísmicos producto de movimientos de magma, gases y fracturas dentro de los volcanes. Todos estos tienen denominaciones y características diferentes, pero se destacan los eventos sísmicos del tipo VT. Este tipo de sismo involucra procesos de fractura de material sólido al interior de un volcán. Suele ocurrir en rocas frágiles alrededor del reservorio de magma y en los conductos dentro del volcán o, incluso, a mayor profundidad dentro de la corteza. La característica más importante a mencionar es que estos sismos generan ondas P y en menor frecuencia ondas S, por lo tanto, en teoría es posible encontrar su origen de forma similar a los sismos generados por desplazamiento de placas.

2.1.2. Estimación de epicentros

La estimación de epicentros para sismos del tipo VT se puede realizar a través de una triangulación; esto al conocer la distancia del evento hacia al menos 3 sensores. Estas distancias se calculan utilizando el desfase temporal entre la llegada de la onda P y la onda S, sumado a un modelo de propagación [11] que viene determinado según las características geológicas del área. Se destaca que dicho modelo establece que las ondas P y S no se desplazan a velocidades constantes, si no que, su velocidad varía en función de las estructuras geológicas por las cuales se desplazan, tomando distintas velocidades en función del medios en que se encuentren. El picado de las ondas P y S debe ser realizado por expertos, en donde el método más preciso se obtiene al realizar una inspección visual de las ondas; claramente este es un proceso muy demandante en tiempo y recursos.

2.1.3. Transformada discreta de Fourier

La DFT (Transformada Discreta de Fourier) corresponde a una aplicación de la transformada de Fourier, en la que se toma una señal temporal discreta y se pasa a su dominio de la frecuencia. Esta transformada se basa en que toda señal temporal se puede llegar a descomponer como en una suma de funciones sinusoidales, las cuales forman una base ortogonal en el espacio de funciones. Una ejemplificación de la DFT para una señal compuesta por la suma de 2 sinusoidales se muestra en la figura 2.1, donde, a la izquierda de la figura, se aprecia la señal original en azul (mostrando sus 2 componentes sinusoidales en gris), al medio se aíslan las sinusoides y a la derecha se muestra el dominio de la frecuencia al aplicar DFT, demostrando claramente que la sinusoide de menor frecuencia tiene mayor energía.

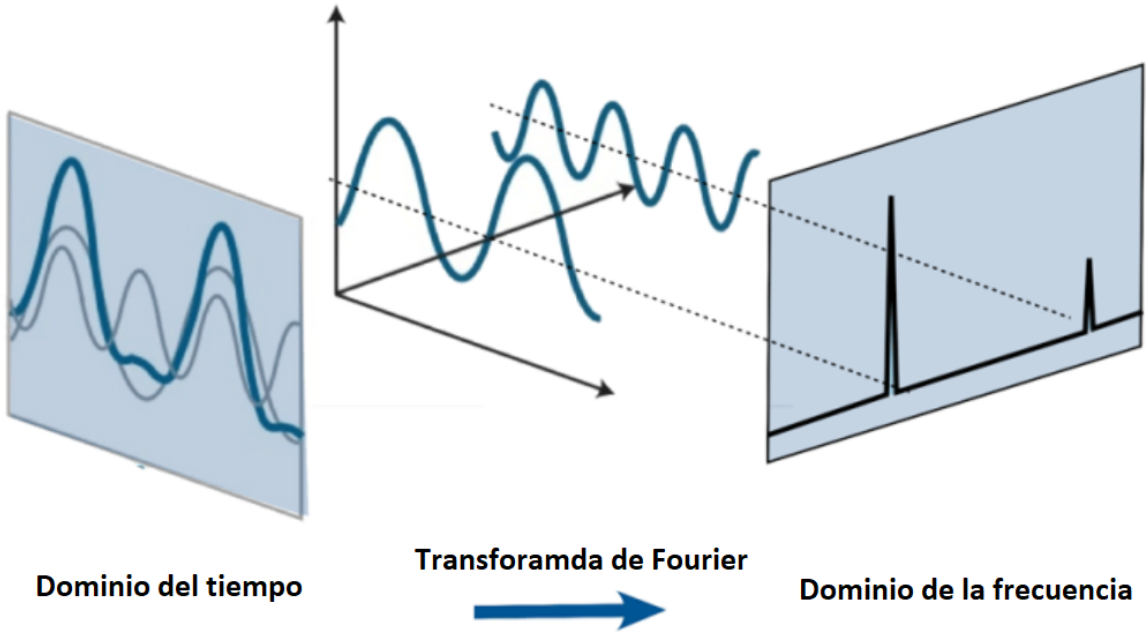

Figura 2.1: Representación gráfica de la transformada de Fourier. Fuente: Adaptada de [12]

La DFT se obtiene utilizando la ecuación 2.1.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-i2\pi}{N}kn} \tag{2.1}$$

para la ecuación 2.1 se toma en consideración una señal discreta de largo N donde $X[k]$ es el k-ésimo coeficiente que se obtiene con la DFT y $x[n]$ corresponde a la n-ésima muestra de la señal.

2.1.4. Transformada de Fourier en tiempo reducido

Si bien la DFT permite analizar el dominio de la frecuencia de una señal, al aplicar la DFT sobre toda la señal se está descartando el análisis temporal. A modo de ejemplo, si se quisiera estudiar como va aumentando la energía de las bandas de frecuencia en el tiempo,

entonces es necesario aplicar la DFT en distintas secciones de la señal.

Un método para analizar el dominio de la frecuencia y a la vez observar como varía este a lo largo de una señal temporal, corresponde la Transformada de Fourier en tiempo reducido (STFT). Esta es una aplicación de la DFT que se encarga de tomar la señal completa a analizar y dividirla en ventanas para posteriormente aplicar la DFT. A cada una de las ventanas se le puede aplicar una función de enventanado de manera que las primeras y últimas muestras dentro de cada ventana no generen artefactos de alta frecuencia.

La ecuación 2.2 corresponde al cálculo de la STFT, donde se puede observar su similitud con la ecuación 2.1. La diferencia radica en que existirán un vector $X[k]$ para cada ventana obtenida de la señal original. Luego, $w[n]$ cumple el rol de la función de enventanado al multiplicar término a término cada componente de la señal.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot w[n] \cdot e^{\frac{-i2\pi}{N}kn} \tag{2.2}$$

Para este trabajo se utilizó la función de enventanado llamada Hanning, en donde se desea suavizar con mayor medida las muestras más cercanas al inicio y al término de la ventana recortada. Esta función se puede observar en la figura 2.2.

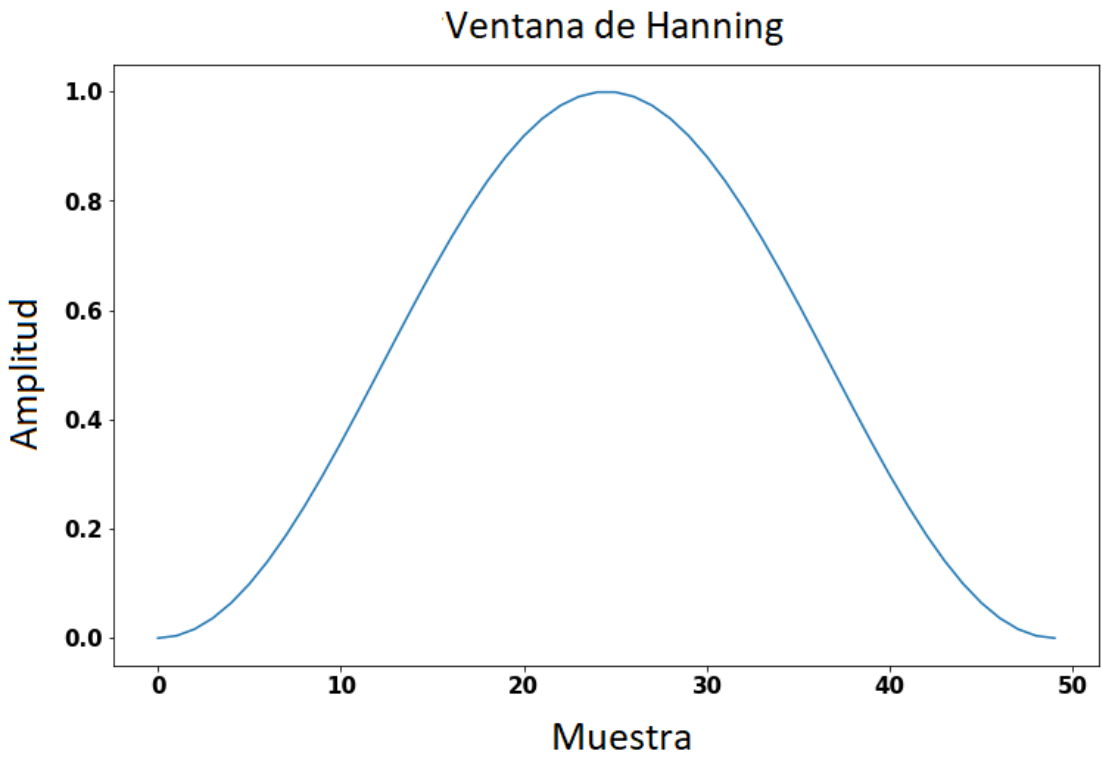

Figura 2.2: Función de Hanning para ventanas de tamaño igual a 50 muestras.

2.1.5. Redes neuronales

Las redes neuronales artificiales o, NN, son sistemas computacionales que se basan en el comportamiento de las redes neuronales biológicas, en donde, se tienen varias neuronas interconectadas entre ellas a través de distintos niveles. Las NN se componen de varios com-

ponentes llamados perceptrones los cuales son la unidad básica de las NN y emulan a las neuronas. Estos se conectan entre ellos a través de capas, donde la salida de cada capa entra a la siguiente, hasta llegar a una capa de salida que entrega la respuesta deseada.

Las interacciones entre las capas se realizan por procesos matemáticos que se definirán en función del tipo de NN. Para el caso del entrenamiento supervisado se utiliza una base de datos que contiene tanto el conjunto de características que entran a la red neuronal como el resultado al que se quiere llegar. Este método de entrenamiento se utiliza para que las NN aprendan de varios ejemplos con el fin de que estas sean capaces de generalizar con el conocimiento adquirido. El conocimiento que adquiere la NN se representa en forma de pesos y *bias*, los cuales son parámetros que se irán actualizando en el proceso de aprendizaje. Al igual que en los sistemas biológicos, las NN son capaces de seguir funcionando aún cuando fallen sus componentes (como por ejemplo perder algún porcentaje de pesos en algunas capas, de hecho, el *dropout* es un método que se basa en esto y se utiliza para mejorar la generalización). A continuación, se realiza una descripción más detallada de los componentes de las NN basándose en la notación y estructura vistas en la figura (2.3).

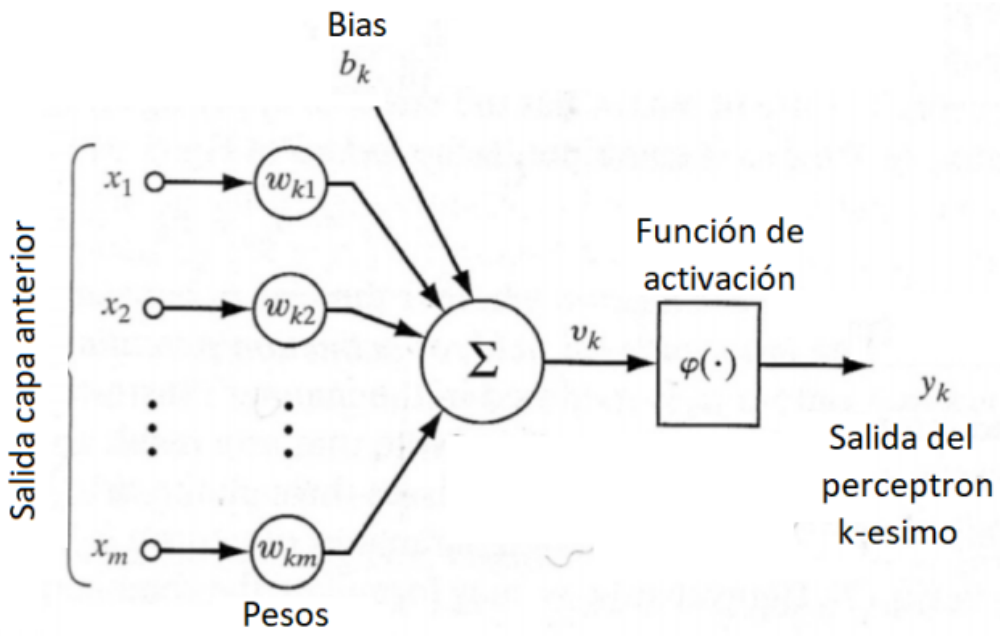

Figura 2.3: Representación gráfica del funcionamiento de un perceptrón. Fuente: Adaptada de [13]

El perceptrón es la unidad básica de las NN, a esta entran los valores de salida que arroja la capa anterior en forma de una suma ponderada como se presenta en la ecuación (2.3), donde w_{kj}, representa el peso asociado de la neurona (o entrada) j-ésima de la capa anterior a la neurona k-ésima, y el valor de x_j corresponde a la salida de la unidad j-ésima de la capa anterior. Al valor de u_k se le suma un sesgo o bias (se asocia uno para cada perceptrón) y luego se le aplica una función de activación φ para generar la salida del perceptrón; esto se representa por la ecuación (2.4).

$$u_k = \sum_{j=1}^{m} w_{kj} \cdot x_j \tag{2.3}$$

$$y_k = \varphi(u_k + b_k) \tag{2.4}$$

La capa de entrada corresponde al conjunto de características que se extraen de las ondas sísmicas de cada evento que ingresa a la NN, tanto para el proceso de entrenamiento como para obtener las estimaciones deseadas al ingresar eventos no vistos durante el entrenamiento.

En cada NN pueden haber una o más capas ocultas. Estas se caracterizan por tener una entrada proveniente de una capa anterior y arrojar una salida a la siguiente capa. Las salidas de las capas ocultas no tienen una interpretación clara como lo es con la capa de salida.

La capa de salida es la última capa en una NN y esta le llega la salida de la última capa oculta; arroja como resultado la respuesta deseada. Para este proyecto se requiere estimar el epicentro, lo que significa estimar latitud y longitud, por lo tanto, la capa de salida tendrá 2 perceptrones, entregando así el epicentro.

Los pesos denotados como w_{kj} corresponden a lo valores asociados con los que se ponderan las salidas de la capa anterior, como se ve en la ecuación (2.3). El uso de estos índices se debe a que se tienen j salidas en la capa anterior y k percetrones en la capa en cuestión. Por ende, se tendrán un total de $j \cdot k$ pesos asociados de la capa anterior a la actual (para el caso de una red feedforward). El bias corresponde a la adición de un sesgo al resultado de la suma ponderada. Tanto los pesos como el bias son los parámetros que almacenan el conocimiento adquirido tras realizar todo el proceso de aprendizaje de la NN, pues, estos serán los valores que se irán ajustando hasta obtener un resultado aceptable en el conjunto de validación (donde aceptable lo definirá el diseñador).

La función de activación se denotó como φ en la ecuación (2.4), corresponde a una función derivable (para poder realizar *back propagation*) que se emplea sobre la suma ponderada de (2.3) más el bias correspondiente. El uso de esta función se basa en la acción de la sinapsis nerviosa, en donde, en función de las señales que recibe una dendrita, estas son procesadas en la neurona y emite otra respuesta eléctrica a través del axón. Como la salida del axón esta limitada por un nivel de energía máximo, la función de activación sigmoide imita este comportamiento arrojando salidas de valor mínimo -1 y máximo 1. A la vez, se tiene que los axones no emiten respuestas negativas o positivas, si no que liberan impulsos con un cierto nivel de energía. Las funciones *ReLU* asemejan este comportamiento, pues, emiten una salida con valor mínimo igual a 0.

La combinación de las salidas de distintos percetrones, utilizando funciones no lineales, le otorgan gran capacidad a las NN de resolver problemas del tipo no lineal.

2.1.6. *Deep Learning* y redes LSTM

El *Deep Learning* se basa en las redes neuronales pero añadiendo más grados de complejidad, tanto para la actualización de pesos como a la cantidad de conexiones que tienen los nodos (ya no se utilizará el término perceptrón, pues nodo es más general y puede ser aplicado a las unidades básicas de las redes *Deep*).

Dentro de las redes *Deep* se encuentran las redes recurrentes, las cuales integran bucles de realimentación que permiten que la información se mantenga durante más épocas de entrenamiento, esto a través de conexiones de la salida de una capa hacia la entrada de si misma.

Las redes recurrentes desde ahora llamadas RNN, son capaces de traducir entradas secuenciales en salidas secuenciales [14] [15], haciendo uso de sus capacidad de modelamiento temporal. Por esto, hace completamente sentido utilizar RNN para procesar las señales de los eventos volcánicos, ya que, este problema es de naturaleza secuencial. Para largas secuencias de datos, al realizar el *back propagation*, el gradiente puede llegar a desaparecer al calcular las derivadas. Esto, claramente impone restricciones en cuán larga puede ser la secuencia de datos que ingresa a la RNN, si es que la información que se desea captar tiene dependencias de larga duración [15] [16]. Para superar este problema fue propuesta una arquitectura llamada LSTM (*long short-term memory*); en [17] se presentan por primera vez.

Las redes LSTM son un subconjunto de las RNN capaces de aprender dependencias de larga duración [17] y son mejores recordando información por largos periodos de tiempo. Este comportamiento es ideal para el problema de este trabajo, pues, para estimar la distancia de un sensor a un epicentro, se debe obtener la relación temporal entre la llegada de la onda P y la onda S, en donde la red *Deep* deberá preservar la información de la onda P hasta que llegue la S.

Las redes LSTM tienen la capacidad de añadir o eliminar información dentro del estado del nodo. Esto se regula a través de varias compuertas que permiten que la información fluya en el tiempo y permite el modelamiento de dependencias temporales de larga duración. Los nodos de las LSTM contienen una celda de memoria, una compuerta de entrada, una de salida y una de olvido. La celda de memoria almacena valores en intervalos de tiempo y las compuertas controlan cómo se van actualizando estos valores en el tiempo. De esta manera, las redes LSTM pueden procesar información de tipo secuencial y retener estados en el tiempo. En la figura 2.4 se puede apreciar una representación de un nodo de LSTM.

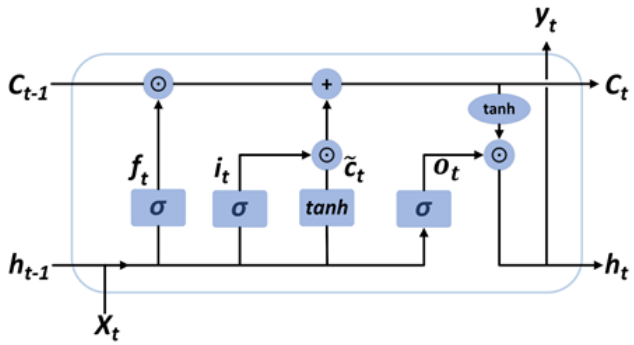

Figura 2.4: Representación gráfica de un nodo de LSTM

La compuerta de entrada i_t, olvido f_t y salida O_t se definen según las ecuaciones (2.5), (2.6) y (2.7), respectivamente. Para obtener el candidato a celda de memoria $\tilde{C}_t$ se utiliza la ecuación (2.8).

$$i_t = \sigma(X_t \cdot U^i + h_{t-1} \cdot W^i) \tag{2.5}$$

$$f_t = \sigma(X_t \cdot U^f + h_{t-1} \cdot W^f) \tag{2.6}$$

$$O_t = \sigma(X_t \cdot U^O + h_{t-1} \cdot W^O) \tag{2.7}$$

$$\tilde{C}_t = tanh(X_t \cdot U^g + h_{t-1} \cdot W^g) \tag{2.8}$$

La función sigmoide es representada por el símbolo σ y las matrices U^i, W^i, U^f, W^f, U^O, W^O, U^g y W^g son usadas para calcular los valores de cada compuerta. Estas son inicializadas al inicio del proceso de entrenamiento.

Estas compuertas determinan la cantidad de memoria antigua o nueva que debe ser ignorada para calcular el estado oculto actual h_t. La celda de memoria C_t del nodo es actualizada en el instante t según la ecuación (2.9).

$$C_t = C_{t-1} \cdot f_t + \tilde{C}_t \cdot i_t \tag{2.9}$$

Donde C_{t-1} es el contenido de la celda de memoria en el instante $t-1$. El estado oculto h_t es actualizado según la ecuación (2.10).

$$h_t = tanh(C_t) \cdot O_t \tag{2.10}$$

La compuerta de olvido decide que información deberá ser descartada o conservada de la celda de memoria del estado anterior del nodo C_{t-1}. Esta decisión se realiza al multiplicar C_{t-1} con el resultado de aplicar la función sigmoide de la compuerta de olvido en una combinación del estado oculto anterior h_{t-1} y la entrada actual X_t. Mientras más cercano a 0 sea este valor más se olvidará de la memoria y mientras más cercano a 1, más de la memoria se conservará.

El estado en tiempo t de la celda de memoria C_t se obtiene al añadir el producto anteriormente mencionado con el candidato de memoria $\tilde{C}_t$, el cual es controlado por la compuerta de entrada. El estado del candidato de memoria se obtiene al pasar el estado oculto y la actual entrada por la función *tanh*. Luego, la compuerta de entrada decide qué tanto contenido de $\tilde{C}_t$ debería ser añadido a la celda de memoria. Esta decisión se realiza por medio del valor de salida de la función sigmoide de la compuerta de entrada aplicada sobre la combinación de h_{t-1} y X_t.

La compuerta de salida decide cuál será el próximo estado oculto. Para esto, se pasa el anterior estado oculto y la entrada actual en la función sigmoide, luego, se pasa el estado del nodo (recién obtenido) por la función *tanh*, se multiplican los 2 resultados antes mencionados para decidir qué información deberá tener el próximo estado oculto. La nueva celda de memoria y el nuevo estado oculto son llevados al siguiente paso de tiempo.

2.1.7. *End to End*

Los esquemas basados en la técnica *End to End* se están convirtiendo en un enfoque práctico prometedor, donde el propósito de estos es obtener buenos resultados sin el uso a priori de un modelo [18] [19] como lo podría ser el modelo de propagación de las ondas sísmicas. Estos esquemas han demostrado resultados competitivos en el área del reconocimiento automático de voz, utilizando tan solo información de entrada y salida, sin la necesidad de aplicar algún modelo entremedio [20] [21] [22]. Para este trabajo, la motivación de usar un acercamiento

basado en *End to End* es evitar la necesidad de detectar las llegadas de las ondas P y S en conjunto con el modelo de propagación,

2.1.8. *Data Augmentation*

En general, mientras más compleja es la arquitectura de una red neuronal más datos de entrenamiento necesita para así poder ajustar los pesos y tener un buen rendimiento, La obtención de estos datos no es una tarea simple y mucho menos para este proyecto. Por esto, las técnicas de *Data Augmentation* tratan de resolver este problema al aumentar los datos de entrenamiento con datos generados artificialmente a partir de datos reales.

Esta técnica ha mostrado excelente resultados en el área de procesamiento de imágenes. Así lo demuestran los trabajos realizados en [23] [24] y aún más relevante es la implementación realizada en [8] donde se realiza *Data Augmentation* sobre señales sísmicas. Para esto toman señales originales y les añaden ruido gaussiano o alteran la forma de onda aleatoriamente sin interferir en las llegadas de ondas P y S. En general, no hay un consenso o técnica predeterminada para realizar *Data Augmentation* en ondas.

2.2. Estado del Arte

2.2.1. Estimación de epicentros con picado automático de ondas P y S

Según la investigación realizada, el problema de estimar epicentros de origen volcánico con un picado automático de las ondas P y S no ha sido desarrollado hasta el momento. Para eventos tectónicos el picado automático de estas ondas sí ha tenido un mayor desarrollo. Se han utilizado detectores basados en STA/LTA (*short term average/ long term average*), kurtosis y Skewness según los trabajos realizados en [25] [26] [27].

En la actualidad, gran parte de la investigación en picados automáticos se ha basado en *Deep Learning*. Esto debido a los excelentes resultados que se obtienen; estos se pueden observar en [7] [8] [9] en donde, utilizando complejas arquitecturas de redes neuronales, se han picado los tiempos de llegada de las ondas P y S con gran precisión.

Los métodos propuestos para picar las ondas dependen en gran medida de la base de datos que se utilice. Estos métodos requieren de un buen SNR y que las ondas (P y S) obtenidas por los sismógrafos sean lo suficientemente claras para que se obtengan buenos resultados, además, pocos de estos métodos son aplicados a las señales de tipo VT [28].

El hecho de que esta área no se ha desarrollado en vulcanología, se puede deber a que cuando los sismos ocurren dentro de un volcán, el bajo nivel de profundidad, la cercanía del epicentro a los sensores, el bajo nivel energético del sismo y la compleja propagación de las ondas por el medio, hacen que la detección de la onda S sea muchísimo más difícil [28]. Es más, según Keiiti Aki en [5] “La estructura dentro de un volcán, es probablemente, el tema más complejo que los sismólogos han encontrado en el planeta Tierra. Es extremadamente heterogéneo, anisotrópico y absorbente con interfases y tipologías irregulares, incluyendo fracturas de todos tamaño y orientación. Los procesos para generar estos sismos también son muchísimo más complejos que en el caso de los terremotos de origen tectónico, esto debido

a que se adicionan dinámicas de gases y magma que deben ser descifradas por sismólogos". Sin perjuicio de lo anteriormente mencionado en [28], se diseña un algoritmo para detectar la llegada de la onda S en sismos volcánicos VT, en donde de un total de 300 ondas procesadas 192 obtienen un picado de alta calidad de forma que el 97,05 % de estas tiene una diferencia menor a los 0,2 [s] con el picado manual.

2.2.2. Estimación de epicentros utilizando *Machine Learning*

En esta sección se muestran trabajos basados en *End to End* para encontrar los epicentros, es decir, no se hizo un picado automático o manual de las ondas P y S; no usaron modelos de propagación. En este tópico tampoco se ha encontrado literatura que enfrente el problema de localización para sismos de origen volcánico.

En [29] se buscó localizar pequeños sismos los cuales fueron inducidos por operaciones de extracción de petroleo y gas. En este caso se enfrentan contra señales con un bajo nivel de SNR, esto debido a la baja intensidad de los sismos. La red es entrenada con tan solo 1013 eventos reales y se realizó *Data Augmentation* para aumentar en 5 veces esta cantidad, se utilizaron 30 sensores y los 3 ejes de estos. Se obtiene un error de epicentro que promedia los 4,9 [Km].

En [19] se utilizó un método para detectar y localizar eventos sísmicos tectónicos, esto a través de redes del tipo CNN unidimensionales. Cabe destacar que se utilizaron 2 sensores y los 3 ejes de estos en una base de datos de 2709 eventos sísmicos. El problema de localización se abordó como uno de clasificación, en donde se subdividió el terreno donde surgían los sismos en 6 *clusters* de aproximadamente 100 $[km^2]$ cada uno. Su sistema fue capaz de clasificar un 74,5 % de los eventos en el *cluster* correcto.

En [30] se enfrentó el problema de localización de epicentros para sismos tectónicos que ocurrían en un *cluster*. Se utilizaron redes del tipo CNN unidimensionales, utilizando 9 estaciones y los 3 ejes de estas con un total de 2118 eventos sísmicos. Su trabajo especifica que un 86 % de los eventos testeados alcanza un error promedio de estimación de 200 [m] y un 69 % un error promedio de 100 [m].

2.2.3. *Data Augmentation* en señales temporales

La *Data Augmentation* puede aplicarse directamente sobre señales en el dominio del tiempo para luego ser procesadas e ingresadas a una red neuronal o se puede aplicar luego de procesar las señales. En [31] se busca clasificar sonidos utilizando redes del tipo CNN. Para aumentar su base de datos emplean técnicas que realizan deformaciones directamente a las señales de audio utilizando 4 técnicas por separado: en la primera, se modifica la velocidad de las ondas, en la segunda, se aumenta o baja el tono, en la tercera, se realiza una compresión de rango dinamica (DRC) y, en la cuarta, se añaden distintos niveles de ruido.

En [32] se realiza *Data Augmentation* para el reconocimiento de audio, se modifica la señal de audio aumentando o disminuyendo su velocidad.

En [33] es presentado un modelo de reconocimiento de emociones en discursos basado en redes CNN, el cual se entrenó utilizando espectrogramas obtenidos con *Data Augmentation*. Para la generación de los espectrogramas se emplearon redes generativas antagónicas (GAN).

Un estudio que también utiliza redes LSTM se muestra en [34], donde se busca hacer un clasificador de distintas actividades humanas a partir del CSI ($Channel\ State\ Information$)

de señales de WiFi. En este trabajo se realiza *Data Augmentation* sobre los espectrogramas de las señales, añadiendo distintos tipos de ruido y filtros de frecuencia, como también realizando shifting y escalamiento de frecuencias.

Capítulo 3

Metodología y Aportes del Trabajo de Memoria

3.1. Formalización del problema

El trabajo realizado en esta memoria tiene como fin la obtención de un sistema de localización de epicentros de origen volcánico en las cercanías del volcán Chillán. Este sistema solo contempla las cercanías de este volcán y probablemente no sea capaz de aplicarse en otros volcanes, debido a que sistemas basados en *Deep Learning* tienden a ajustarse a los modelos que reciben como referencia. Luego, como toda la base de datos es obtenida del volcán Chillan, se espera que la red neuronal se adapte a la estructura geológica por la cual se desplazaron las ondas sísmicas. Esta estructura varía en cada volcán y por lo tanto el sistema podría no ser aplicable en otros volcanes.

Si bien lo antes mencionado demuestra lo acotado que es este sistema, este proyecto busca en parte demostrar la capacidad de localizar epicentros volcánicos tan solo utilizando la información contenida en las ondas utilizando una estrategia *End to End*. Este problema no se ha abordado hasta la fecha y una vez demostrada la capacidad de las redes neuronales para enfrentar este proyecto, se espera generar un incentivo en la comunidad científica para investigar a mayor profundidad este tema.

El objetivo último de este trabajo es poder localizar todos los epicentros de la base de datos de test con un error máximo de 1 [km], puesto que, este es el rango de error máximo con el que trabaja el OVDAS (Red Nacional de Vigilancia Volcánica) y este organismo podría convertirse en usuario del sistema localizador, ya que este sería capaz de reducir los tiempos de localización y reducir los costos de personal involucrado.

Como se mencionó en la introducción del documento, este trabajo ha estado en progreso desde fines del año 2019. Por lo tanto, se han realizado experimentos periódicamente. A continuación, se muestran las características de la base de datos y lo que se ha estado realizado con ella.

3.2. Base de Datos

La base de datos cuenta con registros de eventos sísmicos captados con hasta 9 sismógrafos, simultáneamente con una tasa de muestreo de 100 [Hz]. Esta se divide en 3 grupos, la primera cuenta con 953 eventos los cuales se originaron entre diciembre del año 2015 hasta septiembre del 2017. En esta se tiene un máximo de 6 sismógrafos captando la onda sísmica. La siguiente base de datos que consta de 229 eventos corresponden al año 2019 (desde enero a diciembre). En esta se elimina un sensor de la base anterior y se agregan 2 nuevos. La última base de datos consta de 572 eventos registrados en el 2020 (desde enero a diciembre) y se agregan 2 sensores más.

Las bases de datos fueron creadas utilizando el programa *MATLAB* y la información que contiene cada una de estas es la siguiente:

- Localización del epicentro.
- Fecha en que se captó el evento con precisión de centésima de segundo.
- Coordenadas de los sensores.
- Ondas captadas por los sensores.
- Tasa de muestreo de los sensores.

El primer paso corresponde a unificar las bases de datos para un procesamiento más simple y directo con ellas. Se cambia el formato obtenido con *MATLAB* a uno que se pueda leer utilizando *Excel*.

La base de datos presenta diversos problemas que deben enfrentarse antes de su unificación, estos son:

1. No se tienen los mismos sensores para las 3 bases de datos.
2. Solo se asegura que un mínimo de 3 sensores captaron las ondas sísmicas, es decir, no todos los sensores captan ondas en todos los eventos.
3. La primera base de datos contiene las ondas captadas en los 3 ejes, en cambio las dos últimas bases tan solo contienen al eje Z.
4. Existen eventos en los que no se detalló el epicentro.

Para afrontar cada uno de los anteriores problemas se propone la siguiente solución a cada uno de ellos:

1. Se eliminan los sensores que no estén en las 3 bases de datos.
2. Se utiliza el mínimo de 3 sensores requeridos para localizar, en donde se escogieron los que están encendidos la mayor cantidad de eventos de forma simultánea. Esto reduce el número de eventos en 215.
3. Se trabaja con el eje Z.

4. Se determina que son posibles errores de tipeo al generar la base de datos en *MATLAB*, por esto, serán eliminados para futuros experimentos, reduciendo así en 5 la cantidad total de eventos.

5. No es posible entrenar o testear el modelo sin conocer los epicentros, por ende, se eliminaron estos 2 eventos.

Realizado este tratamiento se obtiene un archivo en *Excel* que contiene la información temporal de las ondas, el epicentro asociado a estas y un índice para no perder el nexo con la base de datos original.

3.2.1. Análisis exploratorio

En primer lugar se visualiza la distribución de los epicentros. En la figura 3.1 se observan todos los epicentros luego del filtrado. Se observa que la mayoría de estos se concentran en una sola área o cluster, dejando una minoría más alejados o en los extremos del mapa. En la figura 3.2 también se marca la ubicación de los 3 sensores utilizados. Se puede apreciar que están cercanos al área de alta densidad de epicentros.

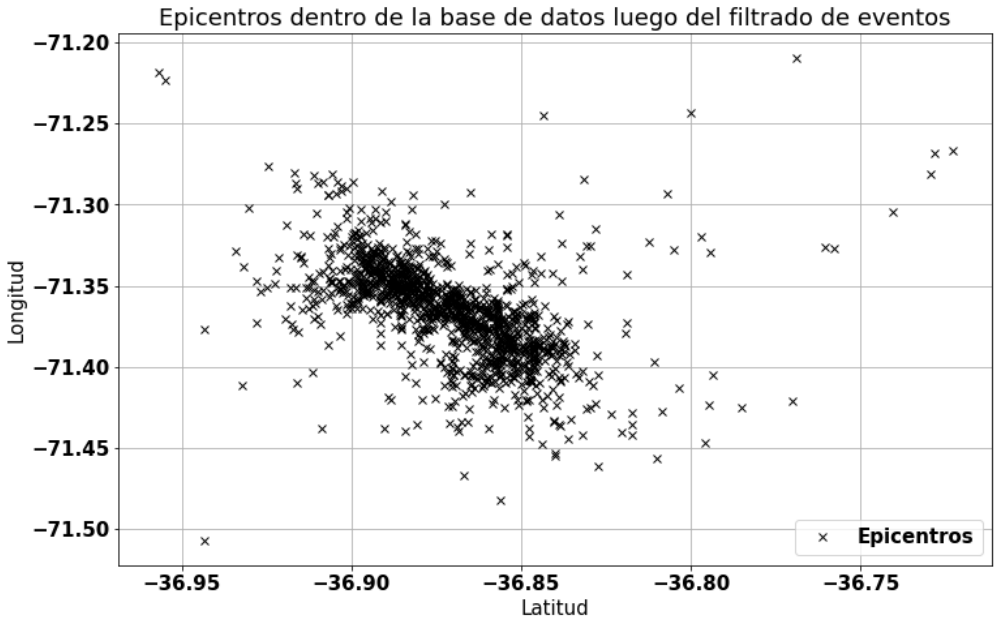

Figura 3.1: Ubicación de todos los epicentros de la base de datos posterior al filtro de eventos

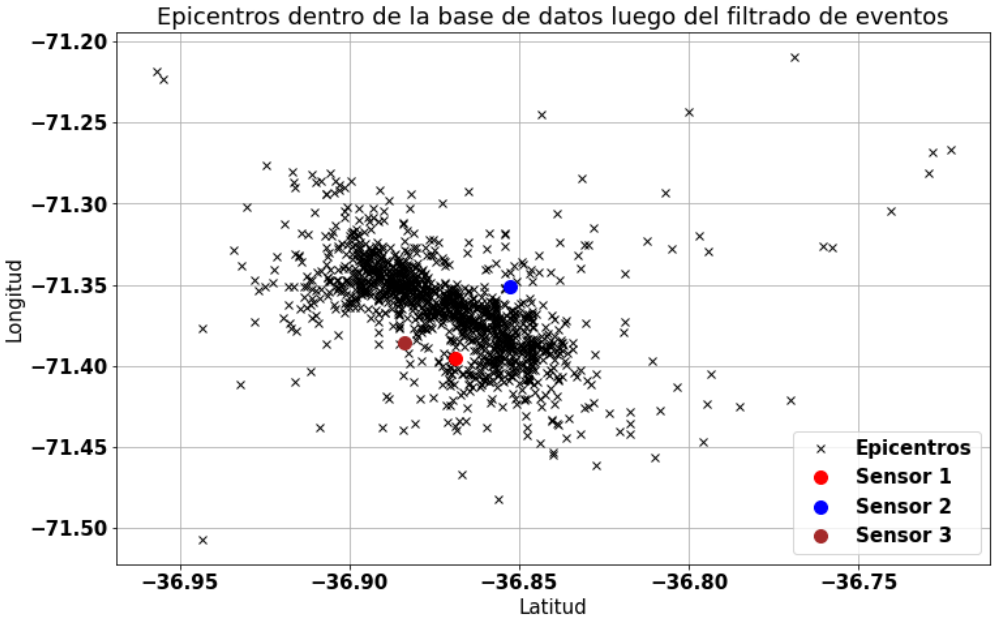

Figura 3.2: Ubicación de todos los epicentros de la base de datos posterior al filtro de eventos incluyendo sensores

Tomando como punto de referencia al centroide que se obtiene promediando todas las localizaciones de epicentros del conjunto de entrenamiento, se procede a visualizar la distribución de epicentros según la distancia de estos a la referencia; esto se presenta en la figura 3.3. Se observa que a mayor distancia de la referencia, la cantidad de eventos se va manteniendo pero, a la vez, la densidad de eventos disminuye pues hay mayor area que abarcar, con la excepción del rango de 3,5 a 4 [km] que es donde se encuentra la mayor cantidad de eventos.

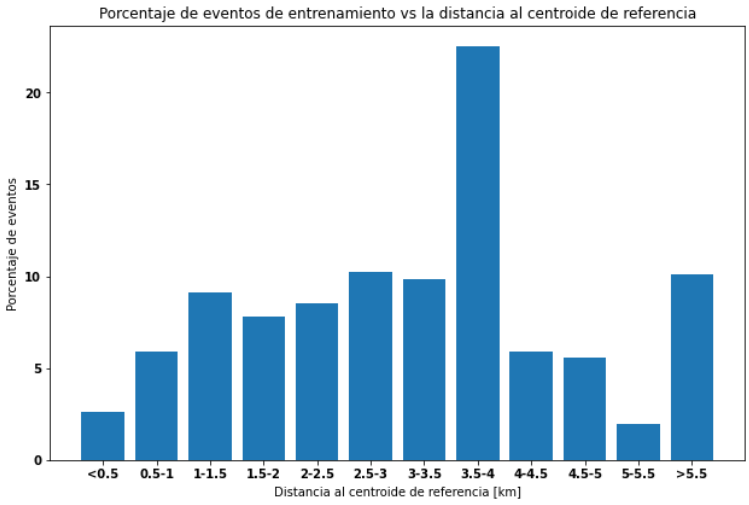

Figura 3.3: Distribución de epicentros de la base de entrenamiento en función de la distancia a la referencia

Las señales sísmicas fueron capturadas en un periodo de aproximadamente 4 años. Dada la gran cantidad de eventos sísmicos, se muestran algunas de las señales obtenidas en la figura 3.4 donde se procedió a recortar las señales originales desde el segundo 8 hasta el segundo

15. En ellas no se logra apreciar con claridad la llegada de las onda P y/o S.

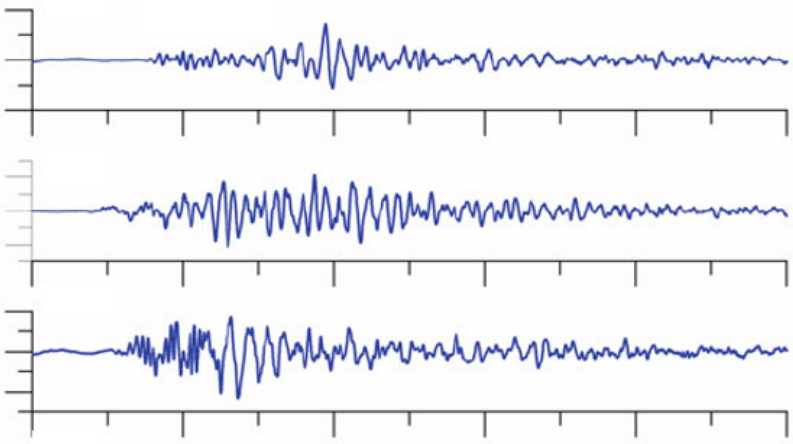

Figura 3.4: 3 Señales de la base de datos

Lo más destacable de las señales es que fueron catalogadas a mano por los expertos y se centró la llegada de la onda P en el segundo 10, aproximadamente. esto permite hacer un recorte adicional a las señales, capturando tan solo los segundos de interés correspondientes a la llegada de la onda P y posterior onda S. Se procede a realizar un corte inicial en el segundo 8 de cada señal, para dar 2 segundos de margen en que solo se captura ruido.

No se conoce la llegada de la onda S pero tomando como base a un modelo de velocidades entregado por la UFRO, la mínima velocidad a la que se transmite la onda P es 6.1 km/h y la onda S tiene, aproximadamente, 1.7 veces a velocidad de la onda P. Por esto, se decide recortar la señal hasta el segundo 15, pues, con esos 5 segundos que se están dando luego de la llegada de la onda P se permite capturar la onda S si y solo si el epicentro fue generado como máximo a 18 km de distancia del sensor. La mayor distancia de un epicentro a un sensor es de 16.83 km, por ende, con este recorte, se espera estar capturando todas las ondas P y S.

3.3. Preprocesamiento de las ondas temporales

En esta sección se describirá la metodología con la que se procesan las señales captadas por los sismógrafos.

Viendo el estado del arte, se plantea nunca ingresar las ondas temporales sin antes hacer un preprocesamiento a ellas. Esto para que a la red neuronal ingrese información lo más representativa posible y su entrenamiento sea más rápido y eficiente.

Inicialmente, las señales son recortadas en ventanas con un cierto traslape entre ellas, como se observa en la figura 3.5 donde se extraen 2 ventanas con un traslape del 50 %.

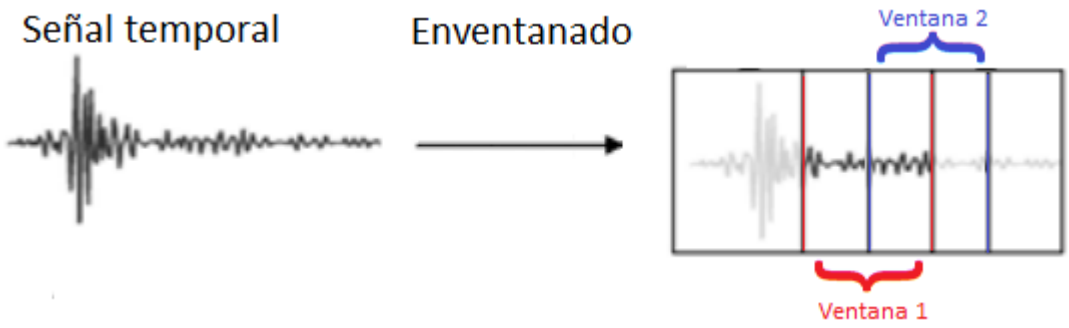

Figura 3.5: Enventanado de una señal. Fuente: Propia

Posteriormente, se extrae información de cada ventana utilizando la DFT para así representar los niveles de energía en cada banda de frecuencia (denominadas *bins*). Luego de obtener la DFT de cada ventana estas son concatenadas. Para seguir la lógica de la figura 3.5 se muestran los pasos correspondientes en la figura 3.6.

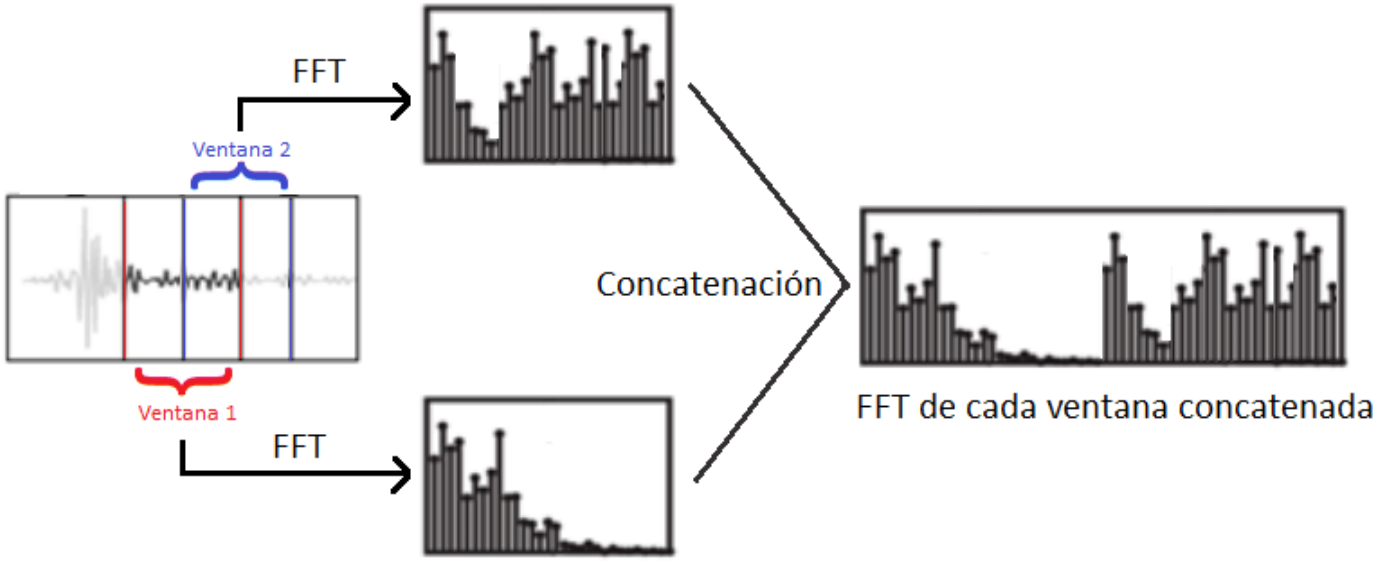

Figura 3.6: Obtención de DFT y posterior concatenación de estas. Fuente: Propia

Ya obtenidas las DFT concatenadas de cada señal temporal se procede a normalizar esta información para así ingresarla a la red neuronal. Para esto, se aplica valor absoluto a cada bin de frecuencia en cada una de las ventanas perdiendo así la fase pero obteniendo una representación de la energía con un solo valor. Luego de aplicar valor absoluto se aplica logaritmo natural a cada valor obtenido anteriormente. Lo que se aplicó a cada bin de frecuencia se muestra en la ecuación (3.1)

$$x_{i,j} = \ln(abs(bin_{i,j})) \tag{3.1}$$

Donde el subíndice “i” corresponde a la ventana obtenida y el subíndice “j” corresponde al rango de frecuencia del bin.

Si bien la energía captada por cada sensor dependerá de la distancia de este al epicentro, por temas de calibración, se decide normalizar por bin y por sensor. Para esto, se agrupan todos los bins obtenidos para cada ventana en que se aplicó DFT (según su banda de frecuencia) y se obtiene promedio y desviación estándar de estos. Posteriormente, a cada bin, se le resta su promedio y desviación estándar respectiva. La normalización empleada es representada en la ecuación (3.2) donde la notación ":,j" indica que se obtienen los bins de todas las ventanas para un rango de frecuencias "j".

$$\overline{x}_{i,j} = \frac{x_{i,j} - prom(x_{:,j})}{std(x_{:,j})} \tag{3.2}$$

3.4. Preprocesamiento de coordenadas de epicentros

Para facilitar el entrenamiento de la red, se realiza una transformación lineal a todas las las coordenadas de epicentros, para que estas queden en el rango de 0 a 1. Para lograr esto, se tomaron los puntos máximos y mínimos de latitud y longitud de todos los eventos y se procedió a realizar una transformación lineal.

Una vez realizado este paso se obtiene el nuevo mapa de coordenadas para los epicentros. Donde las estimaciones de los epicentros se realizan en el rango de 0 a 1 tanto para latitud como longitud. Este valor es posteriormente transformado a la latitud y longitud real realizando la transformación lineal antes aplicada pero ahora de forma inversa.

3.5. Primer método de generación de señales con *Data Augmentation*

Basándose en el trabajo realizado en [8], se decide generar señales sísmicas artificiales añadiendo distintos niveles de ruido gaussiano a las señales originales. De esta manera se está conservando el epicentro asociado y se permite generar una cantidad prácticamente ilimitada de señales. La consideración más importante en esta aplicación de *Data Augmentation*, es no añadir un nivel de ruido tan grande que inhabilite a la red de percibir la llegada de ondas P y S. Por esto, se decide no añadir más de 50 [dB] de ruido a cada señal.

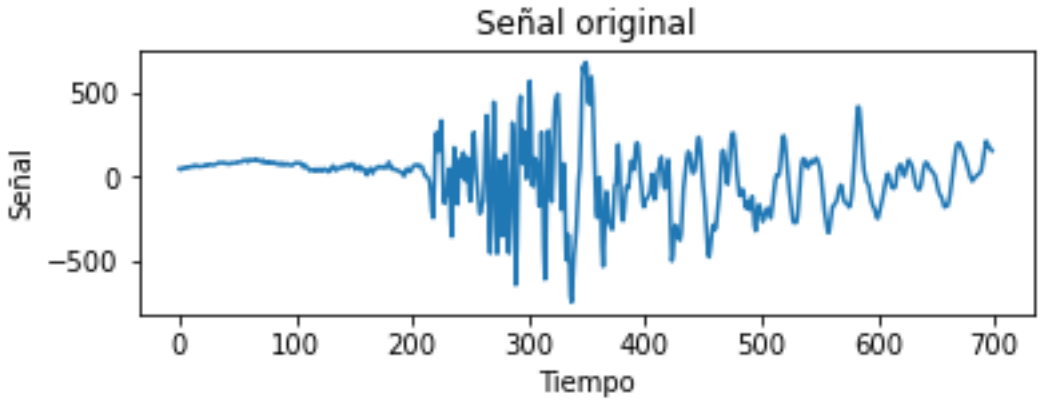

Figura 3.7: Señal de la base de datos

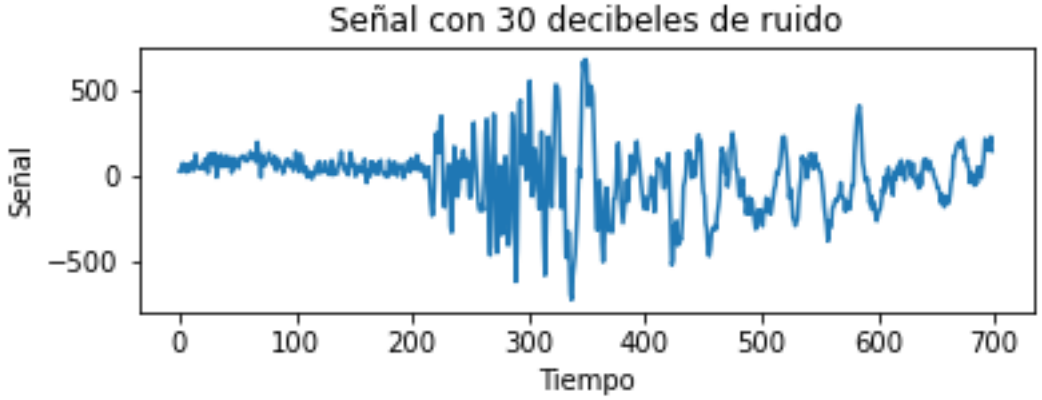

Figura 3.8: Señal de la base de datos al añadir ruido

En la figura 3.7 se muestra una señal de la base de datos y, luego, en la figura 3.8, se presenta la misma señal con 30 [dB] de ruido añadido. Para esta señal en particular se siguen apreciando visualmente las llegadas de la onda P y S aún con 30 los [dB] de ruido, pero no se puede asegurar esto para todas las señales, pues algunas poseen un ruido considerable y es difícil percibir las llegadas de las ondas aún sin la adición de ruido gaussiano extra.

Para enfrentar el problema antes mencionado, se generan 5 bases de datos de entrenamiento distintas obtenidas con *Data Augmentation*. La primera base de datos contiene las señales originales y un duplicado de los datos de entrenamiento al añadir 10 [dB] de ruido. La segunda base de datos es igual a la primera pero añadiendo las señales de entrenamiento con 20 [dB] de ruido. Siguiendo la lógica anterior para todas las bases generadas, la última corresponde a las señales de entrenamiento originales junto con las mismas añadiendo 10, 20, 30, 40 y 50 [dB] de ruido, teniendo en esta 6 veces la cantidad original de datos para entrenar. Con esto, se logran obtener 5 bases de datos distintas con 2260, 3390, 4520 y 5650 señales de entrenamiento. Estos números se deben a que la data se generó utilizando tan solo un conjunto de la base de datos. Este conjunto corresponde a los datos con que se entrena la red neuronal.

3.6. Segundo método de generación de señales con *Data Augmentation*

Para este método, se toma en consideración la distribución de los epicentros con respecto a la distancia del centroide de referencia. Como se muestra en la figura 3.3 existe un rango de distancia que acumula la mayoría de los eventos y hay otros rangos que contienen una cantidad considerablemente menor. Por este motivo se propone regularizar la distribución de los eventos, generando más datos en los rangos con menos cantidad de epicentros y llegar a una distribución más uniforme. Para realizar esto se aíslan los eventos en conjuntos según los rangos de distancia de la figura 3.9 y se multiplicará cada conjunto por un entero entre 1 y 6 a fin de dejar la distribución lo más pareja posible. Los resultados de este proceso son mostrados en la figura 3.9

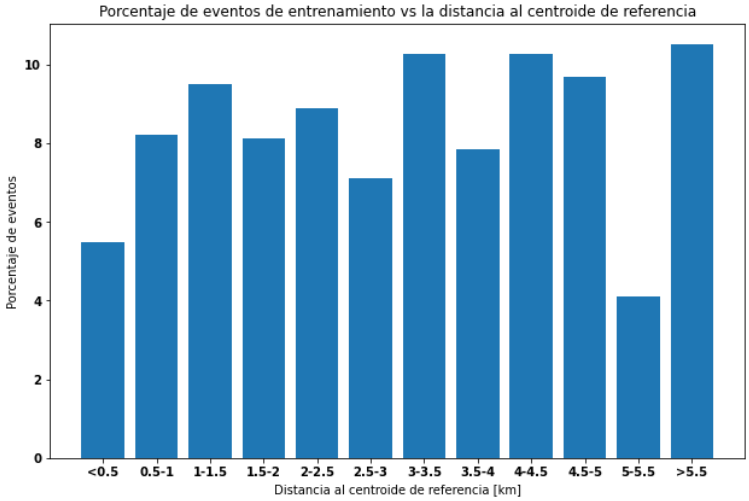

Figura 3.9: Distribución de epicentros de la base de entrenamiento en función de la distancia a la referencia luego de aplicar *Data Augmentation*

Según el valor con el que se multiplicó cada conjunto es la cantidad de eventos que se generan con *Data Augmentation*. Por ejemplo, los eventos entre 0 a 0.5 [km] son multiplicados por 6 y en consecuencia se generan señales de entrenamiento con 10, 20, 30, 40 y 50 [dB] de ruido. En cambio, los eventos entre 2 a 2.5 [km] son multiplicados por 3 y tan solo se añadirán señales con 10 y 20 [dB] de ruido.

Cada rango fue multiplicado con los siguientes valores:

- Rango de 0 a 0.5 [km] = 6
- Rango de 0.5 a 1 [km] = 4
- Rango de 1 a 1.5 [km] = 3
- Rango de 1.5 a 2 [km] = 3
- Rango de 2 a 2.5 [km] = 3
- Rango de 2.5 a 3 [km] = 2
- Rango de 3 a 3.5 [km] = 3
- Rango de 3.5 a 4 [km] = 1
- Rango de 4 a 4.5 [km] = 5
- Rango de 4.5 a 5 [km] = 5
- Rango de 5 a 5.5 [km] = 6
- Rango de más de 5.5 [km] = 3

Siguen existiendo conjuntos con considerablemente menos datos que el resto, como en los rango de 0 a 0.5 [km] y en el rango de 5 a 5.5[km]. Esto se debe a que se estableció no generar datos con más de 50 [dB] de ruido y al generar señales con un ruido que no fuera múltiplo de 10 [dB] no se logra apreciar una diferencia notable entre los rangos superiores e inferiores. Es decir, señales con 15 [dB] de ruido son muy similares a las de 10 y 20 [dB]. Aún así, se ha logrado homogeneizar considerablemente la distribución de los datos y se obtiene una nueva base de datos de entrenamiento con un total de 3508 eventos.

3.7. Entrenamiento de las redes neuronales

En la sección anterior se indicó cómo se obtiene la entrada de la red y la salida de la red. En esta sección se mostrarán los procedimientos utilizados para entrenar diversos tipos de arquitecturas y cómo se evalúan los resultados obtenidos.

Tras una revisión bibliográfica se determinó que la arquitectura de las redes LSTM podría tener la capacidad de resolver este tipo de problemas. Esto debido a su capacidad de capturar las dinámicas en señales variables en el tiempo y remover o añadir información dentro de la celda de memoria. Esto le da la capacidad de modelar dependencias a largo y corto tiempo. Se decide añadir capas *Fully Connected* tras las capas de LSTM para así obtener la latitud y longitud de los epicentros en la salida de la red.

El primer paso corresponde a separar la base de datos en 3 conjuntos. Al primero se le llama conjunto de entrenamiento y corresponde a un 80 % de los datos; es utilizado para entrenar y obtener los pesos de la red. El segundo conjunto, con un 10 % de los datos, es el de validación y se utiliza para evaluar el desempeño de la red tras cada época de entrenamiento y, finalmente, con el 10 % restante se arma el conjunto de test, sobre el cual se evalúa

el desempeño de las redes neuronales que mejor rendimiento obtuvieron en el conjunto de validación. Cabe destacar que los conjuntos se armaron dividiendo la base datos luego de que se ordenara aleatoriamente, para así tener una distribución lo más homogénea posible de eventos por año. Con esto, sin considerar la data generada artificialmente, se obtienen 1130 eventos para entrenamiento, 141 para validación y 142 para test.

Luego, corresponde diseñar las arquitecturas de las redes neuronales a entrenar según una serie de hiperparámetros que se escogen de manera arbitraria. Estos corresponden a: la taza de aprendizaje, el tipo de capa a utilizar, la cantidad de nodos por capa, la cantidad de capas, la función de activación de cada capa, la función de error, el optimizador y la cantidad de épocas. También se deben escoger los parámetros para procesar y obtener la entrada de la red. Estos corresponden a la cantidad de puntos en la transformada de Fourier, el ancho de las ventanas y el traslape entre estas.

Claramente, es una cantidad extensa de hiperparámetros que se deben ajustar con el fin de obtener el mejor resultado posible. Para la búsqueda de hiperparámetros se obtienen una serie de opciones a evaluar, acotando los hiperparámetros que se moverán en cada sesión de enfrenamiento como también en qué rangos se moverán estos. Este paso es de suma importancia, pues, la capacidad de cómputo es limitada y, por ende, mientras más extensiva es la búsqueda más entrenamientos se realizan y, en consecuencia, más tiempo se consume. Por este motivo las sesiones de entrenamiento están diseñadas para no durar más de 2 días y una vez que estas concluyen, se evalúa cada uno de los modelos entrenados considerando las siguientes métricas sobre un conjunto de evaluación:

- Error promedio de estimación.
- Desviación estándar del error.
- Porcentaje de eventos cuya estimación tiene error menor al kilometro.

Para asegurar un buen comportamiento de la red neuronal entrenada, se realizan 3 entrenamientos por cada set de hiperparámetros y las 3 métricas obtenidas anteriormente son promediadas para evaluar el rendimiento de cada entrenamiento.

En la tabla 3.1 se muestran los valores que se utilizaron en los primeros 6 meses de entrenamiento de la red.

Tabla 3.1: Set de parámetros e hiperparámetros barridos

Hiper/parámetro	**Utilizados**
Ancho de ventanas	100 - 300 - 500 - 1000 - 2000 [ms]
Traslape entre ventanas	20 - 30 - 40 - 50 - 60 - 70 %
Número de bins obtenido con DFT	15 - 31
Número de épocas	500
Taza de aprendizaje	0,83 - 0,083 - 0,0083 - 0,00083
Optimizador	ADAM - ADAGRAD
Métrica a optimizar	MAE - MSE
Capas LSTM	1 - 2 - 3 - 4 - 5 - 6
Función de activación (LSTM)	Relu
Nodos por capa LSTM	8 - 16 - 32 - 64 - 128 - 256 - 512 - 1024
Capas *Fully connected*	1 - 2 - 3 - 4
Función de activación (*Fully Connected*)	Relu - Sigmoide
Nodos por capa *Fully Conncted*	8 - 16 - 32 - 64 - 128 - 256 - 512 - 1024

Se debe mencionar que no se probaron todas las combinaciones posibles utilizando los valores mostrados en la tabla 3.1, pues en ese caso se estarían generando 2.949.120 entrenamientos distintos donde el más rápido de ellos tomó 20 segundos. Para reducir considerablemente este número, como primera medida, se buscó fijar el número de bins, ancho y traslape de ventanas. Para esto se fijaron 2 capas por de LSTM y 2 de *Fully connected* con un total de 64 nodos cada una; con esto se generaron 1920 entrenamientos. Tras barrer estos parámetros se optó por dejar fijo el ancho de las ventanas de 500 [ms], traslape del 60 % y un total de 31 bins para todos los futuros entrenamientos, pues, en promedio se obtuvo un mejor rendimiento en el conjunto de validación utilizando estos valores.

La función de error que se optó para seguir entrenando la red corresponde a la función MAE (*Mean Absolute Error*), por lo tanto, la red está buscando minimizar el promedio del error absoluto entre estimaciones y verdaderos epicentros dentro del conjunto de entrenamiento. Utilizando la función MSE (*Mean Square Error*) se obtiene un error promedio de estimación menor, pero el porcentaje de eventos cuya estimación tiene error menor al kilómetro, es considerablemente menor.

3.7.1. Entrenamiento utilizando *Data Augmentation*

Una vez finalizados los entrenamientos de la red sin data generada, se procede a entrenar utilizando las técnicas de *Data Augmentation*. Para este caso no se utilizan las ondas creadas artificialmente en el conjunto de validación o de test, tan solo en el conjunto de entrenamiento, por este motivo, los conjuntos de validación y test se mantienen idénticos.

Debido al aumento de la base de datos se redujo el número de hiperparámetros a barrer, con el fin de lograr resultados en menor tiempo. En la tabla 3.2 se muestran todas las combinaciones utilizadas.

Tabla 3.2: Set de hiperparámetros barridos utilizando *Data Augmentation*

Hiper/parámetro	**Utilizados**
Ancho de ventanas	500 [ms]
Traslape entre ventanas	60 %
Número de bins obtenido con DFT	31
Número de épocas	250
Taza de aprendizaje	0,83 - 0,083 - 0,0083 - 0,00083
Optimizador	ADAM
Métrica a optimizar	MAE
Capas LSTM	1 - 2 - 3 - 4
Función de activación (LSTM)	Relu
Nodos por capa LSTM	32 - 64 - 128 - 256
Capas *Fully connected*	1 - 2 - 3 - 4
Función de activación (*Fully Connected*)	Relu
Nodos por capa *Fully Conncted*	16 - 32 - 64 - 128 - 256 - 512

Capítulo 4

Resultados

4.1. Utilizando la base de datos original

En esta sección se muestran los mejores resultados obtenidos sin utilizar *Data Augmentation*.

En la tabla 4.1 se muestra el set de valores que ha logrado el mejor resultado.

Tabla 4.1: Set de parámetros e hiperparámetros que dan el mejor resultado sin utilizar *Data Augmentation*

Hiper/parámetro	**Mejor a la fecha**
Ancho de ventanas	500 [ms]
Traslape entre ventanas	60 %
Número de bins obtenido con FFT	31
Número de épocas	500
Taza de aprendizaje	0,00083
Función de optimización	ADAM
Métrica a optimizar	MAE
Capas LSTM	4
Función de activación (LSTM)	Relu
Nodos por capa LSTM	256
Capas *Fully connected*	2
Función de activación (*Fully Connected*)	Relu
Nodos por capa *Fully Conncted*	64

Con los valores mostrados en la tabla 4.1 se llegó a un error promedio de 1,51 [km] con desviación estándar de 1,56 [km]. Se logró que un 48,48 % de los eventos en la base de datos de test tengan error de localización menor a 1 [km].

La arquitectura de la red se compone de, inicialmente, 4 capas ocultas de LSTM a lo que siguen 2 capas ocultas de *Fully Connected* con un *dropout* del 10 % en estas últimas y, finalmente, la capa de salida que tan solo tiene 2 nodos para predecir la latitud y longitud del epicentro.

En la figura 4.1 se muestra un histograma obtenido al evaluar el rendimiento de la red

sobre el conjunto de test, en donde cada barra corresponde al porcentaje de eventos cuya estimación de la red neuronal tiene un error menor al mostrado en el eje inferior. Es decir, un 27,27 % de los eventos tiene un error entre 0 a 500 metros comparado con la *ground truth* (o el verdadero epicentro).

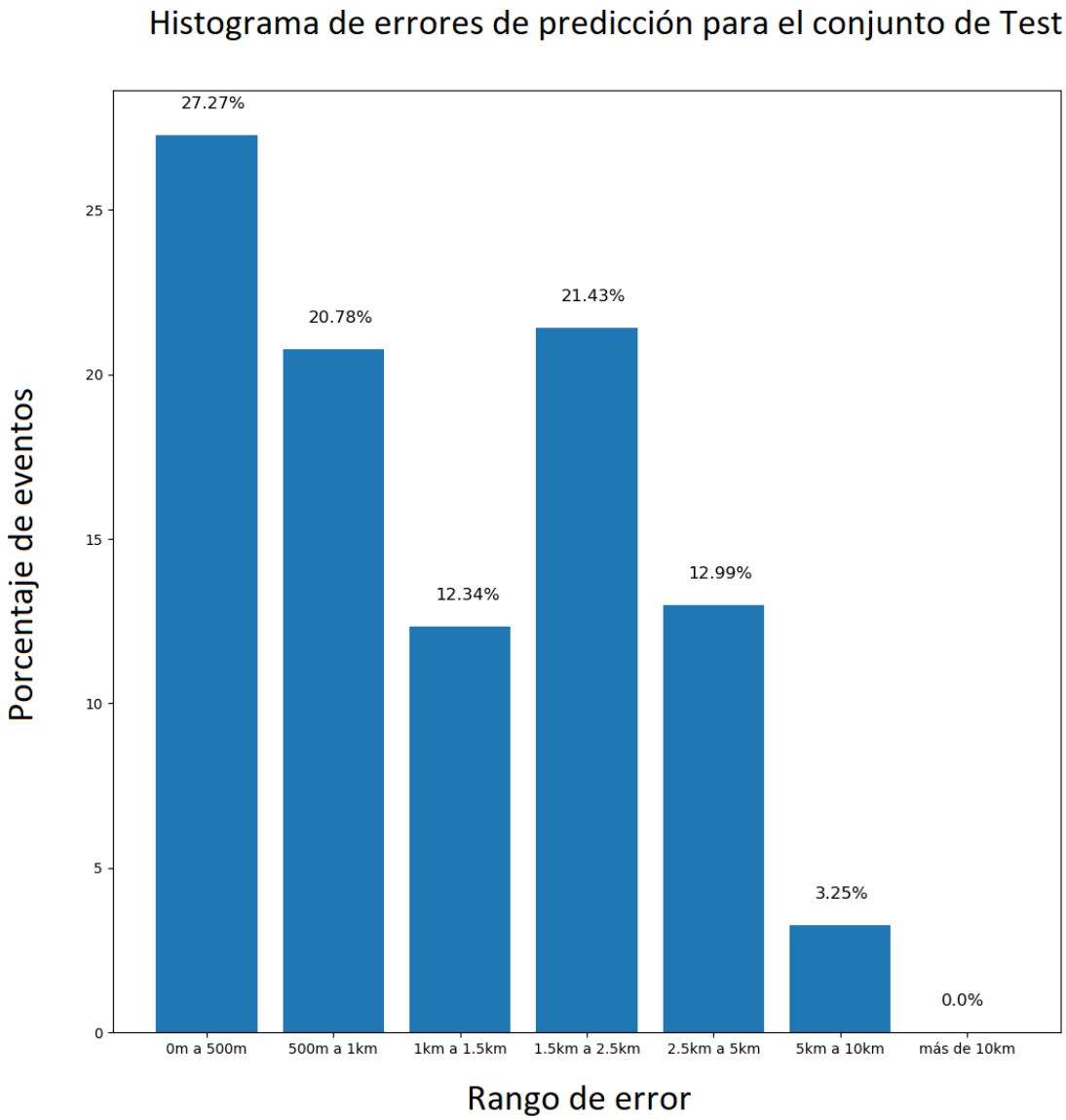

Figura 4.1: Histograma del error de estimación de la red neuronal para el conjunto de test

En la figura 4.2 se muestra el mapa, marcando en rojo la *ground truth*, y en cruces azules las estimaciones que entrega la red.

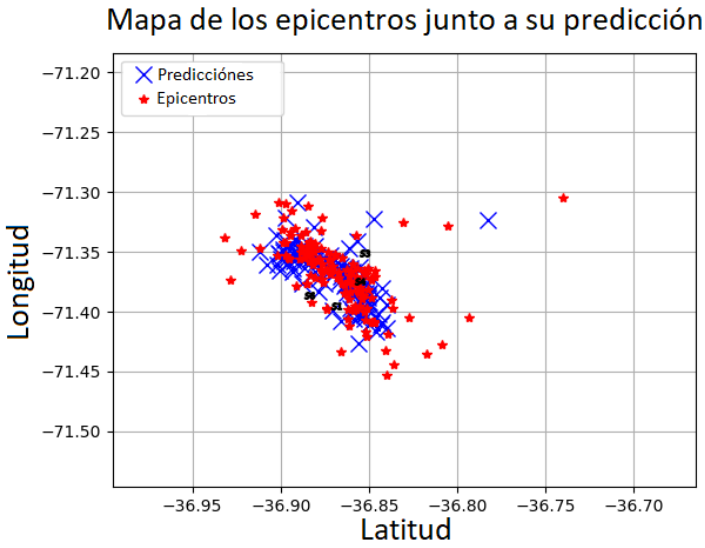

Figura 4.2: Mapa con las localizaciones de los epicentros de test y sus estimaciones

Como se mencioná en la sección de metodología, cada entrenamiento de la NN se repitió 3 veces. Por este motivo, las figuras 4.1 y 4.2, corresponden al resultado de la primera realización del entrenamiento. Las figuras obtenidas con los otros 2 entrenamientos se omiten por redundancia, pues, son casi idénticas. El error promedio de 1,51 [km] con desviación estándar de 1,56 [km] junto con el 48,48 % de eventos con error menor al kilómetro, es el resultado de promediar los 3 entrenamientos.

En la figura 4.1 es posible notar que más del 60 % de las estimaciones tienen un error menor a los 1.5 [km] y ni una de las estimaciones supera los 10 [km] de error, con un bajo porcentaje de eventos con error entre 5 a 10 [km]. Se puede apreciar en la figura 4.2 que los epicentros más alejados del cluster de eventos no aparentan tener estimaciones de la red asociadas a ellos en sus cercanías, pues, la red concentra sus estimaciones dentro del mismo cluster de eventos con la excepción de un par de estimaciones más apartadas del cluster.

4.2. Utilizando *Data Augmentation*

4.2.1. Primer método

A continuación se presentan los mejores resultados con el primer método de *Data Augmentation*, es decir, replicando toda la base de entrenamiento con distintos niveles de ruido inducidos. El mejor resultado se obtuvo al añadir ruido de hasta 40 [dB] a las señales, es decir, con un total de 4520 eventos de entrenamiento. El set de valores que entregó el mejor resultado se muestra en la tabla 4.2

Tabla 4.2: Set de parámetros e hiperparámetros que dan el mejor resultado utilizando el primer método de *Data Augmentation*

Hiper/parámetro	**Utilizados**
Ancho de ventanas	500 [ms]
Traslape entre ventanas	60 %
Número de bins obtenido con DFT	31
Número de épocas	250
Taza de aprendizaje	0,00083
Optimizador	ADAM
Métrica a optimizar	MAE
Capas LSTM	4
Función de activación (LSTM)	Relu
Nodos por capa LSTM	256
Capas *Fully connected*	3
Función de activación (*Fully Connected*)	Relu
Nodos por capa *Fully Conncted*	128

Con los valores mostrados en la tabla 4.2 se llegó a un error promedio de 1,42 [km] con desviación estándar de 1,69 [km]. Se logró que un 51,51 % de los eventos en la base de datos de test tengan error de localización menor a 1 [km].

Similar a los resultados obtenidos sin *Data Augmentation*, la arquitectura de la red se compone de inicialmente 4 capas ocultas de LSTM a lo que siguen 3 capas ocultas de *Fully Connected* con un *dropout* del 10 % en cada una y, finalmente, la capa de salida que tan solo tiene 2 nodos para predecir la latitud y longitud del epicentro.

En la figura 4.3 se muestra un histograma obtenido al evaluar el rendimiento de la red sobre el conjunto de test.

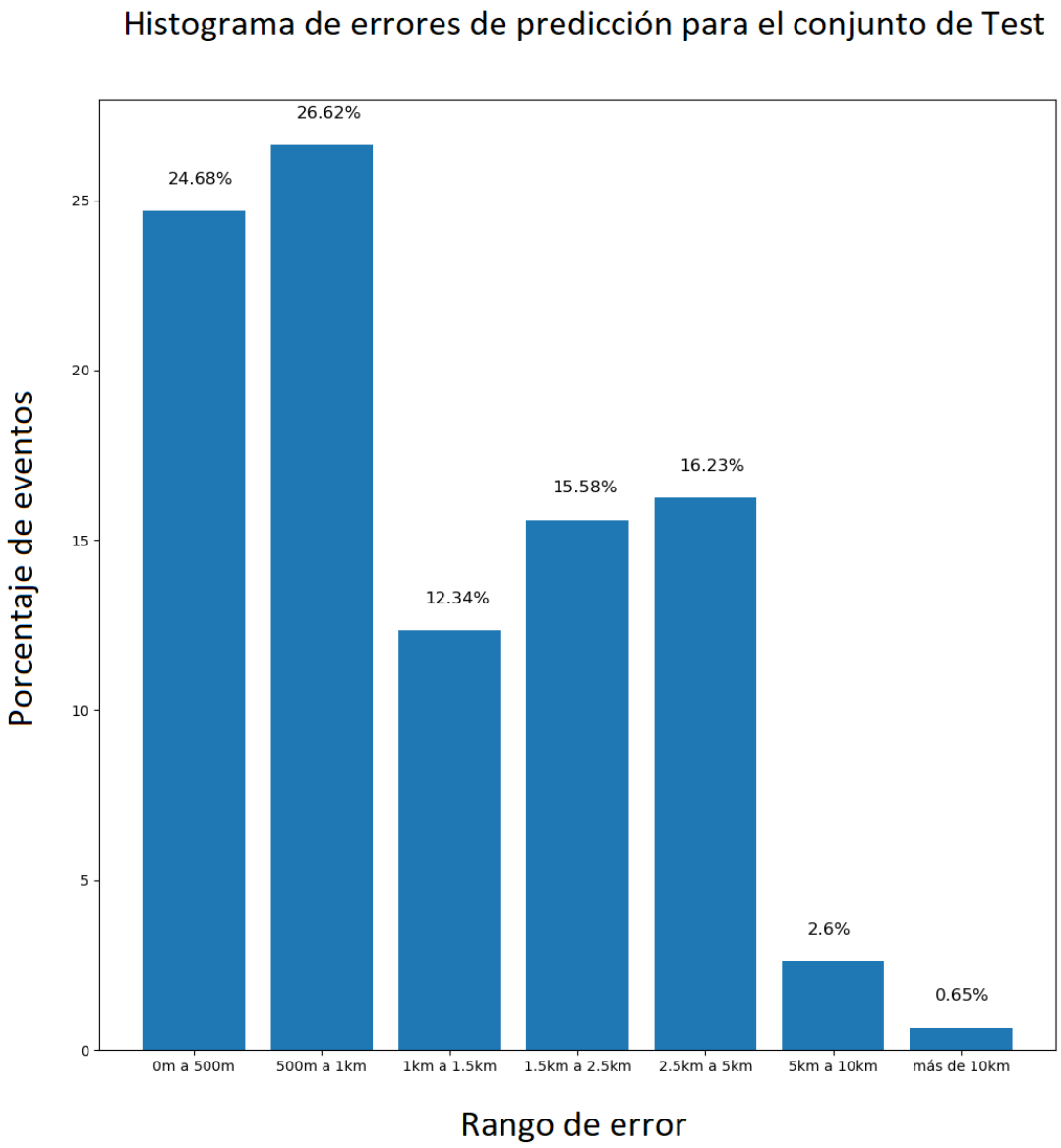

Figura 4.3: Histograma del error de estimación de la red neuronal para el conjunto de test utilizando el primero método de *Data Augmentation*

En la figura 4.4 se muestra el mapa marcando en rojo la *ground truth* y en cruces azules las estimaciones que entrega la red.

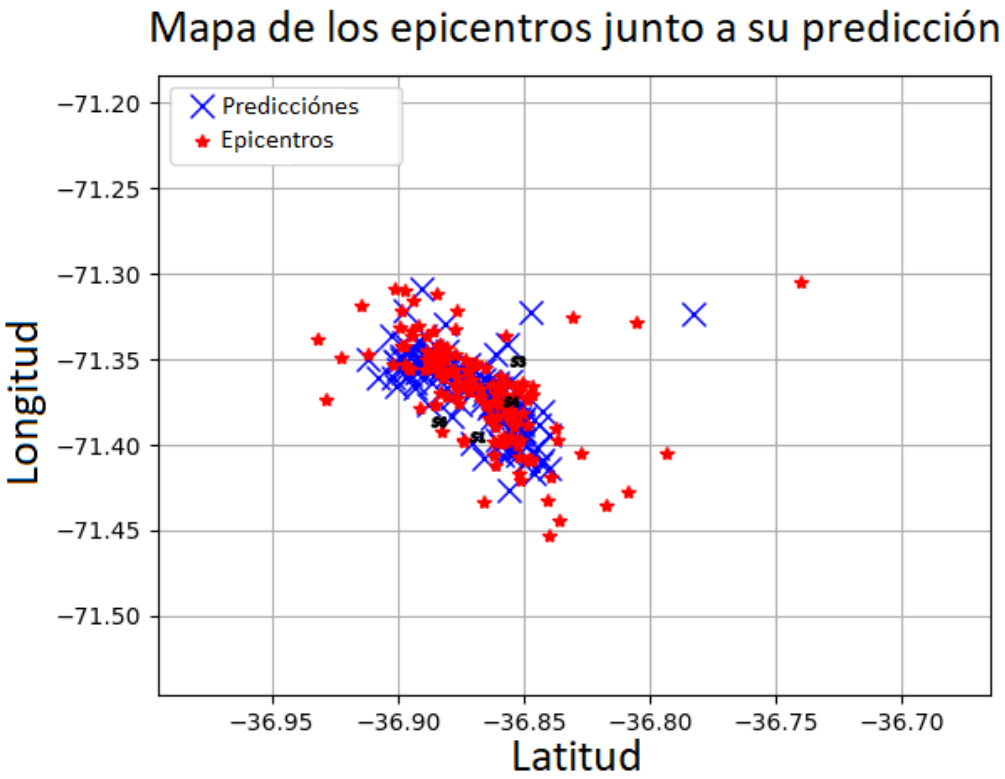

Figura 4.4: Mapa con las localizaciones de los epicentros de test y sus estimaciones al utilizar el primer método de *Data Augmentation*

Nuevamente, se muestran los gráficos obtenidos con la primera realización del entrenamiento. Con este método de *Data Augmentation* se obtiene un mayor porcentaje de eventos con error menor al kilómetro.

4.2.2. Segundo Método

A continuación, se presentan los mejores resultados con el segundo método de *Data Augmentation*, método con el cual se toma en consideración la distribución de epicentros con respecto a la distancia del centroide de referencia. Con este método se tiene un total de 3508 eventos de entrenamiento. El set de valores que entregó el mejor resultado se muestra en la tabla 4.3

Tabla 4.3: Set de parámetros e hiperparámetros que dan el mejor resultado utilizando el segundo método de *Data Augmentation*

Hiper/parámetro	**Utilizados**
Ancho de ventanas	500 [ms]
Traslape entre ventanas	60 %
Número de bins obtenido con DFT	31
Número de épocas	250
Taza de aprendizaje	0,00083
Optimizador	ADAM
Métrica a optimizar	MAE
Capas LSTM	4
Función de activación (LSTM)	Relu
Nodos por capa LSTM	512
Capas *Fully connected*	2
Función de activación (*Fully Connected*)	Relu
Nodos por capa *Fully Conncted*	64

Con los valores mostrados en la tabla 4.3 se llegó a un error promedio de 1,64 [km] con desviación estándar de 1,63 [km]. Se logró que un 43,07 % de los eventos en la base de datos de test tengan error de localización menor a 1 [km].

La arquitectura de la red se compone de inicialmente 4 capas ocultas de LSTM a lo que siguen 2 capas ocultas de *Fully Connected* con un *dropout* del 10 % en cada una y finalmente la capa de salida que tan solo tiene 2 nodos para predecir la latitud y longitud del epicentro.

En la figura 4.5 se muestra un histograma obtenido al evaluar el rendimiento de la red sobre el conjunto de test.

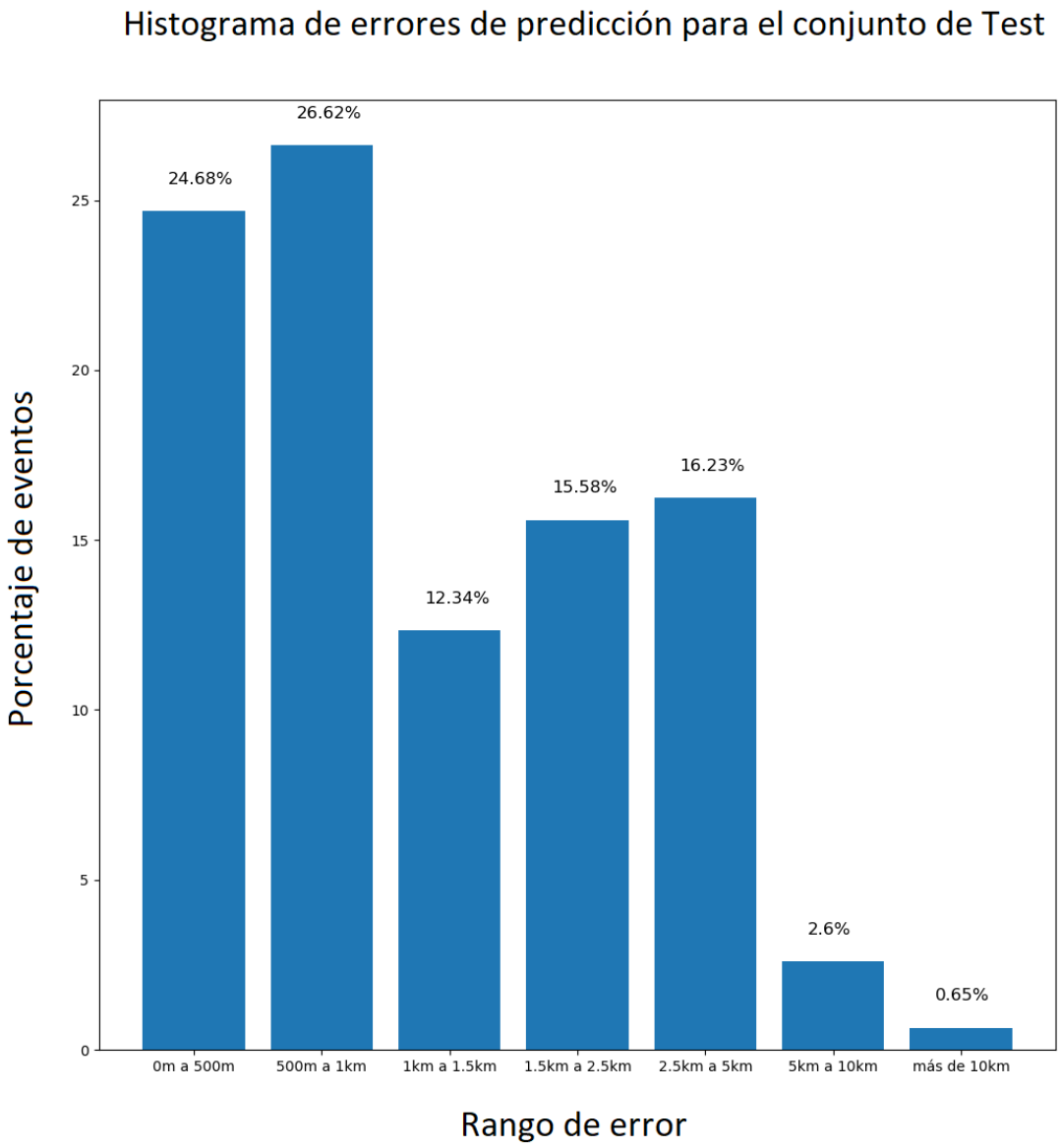

Figura 4.5: Histograma del error de estimación de la red neuronal para el conjunto de test utilizando el segundo método de *Data Augmentation*

En la figura 4.6 se muestra el mapa marcando en rojo la *ground truth* y en cruces azules las estimaciones que entrega la red.

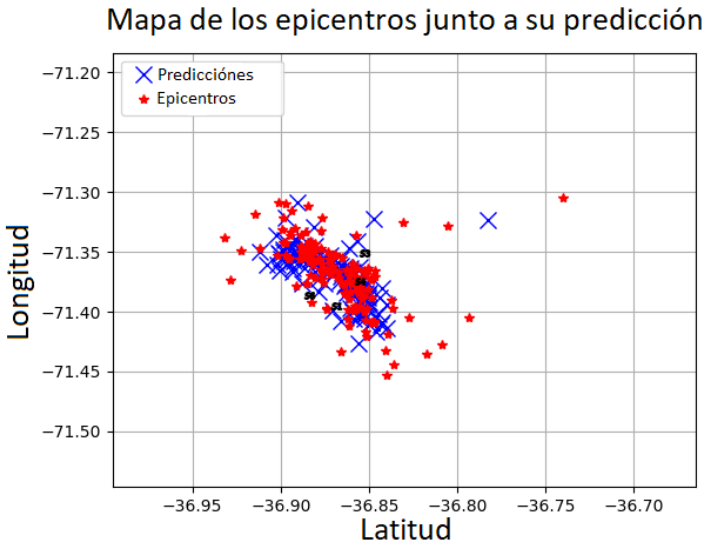

Figura 4.6: Mapa con las localizaciones de los epicentros de test y sus estimaciones al utilizar el segundo método de *Data Augmentation*

Nuevamente, se muestran los gráficos obtenidos con la primera realización del entrenamiento. El segundo método de *Data Augmentation* obtiene el peor porcentaje de eventos con error menor al kilómetro.

4.3. Análisis y discusión de resultados

A continuación se presenta la tabla 4.4 resumiendo los resultados obtenidos. En esta se puede concluir que el mejor resultado se obtiene utilizando el primer método de *Data Augmentation*, ya que, obtiene el mayor porcentaje de eventos con error menor al kilómetro y, además, tiene el menor error promedio, aún así, la desviación estándar del error es la mayor de los 3 casos.

Tabla 4.4: Resumen de los resultados obtenidos con cada método

	Error menor a 1km [%]	Promedio [km]	Desviación estándar [km]
Sin *D.A.*	48,48	1,51	1,56
Método 1 de *D.A.*	51,51	1,42	1,69
Método 2 de *D.A.*	43,07	1,64	1,63

La figura 4.7 fue obtenida para el método sin *Data Augmentation*, en esta se muestran los porcentajes de estimación que tienen error menor al kilómetro en función de la densidad de datos (tomando como punto de partida el centroide).

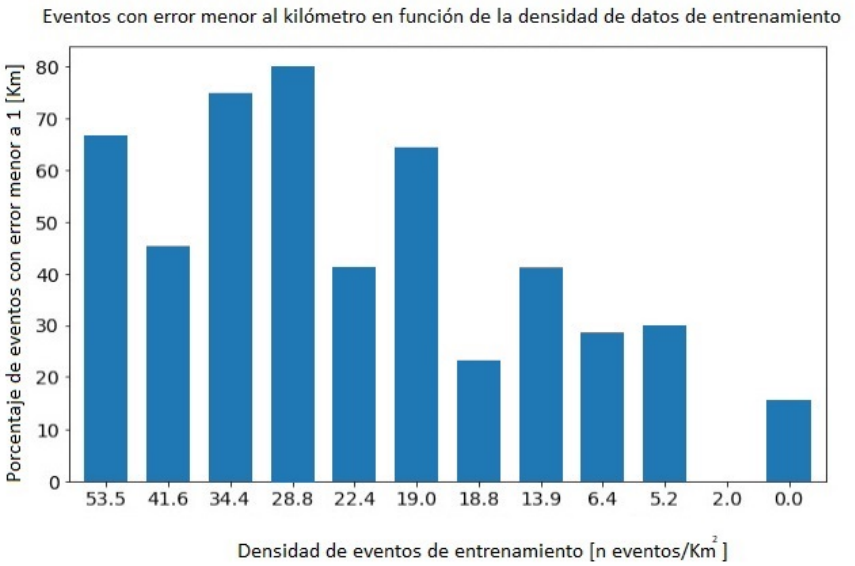

Figura 4.7: Histograma para representar la tasa de eventos con error menor al kilometro en el conjunto de test en función de la densidad de eventos del conjunto de entrenamiento

La figura 4.8 también fue obtenida para el método sin *Data Augmentation*. En esta se muestran los porcentajes de estimación que tienen error menor al kilómetro en función de la distancia al centroide como también se muestra el porcentaje de datos de entrenamiento contenidos en cada rango.

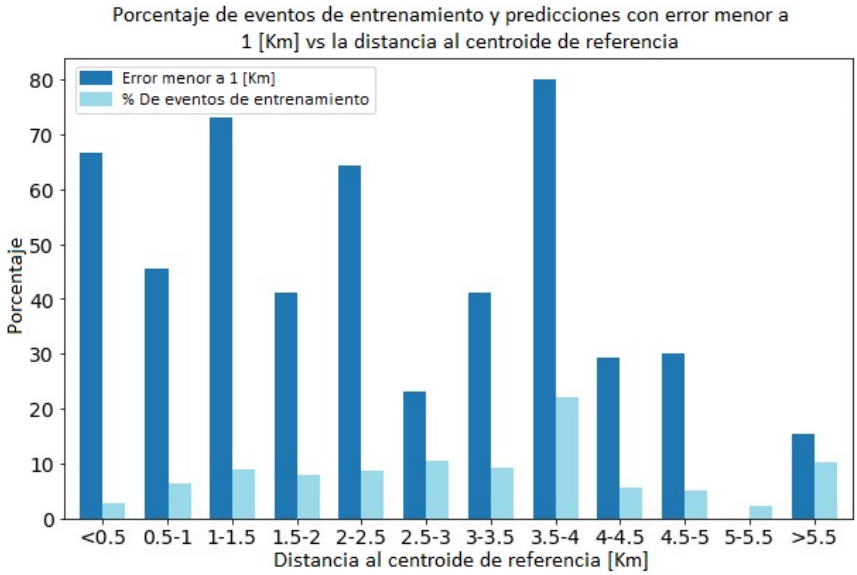

Figura 4.8: Histograma para representar la tasa de eventos con error menor al kilómetro en el conjunto de test en función de la distancia de los eventos al centroide de referencia, también la cantidad de eventos de entrenamiento para dichos rangos

Si bien los resultados mostrados en las figuras 4.1, 4.3 y 4.5 permiten apreciar en porcentajes los rangos de acierto de la red, no se está evaluando cómo afectan los datos de

entrenamiento a la calidad de las estimaciones.

Observando la figura 4.7 se puede apreciar que la precision de las estimaciones va disminuyendo al disminuir la densidad de eventos con respecto al centroide. La única excepción a esta tendencia se muestra en la figura 4.8 donde los eventos localizados entre 3,5 a 4 [km] obtienen la mayor tasa de errores menores al kilómetro. De acuerdo a la figura 4.8 para un radio menor de 2.5 [km] al centroide de referencia hay aproximadamente un 40 % de los eventos del conjunto de entrenamiento. En otras palabras, el 40 % de los datos de entrenamiento se encuentran en un área de 19,6 [km^2]. Entre 2,5 [km] y 5,5 [km] hay un 50 % aproximadamente de los datos de entrenamiento en una area de 75,5 [km^2]. Esto corresponde a una densidad de eventos de entrenamiento mucho menor que en el radio de 2,5 [km].

En el rango de 3,5 a 4 [km] de distancia al centroide de referencia hay una concentración mayor de eventos de entrenamiento, que corresponden a, aproximadamente, el 20 % del conjunto de entrenamiento. De forma consistente la precisión de las estimaciones entre este rango de distancia es tan alta como la de los eventos cercanos al centroide. Este resultado sugiere que la precisión de la red depende de la densidad de los datos de entrenamiento.

En la figura 4.7 se muestra una tendencia: mientras menor es la densidad de datos de entrenamiento menor es la tasa de éxito. Esto es aceptable considerando el hecho de que, en general, la capacidad de generalización de las redes neuronales depende de la cantidad de datos de entrenamiento. De aquí nace un interés en el uso de técnicas de aumento de data para mejorar la capacidad de generalización de la red. Aun así, observando los resultados de la tabla 4.4, se muestra que el mejor resultado se da utilizando el primer método de *Data Augmentation*, en donde se multiplicó la base de entrenamiento sin la intención de homogeneizar la distribución de eventos.

De la misma forma que se obtuvo la figura 4.8, se obtienen las figuras 4.9 y 4.10, pero utilizando la base de datos de entrenamiento para el método 1 y 2 con *Data Augmentation*, correspondientemente.

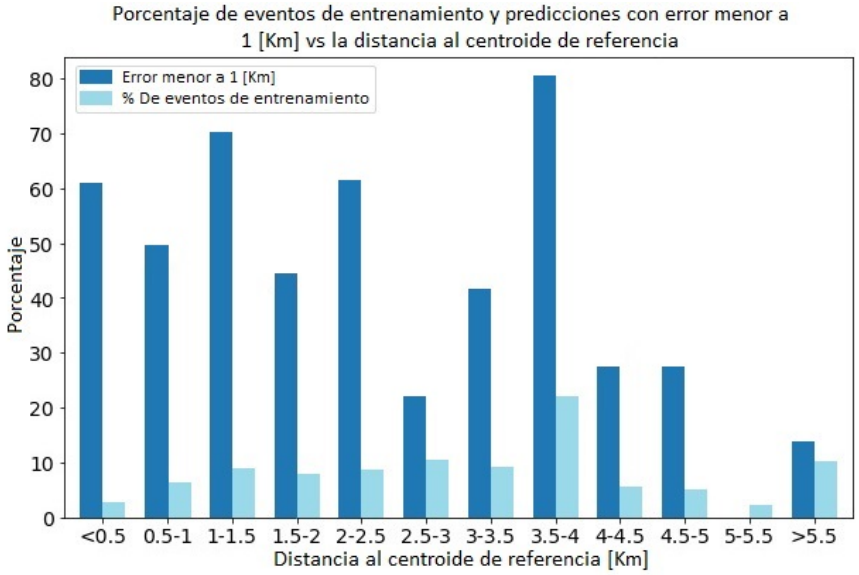

Figura 4.9: Histograma para representar la tasa de eventos con error menor al kilómetro en el conjunto de test en función de la distancia de los eventos al centroide de referencia, también la cantidad de eventos de entrenamiento para dichos rangos, utilizando el método 1 de *Data Augmentation*

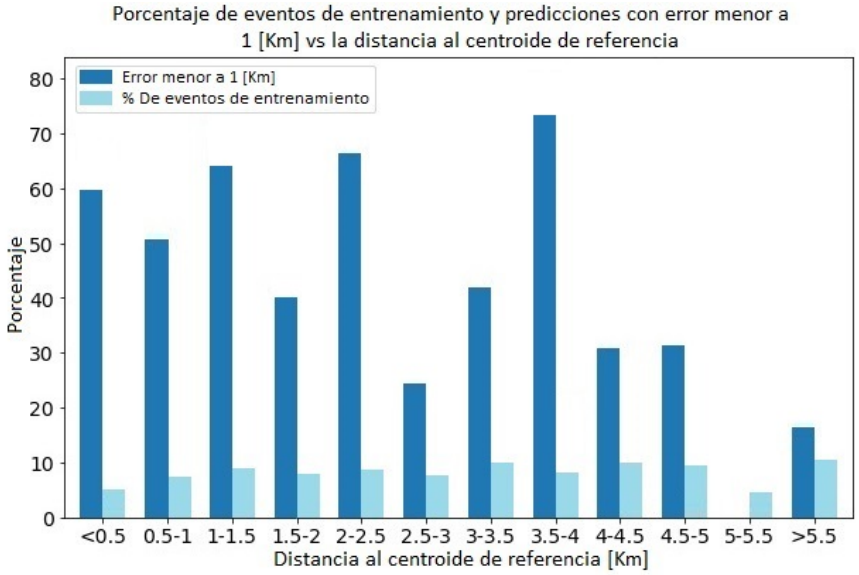

Figura 4.10: Histograma para representar la tasa de eventos con error menor al kilómetro en el conjunto de test en función de la distancia de los eventos al centroide de referencia, también la cantidad de eventos de entrenamiento para dichos rangos, utilizando el método 2 de *Data Augmentation*

Observando las figuras 4.9 y 4.10, se pueden percibir leves variaciones en las barras azules con respecto a la figura 4.8, lo que indica que no varió mucho la tasa de error por rangos, aun cuando en la figura 4.10 se nota claramente que los porcentajes de eventos de entrenamiento están distribuidos de manera más uniforme debido al método de *Data Augmentation* empleado.

Esto deja abierta la discusión respecto a qué método de *Data Augmentation* se debe emplear. La falla en el método 2 se puede deber a que los eventos en los rangos con baja densidad son de menor calidad con respecto a los de alta densidad, resultando en un aumento de la base de datos pero disminuyendo la calidad de los datos de entrenamiento. Otro punto a considerar, es el tamaño de la base de datos y la calidad de las señales generadas, pues, el primer método tiene aproximadamente 1000 eventos más que el segundo. Luego, si es que las señales con adición de ruido son un buen método para generar *Data Augmentation* entonces el multiplicar los eventos en los rangos de mayor densidad es provechoso para el entrenamiento y no deberían excluirse al generar datos.

Capítulo 5

Conclusiones

En este proyecto se han utilizado técnicas de *Deep Learning* junto con una aproximación *End to End* con el fin de afrontar un problema sin solución a la fecha, el cual es la localización automática de epicentros de origen volcánico utilizando la información de las señales sísmicas. Para esto, se tomó el caso del volcán Chillán, utilizando como base de datos sismos ocurridos en este desde el 2016 hasta el 2020. El fin último de este trabajo correspondía a localizar todas las señales del conjunto de test con error de estimación inferior al kilómetro. Esto no se logró alcanzar llegando tan solo a un 51,51 % de estos eventos, con un error promedio de estimación de 1,51 [km]. Aún así este es un logro que puede ayudar a futuros investigadores a continuar con proyectos de localización de epicentros.

Se escogieron las redes LSTM para realizar este proyecto, debido a su capacidad de capturar dinámicas en señales variantes en el tiempo, y de remover o añadir información en la memoria de los nodos de la red, de esta forma modelando dependencias en el tiempo, lo cual se estimó era muy conveniente para el desarrollo de este problema, pues, la información más importante para localizar los epicentros se obtiene al detectar el tiempo en que llegan las ondas P y S.

El uso de *Deep Learning* conlleva varios desafíos para ser empleado, además de todo el diseño de las redes neuronales que se debe realizar, no es del todo claro qué parámetros se deben barrer ni en qué rangos, por lo tanto, se requiere de un alto poder computacional y/o tiempo para poder realizar barridos de parámetros y así ir mejorando el rendimiento de los sistemas. Se debe tomar en cuenta que el preprocesamiento de la base de datos no es una tarea trivial y es de vital importancia, pues, un mal preprocesamiento nunca permitirá que la red obtenga buenos resultados. Para este caso en particular, la base de datos tenía que ser capaz de representar cómo variaba la energía de las señales sísmicas en el tiempo. Esto generaba limitaciones en cuanto al tamaño de las ventanas para separar la ondas, como al traslape que existe entre ellas, tomando así un *trade-off* entre la calidad de la información y el tiempo de entrenamiento de la red.

Una gran limitante de este proyecto es la escasez de datos, pues, es sabido que los sistemas de redes neuronales mejoran su rendimiento mientras más datos de entrenamiento representativos tienen. Esto quedó en el transcurso del proyecto cuando al hacer uso de técnicas de *Data Augmentation* mejoraron los resultados del sistema.

Finalmente, se concluye que es posible tener un sistema automático de estimación de

epicentros de origen volcánico, pero todavía existen grandes limitaciones en el sistema obtenido, pues, solo es aplicable en el volcán Chillán y tiene un error promedio de estimación de 1,51[km]. Además requiere de cortar manualmente las señales 2 segundos antes de la llegada de la onda P para poder procesar las señales e ingresarlas a la red. En base al funcionamiento de las redes neuronales, es esperable que con el paso de los años y, por consiguiente, un aumento en la cantidad de señales sísmicas catalogadas (ergo un aumento en la base de datos), la calidad de los resultados aumentará, debido a que se tendrán más datos de entrenamiento y la red tendrá mayor capacidad de generalización.

Bibliografía

[1] G. Vargas and S. Rebolledo, "Paleosismología de la falla san ramón e implicancias para el peligro sísmico de santiago," in *XIII Congreso Geológico Chileno, Antofagasta, Chile, Meeting Abstracts*, pp. 851–853, 2012.

[2] M. Herve, "Movimiento normal de la falla paposo, zona de falla atacama, en el mioceno, chile," *Andean Geology*, no. 31, pp. 31–36, 1987.

[3] D. T. Calle Gardella, "Tomografía sísmica pasiva 3d de estructuras corticales en el margen costero de chile central: Caracterización en profundidad de la falla de pichilemu," 2018.

[4] V. M. Zobin, *Introduction to volcanic seismology*, vol. 6. Elsevier, 2012.

[5] P. Gasparini, R. Scarpa, and K. Aki, *Volcanic seismology*, vol. 3. Springer Science & Business Media, 2012.

[6] S. Stein and M. Wysession, "An introduction to seismology," *Earthquakes, and Earth*, vol. 7, no. 9, p. 10, 2003.

[7] W. Zhu and G. C. Beroza, "Phasenet: a deep-neural-network-based seismic arrival-time picking method," *Geophysical Journal International*, vol. 216, no. 1, pp. 261–273, 2019.

[8] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.

[9] K. C. Tsai, W. Hu, X. Wu, J. Chen, and Z. Han, "First-break automatic picking with deep semisupervised learning neural network," in *SEG Technical Program Expanded Abstracts 2018*, pp. 2181–2185, Society of Exploration Geophysicists, 2018.

[10] J. Schieber, "Representación de ondas p y s," Ultimo acceso a esta: 12/07/2021. [Online]. Disponible en: https://geol105.sitehost.iu.edu/images/gaia_chapter_3/seismic.htm,.

[11] S. Stein and M. Wysession, *An introduction to seismology, earthquakes, and earth structure*. John Wiley & Sons, 2009.

[12] I. Gendler, "Transformada de fourier," Ultimo acceso a esta: 22/08/2021. [Online]. Disponible en: https://isaacscienceblog.com/2017/08/13/fourier-transform/,.

[13] J. R. del Solar, "Clase de redes neuronales artificiales," Ultimo acceso a esta: 28/05/2021. [Online]. Disponible solo para miembros de la institución: https://www.u-cursos.cl/ingenieria/2020/1/EL4106/1/material_docente/bajar?id_material=3121932,.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[16] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, PMLR, 2013.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] S. G. Mosher and P. Audet, "Automatic detection and location of seismic events from time-delay projection mapping and neural network classification," *Journal of Geophysical Research: Solid Earth*, vol. 125, no. 10, p. e2020JB019426, 2020.

[19] T. Perol, M. Gharbi, and M. Denolle, "Convolutional neural network for earthquake detection and location," *Science Advances*, vol. 4, no. 2, p. e1700578, 2018.

[20] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end asr-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[21] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6069–6073, IEEE, 2020.

[22] M. Zeineldeen, A. Zeyer, R. Schlüter, and H. Ney, "Layer-normalized lstm for hybrid-hmm and end-to-end asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7679–7683, IEEE, 2020.

[23] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, pp. 117–122, IEEE, 2018.

[24] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[25] Z. E. Ross and Y. Ben-Zion, "Automatic picking of direct p, s seismic phases and fault zone head waves," *Geophysical Journal International*, vol. 199, no. 1, pp. 368–381, 2014.

[26] I. Álvarez, L. García, S. Mota, G. Cortés, C. Benítez, and Á. De la Torre, "An automatic p-phase picking algorithm based on adaptive multiband processing," *IEEE Geoscience and remote sensing letters*, vol. 10, no. 6, pp. 1488–1492, 2013.

[27] F. Tan, H. Kao, E. Nissen, and D. Eaton, "Seismicity-scanning based on navigated automatic phase-picking," *Journal of Geophysical Research: Solid Earth*, vol. 124, no. 4, pp. 3802–3818, 2019.

[28] L. García, G. Alguacil, M. Titos, O. Cocina, A. De la Torre, and C. Benítez, "Automatic s-phase picking for volcano-tectonic earthquakes using spectral dissimilarity analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 874–878, 2019.

[29] X. Zhang, J. Zhang, C. Yuan, S. Liu, Z. Chen, and W. Li, "Locating induced earthquakes with a network of seismic stations in oklahoma via a deep learning method," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[30] M. Kriegerowski, G. M. Petersen, H. Vasyura-Bathke, and M. Ohrnberger, "A deep con-

volutional neural network for localization of clustered earthquakes based on multistation full waveforms," *Seismological Research Letters*, vol. 90, no. 2A, pp. 510–516, 2019.

[31] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[32] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[33] A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou, T. Giannakopoulos, A. Katsamanis, A. Potamianos, and S. Narayanan, "Data augmentation using gans for speech emotion recognition.," in *Interspeech*, pp. 171–175, 2019.

[34] J. Zhang, F. Wu, B. Wei, Q. Zhang, H. Huang, S. W. Shah, and J. Cheng, "Data augmentation and dense-lstm for human activity recognition using wifi signal," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4628–4641, 2021.