



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

METHODS OF TIME SERIES DATA MINING

DISEÑO E IMPLEMENTACIÓN DE UN DETECTOR DE ACTIVIDAD VOLCÁNICA DEL SANGAY

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE FÍSICO**

JORGE ANDRES MEDINA MOREIRA

jorge.medina@epn.edu.ec

DIRECTOR: RAMON XULVI-BRUNET

ramon.xulvi@epn.edu.ec

DMQ, SEPTIEMBRE 2022

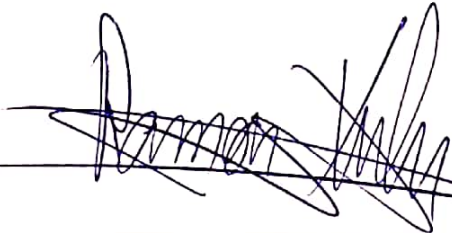
CERTIFICACIONES

Yo, JORGE ANDRES MEDINA MOREIRA, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



JORGE ANDRES MEDINA MOREIRA

Certifico que el presente trabajo de integración curricular fue desarrollado por JORGE ANDRES MEDINA MOREIRA, bajo mi supervisión.



Ramon Xulvi-Brunet
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JORGE ANDRES MEDINA MOREIRA

Ramon Xulvi-Brunet

DEDICATORIA

Para mis abuelas y abuelos que son y serán por siempre parte de mí.

Para mi ñaña Alejandra y para mi Vale que iluminan mi vida.

Para la música.

AGRADECIMIENTO

Gracias a mis padres por apoyarme y educarme durante toda mi vida. Porque siempre tengo lecciones que aprender de ellos y nada de esto hubiese sido posible sin su amor. A mi tía Daniela por enseñarme a siempre hacer lo que ame y a amarme.

A Ismael, Christian, Marcos, Daniel y Luis, por su amistad y hacer que estos años, y la vida, valgan la pena.

A Ramon, por educarme y guiarme desde el primer semestre en Física hasta este trabajo final. Porque sin su instrucción no hubiese aprendido lo que realmente es un físico y lo que es la ciencia. A Eddy, por ser mi primer maestro.

A Stephen Hernández y Mario Ruíz del Instituto Geofísico, porque esta investigación no hubiese sido posible sin ellos.

Índice general

1. Descripción del componente desarrollado	1
1.1. Marco teórico	1
1.2. Objetivo general	6
1.3. Objetivos específicos	6
1.4. Alcance	6
2. Metodología	9
2.1. Preprocesamiento de datos	10
2.2. Análisis de Componentes Principales	13
2.3. Autoencoders	14
2.3.1. Inteligencia Artificial	15
2.3.2. Implementación de Autoencoders	20
2.3.3. Autoencoder en el espacio de tiempo	21
2.3.4. Autoencoder en el espacio de frecuencias	23
2.4. Extracción de estadísticos	26
2.5. Clasificador binario	27
2.5.1. Máquinas de Soporte Vectorial	28
2.5.2. Validación cruzada	29
2.5.3. Evaluación del clasificador	30

3. Resultados, conclusiones y recomendaciones	32
3.1. Resultados	32
3.1.1. Métodos fallidos	32
3.1.2. Métodos exitosos	33
3.2. Conclusiones y recomendaciones	40
A. Anexos	43
A.1. Preprocesamiento de series temporales	43
A.2. Aplicación de PCA y Autocorrelación	43
A.3. Dimensión espacio latente de los autoencoders	43
A.4. Entrenamiento de autoencoders	43
A.5. Extracción de PSD (Método de Welch)	44
A.6. Extracción de estadísticos con los datos originales y reconstruidos	44
A.7. Entrenamiento de clasificador binario	44
Bibliografía	45

Índice de figuras

2.1. Arquitectura del clasificador binario de eventos del volcán Sangay.	9
2.2. Neurona artificial con un vector x como entrada y salida o [50].	15
2.3. Ejemplos de funciones de activación [52].	16
2.4. Red neuronal densa [54].	16
2.5. Error en el entrenamiento en función del número de <i>epochs</i> . Azul: conjunto de entrenamiento, Naranja: conjunto de validación.	19
2.6. Arquitectura de un autoencoder [72].	20
2.7. Muestra de actividad del Sangay. Izquierda: serie temporal preprocesada (unidades arbitrarias); derecha: densidad de potencia espectral calculada con el método de Welch (unidades arbitrarias).	24
2.8. Kernel radial aplicado en una Máquina de Soporte Vectorial para clasificación binaria [85].	28
2.9. Matriz de confusión para clasificación binaria [89].	30
3.1. Ejemplos de autocorrelación para series temporales en dirección Este para eventos y no eventos del Sangay en la estación BULB.	33

3.2. Precisión en función de la dimensión de la capa latente para la estación PUYO.	34
3.3. Error cuadrático medio para el conjunto de entrenamiento y validación para los autoencoders de la estación PKYU. Izquierda: series temporales; derecha: densidad de potencia espectral.	34
3.4. Ejemplo de series temporales originales y reconstruidas por el autoencoder para la estación PKYU. Izquierda: evento del volcán Sangay; derecha: no evento del volcán Sangay.	35
3.5. Ejemplo de PSDs originales y reconstruidas por el autoencoder de frecuencias en la estación PKYU. Izquierda: evento del volcán Sangay; derecha: no evento del volcán Sangay.	36
3.6. Histograma bidimensional de calor para estadísticos entre eventos y no eventos del Sangay en la estación BMAS.	37
3.7. Exactitud del detector entrenado con los datos de la capa latente y estadísticos (verde), sólo estadísticos (azul) y sólo de la capa latente (rojo).	38
3.8. Histograma relativo de la probabilidad de clasificación para eventos (azul) y no eventos (rojo) del Sangay en la estación PUYO. Las líneas segmentadas representan el percentil 5 de los datos.	39
3.9. Tasa de eventos y no eventos del Sangay clasificados correctamente con una probabilidad mayor al 99% para las estaciones.	39
3.10 Exactitud, precisión y exhaustividad (recall) en el detector entrenado para las estaciones.	40

RESUMEN

El volcán Sangay es uno de los más activos del Ecuador. Debido a la caída esporádica de ceniza y las condiciones climáticas, es difícil mantener un monitoreo sismo-acústico continuo utilizando la estación sísmica más próxima al volcán. En este estudio, evaluamos si una red regional puede ayudar a llenar los vacíos de observación cuando no se dispone de medidas de la estación más cercana. Con este fin, diseñamos e implementamos un detector binario utilizando señales de sismómetros. Se tomó un conjunto de tiempos generado previamente, correspondientes a eventos y no eventos etiquetados del Sangay entre 2017 y 2020. El detector se implementó independientemente para siete estaciones ubicadas a más de 50 km de la cumbre del Sangay, tomando las dos componentes horizontales (Norte y Este) del sismómetro. Para reducir el número de datos, se entrenaron dos autoencoders en cada estación con los eventos del Sangay: uno con la serie temporal y el otro con la densidad de potencia espectral. Encontramos que 10 dimensiones son suficientes para reconstruir adecuadamente los datos. Se entrenó un clasificador de máquina de soporte vectorial (SVM) en cada estación para la detección de actividad utilizando los datos de los autoencoders y otros indicadores estadísticos, obteniendo una exactitud de clasificación mayor al 98% en todas las estaciones. Finalmente, se asignó la probabilidad de cuando un registro corresponde o no a un evento sísmico del Sangay con el método de validación cruzada de 5 iteraciones.

Palabras clave: volcán Sangay, autoencoder, máquinas de soporte vectorial, detección volcánica.

ABSTRACT

The Sangay is one of the most active volcanoes in Ecuador. Due to sporadic ash fall and climatic conditions, it is difficult to maintain continuous seismo-acoustic monitoring using the closest seismic station to the volcano. In this study, we assess if a regional network can help fill in the observational gaps when nearest station measurements are not available. To this end, we design and implement a binary detector using signals from seismometers. A previously-generated dataset of times corresponding to labeled events and non-events from Sangay were taken between 2017 and 2020. The detector was implemented independently for seven stations located more than 50 km from the Sangay's summit, taking the two horizontal components (North and East) of the seismometer. To reduce the number of data points, two autoencoders were trained at each station with the Sangay events: one with the time series and the other with the spectral power density. We find that 10 dimensions are sufficient to adequately reconstruct the input data. A support vector machine (SVM) classifier was trained for each station for activity detection using the data from the autoencoders and other statistical indicators, obtaining a detection accuracy greater than 98% at all stations. Finally, we assigned the probability of whether or not the measurement in a station correspond to an event from the Sangay using 5-fold cross-validation.

Keywords: Sangay volcano, autoencoder, support vector machine, volcanic detection.

Capítulo 1

Descripción del componente desarrollado

1.1. Marco teórico

El volcán Sangay es el más activo del Ecuador [1]. Entre las amenazas asociadas a éste están los flujos de lava, emisión de gases volcánicos y las emisiones y caídas de piroclastos. Por ese motivo es importante tener un registro continuo de la actividad del volcán [1]. A pesar de esto, no existen muchos estudios en el Sangay debido a las extremas condiciones ambientales y el bajo riesgo hacia las poblaciones humanas [2]. El Instituto Geofísico de la Escuela Politécnica Nacional (IG-EPN) es el organismo geológico más importante del Ecuador, y usa diferentes instrumentos para monitorear al Sangay. Entre los métodos de detección están los sismómetros de banda ancha, micrófonos que registran señales acústicas, así como en infrasonido, y sensores de emisión de gases volcánicos y monitoreo satelital [2].

Los sismómetros de banda ancha son instrumentos que registran el movimiento del suelo, en una dirección, en un rango amplio de frecuencias (0.01 a 100 Hz [3]). Los sismómetros modernos miden el desplazamiento del suelo en función del tiempo mediante señales que son almacenadas en bases de datos para su procesamiento y análisis. Estas señales son series temporales, es decir, se tratan de una colección de escalares indexados cronológicamente con el tiempo de la detección [4]. Cuando un

sismómetro detecta el movimiento en más de una dirección (componente) es conocido como un sismómetro multicanal [5]. Estos, generalmente, registran el movimiento del suelo en dirección Vertical, Este y Norte, donde estas dos últimas son conocidas como las componentes horizontales. En los sismómetros multicanales, por lo tanto, se registra una serie temporal por cada componente.

En particular, la Red Nacional de Sismógrafos (RENSIG) del IG-EPN es la red regional más importante del Ecuador para la detección de actividad sísmica. Los sismógrafos ubicados en las estaciones de la RENSIG registran eventos como terremotos, erupciones volcánicas, ruido de las ciudades o explosiones, todo en forma de series temporales almacenadas en la base de datos del IG-EPN.

Como las vibraciones a través del suelo son atenuadas en función de la distancia entre la fuente y el punto donde se realiza la medición, es importante tener sismómetros cerca de lugares con potencial movimiento sísmico [6]. Los eventos volcánicos generan ondas sísmicas que se transmiten a través de la corteza terrestre, por lo que los sismómetros cercanos a un volcán registran más fuertemente la actividad del mismo.

De esta forma, una red local de un volcán es un conjunto de instrumentos de detección cercanos al mismo (a pocos kilómetros de la cumbre) que permite la detección continua de actividad. En el caso del volcán Sangay, debido a la caída esporádica de ceniza y a la frecuente nubosidad es difícil mantener un monitoreo sísmico local.

La mayoría de investigaciones enfocadas en el Sangay han sido realizadas sobre señales acústicas y sísmicas de emisión de gases a 2 km de la cumbre del volcán [7-9]. También se ha estudiado la actividad del Sangay por infrasonido entre el 2006 y 2012 [10], así como la detección de emisión de dióxido de azufre (SO_2) mediante sensores satelitales [11]. De igual forma se ha estudiado la deformación en el suelo del volcán mediante imágenes de radar obtenidas por satélite [12].

Uno de los estudios más exhaustivos sobre la actividad del Sangay, durante las últimas dos décadas, usó datos de diez estaciones sísmicas de la RENSIG, sensores acústicos y de gases volcánicos, así como observaciones satelitales [2]. En esta se encontró que al observar el espectro-

grama, representación visual de cómo el espectro de frecuencias de una señal varía en el tiempo, de eventos provenientes del Sangay, la distancia de las estaciones lejanas al volcán hace que los eventos registrados en los sismómetros se atenúen y tengan una baja magnitud, por lo que hay la menor tasa de ruido-signal en una banda de $0.6 - 1.2 \text{ Hz}$ [2]. Es decir, en este intervalo de frecuencias se tiene la mayor cantidad de información registrada sobre el volcán Sangay en comparación al ruido de otros eventos (como ruido de terremotos o ruido ambiental).

Con los datos de las estaciones sísmicas lejanas, Vásquez et al. [2] implementaron un detector de actividad volcánica capaz de indicar si hubo o no un evento en función de la señal registrada en diez sismómetros de la RENSIG. En ese estudio se usaban los datos de todas las estaciones a la vez (componente vertical) para llegar a una tasa de identificación de verdaderos positivos mayor al 97% (tasa de eventos del Sangay detectados correctamente) [2]. Para eventos provenientes del volcán hay diferente intensidad de la función de onda registrada en el sismómetro y diferente tiempo de arribo de la señal, esto se da debido a las distintas distancias de la cumbre a los sismómetros.

La RENSIG posee una estación cercana al Sangay, estación SAGA, cuyo propósito es estudiar al volcán como una red local de monitoreo. Se vuelve difícil darle una calibración continua al sismómetro debido a las extremas condiciones climáticas, por lo que el registro de eventos volcánicos en esa estación es intermitente. Por eso, es importante la detección de actividad en el volcán utilizando sismómetros lejanos de una red regional, RENSIG, a pesar de la atenuación de la magnitud de la señal y el aumento de ruido proveniente de otras fuentes.

Por todo lo expuesto anteriormente, se plantea diseñar e implementar un detector capaz de determinar si hay o no actividad volcánica del Sangay usando las componentes horizontales registradas por sismómetros multicanales de banda ancha de estaciones lejanas de la RENSIG a pesar de que estén a más de 50 km de la cumbre del volcán.

A diferencia de la investigación de Vásquez et al [2], implementaremos el detector con los datos de cada estación, individualmente, y con las componentes horizontales del sismógrafo con el objetivo de obtener una precisión de detección similar y conocer si es posible detectar actividad

volcánica con una única estación.

En un modelo físico siempre se debe entender lo que está pasando en el sistema (el cómo funciona) y la capacidad de predicción a futuro. En un detector sólo se busca cumplir con lo segundo, es decir, ser capaces de poder predecir si se trató o no de un evento del Sangay cuando llega una señal sísmica al sensor (sismómetro).

Como físicos estamos interesados en construir un modelo capaz de entender el cuándo un evento proviene del Sangay o no. Dado que las series temporales poseen muchos datos en una muestra (para un evento), es complicado diferenciar entre una que proviene del Sangay de una que no con sólo observar la función de onda. Por lo tanto, se deben transformar las series temporales registradas por los sismógrafos a un conjunto de datos lo suficientemente *buenos* para detectar un evento volcánico.

En este trabajo se considera al número de datos como el valor que toman los registros indexados cronológicamente en una serie temporal, ya que a cada instante de la señal ese valor puede diferir para una muestra u otra. Por ejemplo, si tenemos un sensor que mide la posición de una masa acoplada a un resorte cada segundo durante un minuto se tendrán 60 datos en la muestra (serie temporal). El sismómetro actúa de la misma forma al registrar el movimiento del suelo en una serie temporal a cierta tasa de muestreo (datos medidos por segundo) durante una ventana de actividad (duración de la serie temporal).

Nosotros partimos de una señal con cientos de datos. Para construir un modelo físico debemos transformar todos esos datos a un conjunto pequeño de variables, y que estas sean entendibles (interpretación física), ya que así podríamos construir un modelo tanto para predecir como entender la actividad volcánica en el Sangay.

Por consiguiente, las variables que obtenemos transformando la serie temporal deberían ser capaces de determinar si tenemos o no un evento del Sangay en función de los valores que pueden tomar. Una lista de variables lo suficientemente pequeña, pero capaz de distinguir los dos tipos de series (evento o no evento), es lo ideal. Por esto, el detector de actividad volcánica que queremos construir estará basado en Máquinas de Soporte Vectorial (SVM) [13] con una probabilidad alta de ser o no ser

un evento en función de las variables.

El SVM es una técnica de inteligencia artificial capaz de separar las muestras en dos regiones del espacio de variables por una hipersuperficie, por lo que podemos detectar si se trata o no de un evento volcánico si estas son elegidas adecuadamente [14]. A breves rasgos, un clasificador binario (sí o no) funciona como un detector. Cada punto corresponde a unos valores particulares de las variables del modelo con el que esperamos caracterizar de forma adecuada la serie temporal de datos que estamos tratando de entender.

Desde el punto de vista de los sistemas complejos no entendemos la respuesta del sistema, nuestra serie temporal, hasta no tener un modelo cuantitativo (basado en un conjunto de variables) suficientemente adecuado que prediga y explique la actividad en el Sangay. De igual forma, esas variables que definan nuestro modelo deben captar las características claves en las funciones de onda provenientes del volcán. Esas variables no son los datos que registramos en una serie temporal, es lo que obtenemos al transformarla.

Como no sabemos muy bien cómo encontrar esas variables, usaremos redes neuronales para sugerir un número o tipo de ellas. En particular, usaremos una red neuronal de tipo autoencoder que funciona como un codificador y decodificador que transforma no linealmente los datos de una muestra a un menor número y después intenta reconstruir la serie temporal original desde ese conjunto mínimo de datos (conocidos como datos latentes) [15, 16].

El problema del autoencoder, y en general, con las técnicas de *deep learning* (redes neuronales) [17], es que tienen una baja interpretabilidad física al funcionar como una caja negra [18]. En el contexto de las redes neuronales, lo que para nosotros, los físicos, son datos, se los conoce como *variables*. Este *abuso* de notación se da porque las neuronas pueden tener una salida *variable* en función de las entradas, de allí su nombre.

La red neuronal es capaz de *captar* qué tipo de transformación a la serie temporal hace que podamos distinguir de si se trata o no un evento volcánico. La cuestión es que estos datos encontrados por el autoencoder son a través de cientos de miles de transformaciones no lineales de la

serie temporal original, por lo que no podemos encontrarle una interpretabilidad. Por esto, se debe extraer información de estos datos encontrados por la red neuronal para inspirar indirectamente a encontrar nuevas variables físicas capaces de construir un nuevo modelo para la detección de actividad.

En este trabajo, por tanto, nos centramos en predecir, no entender. No buscaremos cuáles son las variables físicas que podrían caracterizar los fenómenos del Sangay debido a la limitación de tiempo, pero aplicaremos autoencoders para encontrar un número mínimo de datos (variables en el contexto de redes neuronales) que sean lo suficientemente buenos como para determinar de si se trata o no de un evento volcánico. A partir de este modelo de *red neuronal* se podría construir otro más físico que nos permita, además de predecir, entender el sistema.

1.2. Objetivo general

Desarrollar un método de detección binaria de actividad volcánica partiendo de las componentes horizontales medidas por un sismógrafo cualquiera de la RENSIG.

1.3. Objetivos específicos

1. Extraer y preprocesar las series temporales de las componentes horizontales en estaciones de la RENSIG lejanas al volcán.
2. Transformar las series temporales a un menor número de datos por muestra.
3. Implementar un detector binario de actividad volcánica en cada estación con los datos procesados y transformados.

1.4. Alcance

Existen muchas herramientas para el análisis de las series temporales e identificación de eventos volcánicos, como el procesamiento de las

señales en el espacio de frecuencia o tiempo [19-21], análisis estocástico [22] o modelamiento matemático [23]. Las Máquinas de Soporte Vectorial (SVM) separan áreas/volúmenes del espacio de variables mediante una hipersuperficie [24-26], de manera que si un punto pertenece o no a uno de los volúmenes podemos decir si la serie temporal caracterizada por los valores de las variables (del modelo), correspondientes a ese punto del espacio de variables, es un evento o no.

El detector se implementará en cada una de las siguientes siete estaciones de la RENSIG: PUYO, PKYU, TAIS, BPAT, BMAS, TAMH, PORT. Las estaciones se consideran lejanas dado que se encuentran de 50 a 110 *km* de la cumbre del volcán Sangay.

El IG-EPN registra el inicio de eventos de actividad volcánica del Sangay y de otros eventos ajenos al volcán en ficheros. Por lo que se usarán estos tiempos para recortar las series temporales correspondientes a las componentes horizontales (Norte y Este) del sismómetro de cada estación.

El Análisis de Componentes Principales [27, 28] (PCA), Análisis Factorial [29] o Descomposición en Componentes Principales [30] son métodos lineales comúnmente utilizados para reducir el número de datos. En particular, el PCA es bastante usado en el procesamiento de series temporales sismológicas [31, 32], por lo que en este trabajo se lo aplicará en los conjuntos de datos preprocesados.

En el PCA las variables son la dirección Este y Norte. Como las señales provenientes del Sangay deben venir con una cierta dirección en el plano Este/Norte, intentaremos usar PCA para encontrar la dirección en la que llega una señal para discriminar si se trata o no de un evento volcánico. De esta forma proyectaremos los datos de las series temporales Este y Norte en una única serie temporal en la dirección encontrada por el PCA. Además, probaremos la autocorrelación, similaridad de una serie temporal con el desfase temporal de la misma serie [33], para caracterizar la diferencia entre las series temporales provenientes y no provenientes del Sangay.

Encontraremos una lista de variables, lo más corta posible, que permita caracterizar las series temporales utilizando dos redes neuronales de tipo autoencoder [34-36], una para las series temporales y otra para su

espectro de frecuencias. Aplicaremos autoencoders en ambos espacios, aunque aumente el número de variables, con el propósito de mejorar la detección de eventos volcánicos del Sangay.

Los autoencoders son un tipo de red neuronal que transforman no linealmente una muestra de datos en un menor número de ellos [37-41] y se detallan a profundidad en el siguiente capítulo, al igual que las demás técnicas mencionadas en este apartado.

Con las series temporales transformadas a un menor número de datos, se implementará un clasificador binario capaz de predecir, o estimar, una salida de si hay o no actividad en el volcán en función de las entradas [42, 43].

El rendimiento del detector será evaluado para diferentes métricas como precisión o exactitud [44] de clasificación. Estas medirán qué tan bien el detector podrá clasificar eventos o no eventos del Sangay.

Capítulo 2

Metodología

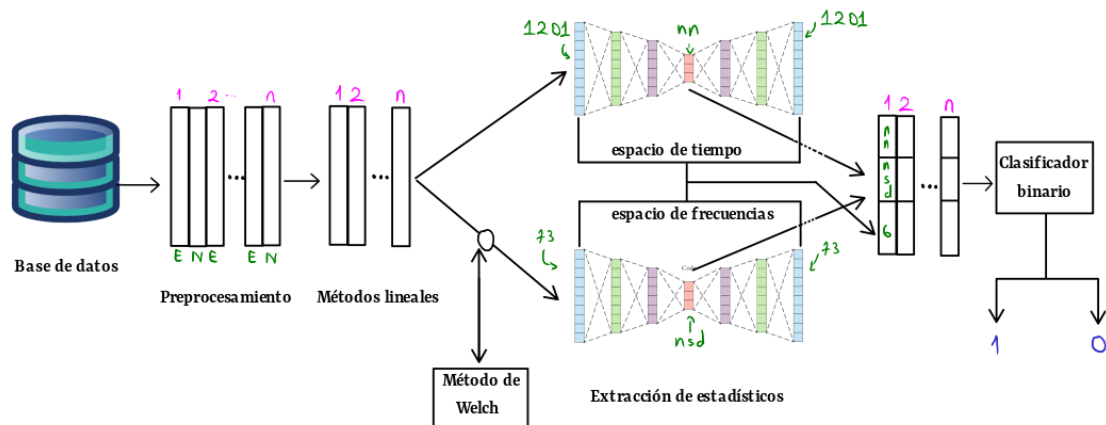


Figura 2.1: Arquitectura del clasificador binario de eventos del volcán Sangay.

El detector de actividad volcánica que pretendemos construir para cada estación sigue la arquitectura de la Figura 2.1. Los números verdes indican el número de datos, escalares registrados en las series temporales, y los números morados corresponden al número de la muestra. La salida del clasificador binario (número de color azul) es 1 si se trata de un evento o 0 si es un no evento del volcán. Todos los pasos de la arquitectura del clasificador binario son detallados exhaustivamente en este capítulo.

Primero se realiza el preprocesamiento de las series temporales al ex-

traer las componentes horizontales, Este y Norte, de cada evento para la base de datos de una estación. Después se reduce el número de datos al obtenerse una sola serie temporal por cada evento. Seguido a esto está la fase de entrenamiento y extracción de estadísticos usando redes neuronales, de tipo autoencoder, en el dominio de tiempo y de frecuencias entrenados con eventos provenientes del Sangay.

Como los autoencoders serán entrenados con eventos del Sangay se espera que estos sean capaces de reconstruir mejor las series temporales correspondientes a eventos que un no evento, por lo que se usarán estadísticos como el error medio entre la señal original y reconstruida, tanto para la serie temporal y el espectro de frecuencias, con el fin de caracterizar si se tiene o no un evento del volcánico. Finalmente, se implementa un clasificador binario basado en una máquina de soporte vectorial (SVM).

2.1. Preprocesamiento de datos

Las series temporales almacenadas por los sismógrafos están en formato miniSEED, por lo que se usará Obspy [45, 46], una librería de código abierto utilizada por sismólogos en el entorno de programación de Python, para la extracción y preprocesamiento de los datos.

Los datos provistos por el IG-EPN corresponden a los registros diarios entre 2017 y el 2020 de las estaciones de la RENSIG: PUYO, PKYU, TAIS, BPAT, BMAS, TAMH y PORT. Como los sismógramos son multicanales, registran las componentes Norte (N), Este (E) y Vertical (Z). Para este trabajo sólo se tomarán las componentes horizontales (Este y Norte) de los sismómetros.

Los sismómetros registran las series temporales en diferentes tasas de muestreo porque no todas las estaciones tienen los mismos sensores. El almacenamiento de los datos se lo hace diariamente, y para que se diferencien el nombre de los ficheros entre estación, día, año, tasa de muestreo o componente, se guardan en el formato: EC.XXXX..YYA.D.YEAR.DAY, donde XXXX indica el nombre de la estación, YY representa la tasa de muestreo del sismómetro, de tal manera que si YY=HH entonces la tasa de muestro es de 100 *sps* (muestras por segundo) y si YY=BH entonces la

tasa de muestreo es de 50 *sps*. El tipo de componente es (A), tal que: E, N y Z son la componente Este, Norte y Vertical, respectivamente. Finalmente, YEAR y DAY son el año y día en formato juliano de la medición en el sismógrafo.

Para la extracción de los conjuntos de muestras de eventos y no eventos de actividad volcánica en el Sangay se leen ficheros con el nombre: XXXX_labelsY.UTC.dat, donde XXXX indica el año y Y corresponde a eventos (1) o no eventos (0) del volcán. Dentro de estos ficheros está registrado el tiempo, en formato UTC, de inicio de una señal durante todo el año. Entre el 2017 y 2020 se tiene un registro total de 25359 eventos y 12446 no eventos de actividad volcánica en el Sangay. Además, se tomarán ventanas de actividad de 120 *s*, es decir, las series temporales son recortadas con el tiempo de inicio señalado en el fichero hasta dos minutos después. Estas listas de tiempos son provistas por el Instituto Geofísico de la Escuela Politécnica Nacional.

Con la base de datos de las siete estaciones, y los ficheros de tiempos de inicio, se guarda un conjunto de series temporales con las dos componentes horizontales por cada año, estación y tipo de evento. El preprocesamiento se realizó mediante la librería [Obspy](#) de Python en un programa (Anexo 1) que ejecutó las siguientes actividades:

- En un triple bucle se extraen las series temporales Este y Norte en un fichero en formato binario como: `est_year_event.npy`, donde `est` es el nombre de la estación, `year` es el año y `event` es 1 o 0 según si son eventos o no del Sangay.
- Los diccionarios son estructuras de Python que permiten almacenar cualquier tipo de datos, como números o caracteres, asociados a una llave como identificador [47]. Así, el bucle lee los ficheros de tiempos de eventos y genera un diccionario de tal modo que cada llave corresponde a un día de actividad y dentro de ella están los tiempos de inicio de las señales.
- La función `trim` de `Obspy` permite recortar una serie temporal desde un tiempo de inicio y fin en formato UTC. Con el diccionario de días se realiza un bucle en donde se corta la serie temporal con el tiempo inicial del diccionario hasta 120 *s* después.

- Las funciones *detrend* y *taper* de Obspy eliminan la tendencia (lineal y media) y suavizan las series temporales al inicio y fin del registro, 10% de los datos, respectivamente. Estos procesos son usuales en la sismología y se los realiza para evitar la aparición de artificios en el análisis de las series temporales registradas por los sismómetros.
- Seguido a esto, se realiza un filtrado de frecuencias de tipo pasa banda entre $0.6 - 1.2 \text{ Hz}$ con la función *filter* de Obspy. En esta banda de frecuencias se tiene la menor tasa ruido-señal en los eventos detectados del volcán Sangay, por lo que se reduce el ruido proveniente de fuentes ajenas al Sangay.
- A continuación, se hace un remuestreo de las series temporales al reducir la tasa de muestreo de 100 sps (muestras por segundo) o 50 sps a 10 sps . Este se realiza con la función *decimate* de Obspy. El teorema de muestreo de Nyquist [48] señala que la frecuencia a la que se debe registrar una señal debe ser mayor o igual al doble de la frecuencia más alta medida en una serie temporal para evitar la pérdida de información. Como el filtro pasa banda sólo permite una frecuencia máxima de 1.2 Hz , el remuestreo debe realizarse a una tasa de al menos 2.4 sps . Este paso es importante porque permite reducir el número de datos registrados en la serie temporal sin representar una pérdida de información de los eventos.
- Mediante condicionales, el programa no almacena eventos que no fueron registrados por la estación, ni aquellos que estén incompletos (el sismógrafo tuvo intermitencias durante la ventana de actividad). Esto hace que el número de muestras de actividad y no actividad del volcán sean diferentes para cada estación.
- Finalmente, las series temporales preprocesadas tienen 1201 datos por componente. Estas son concatenadas, por columna, en una matriz según el orden de aparición del evento. Es decir, la primera y segunda columna contienen la componente Este y Norte de la primera muestra, la tercera y cuarta columna contienen la componente Este y Norte de la segunda muestra, y así consecutivamente. Por lo que la matriz será de $(1201 \times 2n)$ con n el número de muestras. Esta es

almacenada en formato binario para únicamente reducir el tamaño de almacenamiento.

2.2. Análisis de Componentes Principales

El Análisis de Componentes Principales (PCA) es una técnica comúnmente utilizada para la reducción de datos de series temporales sismológicas [31, 32]. Esta técnica la implementamos pero no obtuvo buenos resultados, como se muestra en el último capítulo de este trabajo, pero creemos relevante presentar brevemente el cómo desarrollamos e implementamos este método.

Sólo, y exclusivamente, en el PCA tenemos dos variables porque intentamos encontrar en qué dirección del plano Este/Norte proviene un evento. Estas variables son la dirección Este y Norte de las series temporales registradas. Se espera encontrar una dirección característica cuando un evento proviene del Sangay de cuando no, ya que el movimiento sísmico sobre la corteza terrestre debe venir en esa dirección. Además, esta dirección debe ser diferente en función de la estación de donde tomamos los datos.

En este método se encuentra una base ortogonal donde se tiene la mayor varianza de datos. Los vectores encontrados se los conoce como las componentes principales, y son ordenados de forma descendente tal que la primera componente principal es aquella con mayor varianza de los datos, y así consecutivamente.

Así, si se toma la proyección de los datos (cambio de base) sobre las primeras componentes principales, se puede reducir el número de datos al quedarnos únicamente con aquellos con mayor cambio (varianza) y, por tanto, más significativos.

Con los datos ya preprocesados se puede realizar PCA para encontrar la dirección con mayor varianza, es decir, la dirección en el plano Este-Norte de los sismógrafos de la primera componente principal encontrada.

La hipótesis detrás de la implementación de este método era que la dirección de mayor varianza (primera componente principal) está relacionada con la dirección del volcán Sangay con respecto al sismógrafo.

Esta dirección podría ser caracterizada como el ángulo con respecto al eje Norte de la primera componente principal, por lo que implementamos un programa, Anexo 2, para encontrar ese ángulo.

Primero se elige el nombre de la estación en *est* y el tipo de evento *event* (1 o 0). Después se cargan los ficheros de series temporales preprocesadas en una sola matriz para los cuatro años. Con la función `PCA` de `scikit-learn` (una librería abierta de Python para aplicaciones de inteligencia artificial) se puede aplicar PCA a cada muestra (entre la componente Este y Norte) para encontrar la dirección y varianza de la primera componente principal.

Antes de realizar el ajuste con la función `PCA` se la debe inicializar señalando el número de componentes principales por encontrar (dos componentes). Con el atributo *fit* se hace el ajuste de la función para el conjunto de datos.

Como se desea encontrar la primera componente principal para cada muestra, la función debe ser inicializada y ajustada en un bucle que recorra el número de muestras en la matriz de eventos provenientes del Sangay para cada estación.

Para extraer la dirección de la primera componente y la varianza se usan los atributos *components_[0]* y *explained_variance_ratio_[0]* respectivamente. Finalmente se calcula el ángulo con respecto al eje Norte usando el arcotangente del cociente del componente Este entre la componente Norte de la primera componente principal.

En concreto, si hay una alta varianza en la primera componente se pueden reducir los datos a la mitad al tomar sólo la proyección de las series temporales, Este y Norte, en esa dirección.

2.3. Autoencoders

Como la implementación del PCA fue un método fallido en este trabajo, preprocesamos las series temporales para que sólo quede la componente Este dividida por el máximo absoluto entre las dos componentes horizontales. Esto se realiza para estandarizar los datos entre $[-1, 1]$ y no perder información sobre el máximo entre las dos componentes.

Las redes neuronales de tipo autoencoder [34-36] son un tipo de inteligencia artificial ampliamente usadas para la reducción de datos en series temporales [37-41], por lo que será aplicado como última etapa de procesamiento de los datos antes del detector binario.

2.3.1. Inteligencia Artificial

2.3.1.1. Redes Neuronales

Las neuronas artificiales fueron introducidas como un método computacional para mimetizar las excitaciones recibidas por las neuronas biológicas [49].

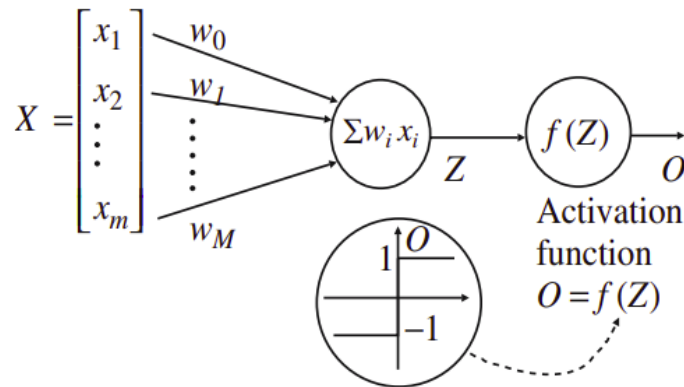


Figura 2.2: Neurona artificial con un vector X como entrada y salida O [50].

En la Figura 2.2 se observa una única neurona, perceptrón, con una salida O de tipo binaria: 1 o 0. Esta neurona tiene de entrada un vector X con m datos, donde cada una de ellas es escalada linealmente por pesos o parámetros (escalares) w .

Después de escalar linealmente la entrada, estas son sumadas y pasadas a una función $f(Z)$ conocida como función de activación, y da el carácter no lineal a las neuronas. Entre las funciones de activación más utilizadas están las sigmoide, relu o tanh [51]. En la Figura 2.3 observamos que la función ReLU es no lineal, a pesar de esto es la función de activación más utilizada en las redes neuronales debido a que para cierto valor la salida de la neurona será nulo y encima de ese valor será evaluado, además de su eficiencia computacional y simplicidad.

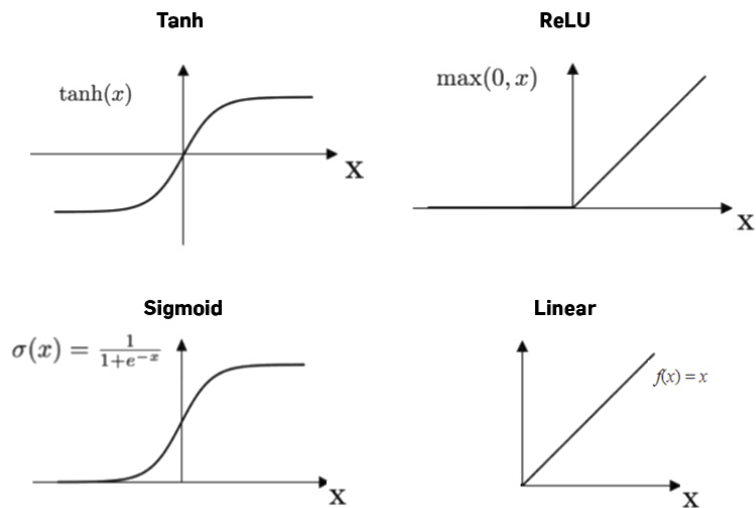


Figura 2.3: Ejemplos de funciones de activación [52].

Otra peculiaridad de las neuronas es que estas deben ser *entrenadas*, ya que deben imitar un comportamiento de salida a partir de una entrada. Este entrenamiento consiste en encontrar los pesos, w , de las neuronas y es similar a ajustar (*fittear*) los parámetros de una función mediante mínimos cuadrados.

De esta forma, las neuronas artificiales son la unidad mínima en la mayoría de modelos de inteligencia artificial. Las redes neuronales artificiales son un conjunto de neuronas interconectadas en diferentes capas [53] por lo que pueden replicar comportamientos no lineales con mayor facilidad.

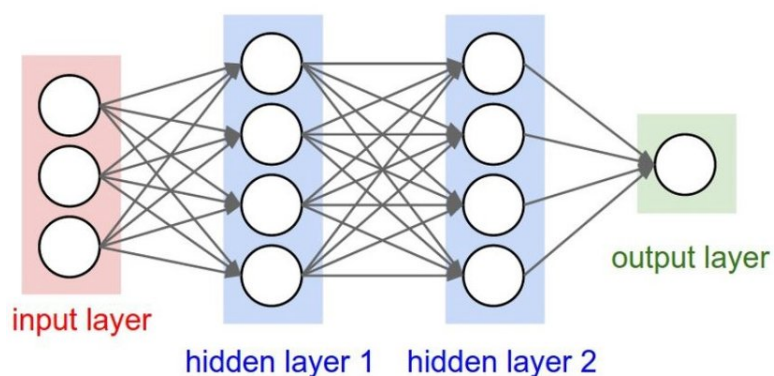


Figura 2.4: Red neuronal densa [54].

En la Figura 2.4 se observa un ejemplo de una red neuronal densa

donde cada círculo representa una neurona. La primera capa (color rojo) consiste en la entrada de la red neuronal, donde cada dato de un vector de entrada es inicializado en cada neurona. Las salidas de la capa de entrada son la entrada de una segunda capa, capa escondida 1 (azul). Esta a su vez tiene salidas que son la entrada de la capa escondida 2 (azul), y finalmente se tiene una capa de salida (verde) que arroja el resultado de la red neuronal densa.

Las redes neuronales densas siguen, generalmente, esta estructura. Hay una capa de entrada, una de salida, y las capas intermedias son conocidas como capas ocultas. La cantidad de capas y neuronas hacen que esta técnica pueda llegar a tener cientos de miles de pesos por encontrar [55].

Las redes neuronales densas tienen una amplia gama de aplicaciones, como la clasificación de eventos [56], predicción [57] o generación de datos [58]. Se dice que un modelo de inteligencia artificial es de tipo supervisado [42] cuando el propósito de la misma es replicar una salida en función de una entrada, para esto se debe tener un conjunto de muestras en la cual se conoce, a priori, la salida esperada.

El algoritmo de backpropagation surge debido al alto coste computacional que demanda encontrar los pesos de una red neuronal en su entrenamiento [59]. Esta es una técnica que actualiza iterativamente los parámetros de una red neuronal al computar el gradiente (cambio) de los mismos cuando se aplica el algoritmo del descenso del gradiente [60, 61].

Al algoritmo del descenso del gradiente se lo puede ver como la minimización de una función (función de pérdida) que caracteriza qué tan bien una red neuronal puede obtener la salida esperada. Por esto, este algoritmo fue crítico en la revolución de los algoritmos supervisados de inteligencia artificial. Un tipo de función de pérdida es el error cuadrático medio (mse) [62] que calcula la diferencia cuadrática promedio entre la salida esperada y estimada por la red neuronal.

Existen varios algoritmos para optimizar la función de pérdida con backpropagation. En particular, el optimizador de Adam [63] es un algoritmo adaptativo de primer orden basado en el descenso del gradiente. Este es bastante utilizado debido a su eficiencia computacional en com-

paración con otros algoritmos como SGD Nesterov [64] o Adagrad [65].

La muestra de datos de este proyecto es el conjunto de todas nuestras series temporales. Este, generalmente, se divide aleatoriamente en un conjunto de entrenamiento y uno de prueba [42, 43]. El conjunto de entrenamiento tiene otra subdivisión conocida como el conjunto de validación que sirve para medir cómo evoluciona el entrenamiento a medida que se aplica el algoritmo de backpropagation iterativamente. La red neuronal es entrenada, únicamente, con el conjunto de entrenamiento.

De manera similar a lo que ocurre con el sobre-ajuste de funciones cuando hay datos de un laboratorio o simulación, existe el sobre-entrenamiento en la inteligencia artificial. Esto se da cuando una red neuronal replica muy bien los resultados del conjunto de entrenamiento pero este no puede generalizarse a un conjunto distinto (conjunto de prueba). Por esta razón, en el aprendizaje supervisado se dividen las muestras en estos conjuntos disconexos y así validar qué tan bueno es el entrenamiento.

La calidad del entrenamiento de una red neuronal es cuantificado mediante indicadores (métricas), como el error cuadrático medio o absoluto en los conjuntos de entrenamiento, validación y de prueba.

Los algoritmos basados en redes neuronales, y en especial las redes densas, tienen cientos de miles de pesos por lo que puede que la red se sobre-entrene. Para entender cómo es el sobre-entrenamiento usaremos un ejemplo: supongamos que tenemos una red neuronal que intenta aprender cómo es un felino a través de un conjunto de imágenes de tigres, esta va a aprender que los felinos tienen rayas negras, son anaranjados o que tienen orejas pequeñas; pero si ingresamos un león o un puma a la red vamos a tener como resultado de que no es un felino. Esto es el sobre-entrenamiento.

Para evitar el sobre-entrenamiento de una red, se ajustan diferentes constantes conocidas como hiperparámetros. Los hiperparámetros son escalares definidos manualmente antes del entrenamiento y no cambian durante el aprendizaje [66]. Estos sirven para optimizar el desempeño de los algoritmos de inteligencia artificial. Existen técnicas como la búsqueda en grilla [67] que permite encontrar el mejor hiperparámetro en un

conjunto discreto de posibles valores.

Dos hiperparámetros usuales en el entrenamiento de redes neuronales son el *batch_size* y el número de *epochs* [68]. El primero se relaciona con el número de muestras que pasa por el optimizador antes de actualizar los parámetros iterativamente. El segundo define el número de veces que el algoritmo de aprendizaje usa todo el conjunto de entrenamiento. Por esto, definir el *epochs* es importante porque podría llevar a un sobre-entrenamiento de la red al ser capaz de replicar bien la salida en el conjunto de entrenamiento pero no en el de prueba.

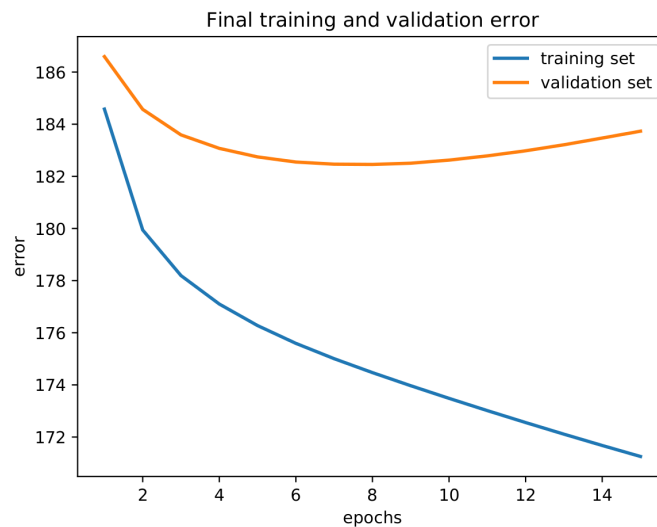


Figura 2.5: Error en el entrenamiento en función del número de *epochs*. Azul: conjunto de entrenamiento, Naranja: conjunto de validación.

En la Figura 2.5 observamos cómo el número de *epochs* disminuye el error del conjunto de entrenamiento porque la red neuronal se vuelve muy buena para predecir con este conjunto, pero vemos que el error en el conjunto de validación disminuye hasta un *epoch* aproximado de 7 y a partir de allí empieza a aumentar. Esto recalca la importancia de definir el valor de este hiperparámetro, porque podría llevar al sobre-entrenamiento de nuestra red neuronal.

2.3.1.2. Autoencoder

Una red neuronal de particular interés son los autoencoders [69-71] debido a su aplicación para reducir el número de datos [38]. Estos tipos

de redes neuronales densas tienen una arquitectura como se observa en la Figura 2.6.

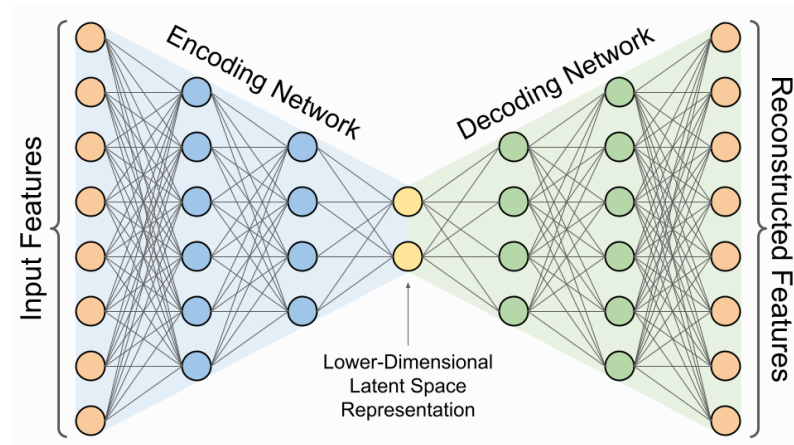


Figura 2.6: Arquitectura de un autoencoder [72].

El autoencoder intenta replicar la entrada de la red como salida (algoritmo auto-supervisado) en una arquitectura de *cuello de botella* [73]. En este tipo de red neuronal densa las capas ocultas después de la capa de entrada van reduciendo, progresivamente, el número de neuronas hasta una capa con un mínimo número de neuronas conocida como capa latente. Luego aumenta el número de neuronas hasta tener el mismo número de datos de entrada. De esta forma, el autoencoder codifica la información de la capa de entrada (neuronas azules de la Figura 2.6) en un mínimo número de datos (neuronas amarillas) y decodifica de vuelta (neuronas verdes).

El valor que las neuronas de la capa latente arrojan para una cierta muestra de entrada es la transformación no lineal de datos en sí. Una fase fundamental en la implementación de autoencoders para la transformación de datos a un menor número de datos es saber cuántas neuronas mínimas, en la capa latente, bastan para representar la información.

2.3.2. Implementación de Autoencoders

Para reducir el número de datos se implementan dos autoencoders por cada estación. Un autoencoder es aplicado en la serie temporal de la componente Este procesada y otro para la densidad de potencia es-

pectral de la serie temporal anterior. La razón por la que aplicamos dos autoencoders es para aumentar la eficiencia de detección de eventos.

Este entrenamiento se da únicamente para las muestras provenientes del Sangay, porque así el autoencoder aprenderá a replicar aproximadamente el comportamiento de eventos del Sangay, en las series temporales y densidades espectrales, y no para eventos externos al volcán.

2.3.3. Autoencoder en el espacio de tiempo

Como cada serie temporal contiene 1201 datos por muestra, esta debe ser replicada como salida en la red neuronal de tipo autoencoder. Es decir, el autoencoder debe tener 1201 neuronas en la capa de entrada y de salida. Para implementar esta red neuronal en Python se llama a *Sequential()*, un módulo de la librería Tensorflow que permite trabajar con redes neuronales.

Como el autoencoder es una red neuronal densa que va desde una entrada a una capa donde se tiene una menor cantidad de datos (capa latente) y luego reconstruye la entrada a partir de estos datos, es necesario determinar la dimensión, número de neuronas, para optimizar el clasificador binario. Para cumplir con este objetivo implementamos un programa (Anexo 3) donde se entrena un primer autoencoder con un número variable de neuronas (tres a veinte) en la capa latente.

En el programa se define el nombre de la estación y se cargan las series temporales preprocesadas para eventos y no eventos del Sangay. En un bucle que varía la dimensión de la capa latente, que determina la precisión de detectar un evento del Sangay, se realizan las siguientes actividades:

- Se inicializa y entrena el autoencoder con el conjunto de muestras provenientes del Sangay con un conjunto de entrenamiento aleatorizado del 50% con respecto al total de muestras. La red neuronal es compilada con una función de pérdida de tipo error cuadrático medio (*mse*) y un optimizador con el algoritmo *Adam*. Como todos los puntos de la serie temporal deben ser reconstruidos correctamente por el autoencoder, se toma una función de pérdida de tipo

mse porque todos los puntos tienen el mismo peso.

- Después del entrenamiento se define una función, mediante el método *function*, que permite extraer los datos de una serie temporal ingresada en el espacio latente del autoencoder. Es decir, los datos a los que se transforma la serie temporal en la capa latente del autoencoder. Esta transformación se realiza al conjunto de eventos y no eventos del Sangay para la estación.
- Finalmente, se entrena un algoritmo de clasificación binaria que en este caso sirve para detectar si se trata de un evento o no evento del Sangay. El algoritmo implementado es una máquina de soporte vectorial (SVM) con kernel de base radial ([rbf](#)) y un hiperparámetro gamma ajustable según el número de muestras. Las máquinas de soporte vectorial son explicadas con más profundidad en la siguiente sección de este capítulo.
- Para la implementación del SVM se toma el valor de 1 para eventos provenientes del Sangay y -1 para los que no, de igual forma se aleatoriza el orden de las muestras y se toma un conjunto de entrenamiento del 30 % con respecto al total. Antes del entrenamiento de la SVM se estandarizan los datos con la función [StandardScaler\(\)](#) que remueve la media y escala los datos con una varianza unitaria. Este procedimiento se realiza para evitar el impacto de datos atípicos en la aplicación del clasificador.
- Se almacena la precisión de predicción en función del número de datos en el espacio latente. Este procedimiento se realiza con los conjuntos de datos de las diferentes estaciones y se toma un valor de la capa latente nn donde la precisión de clasificación se estabiliza.

Se tomará un nn , en la capa latente, cuando la precisión de detectar eventos o no eventos del Sangay sea máxima y nn sea mínima.

2.3.3.1. Entrenamiento del autoencoder

Ya determinado el valor de nn se implementa un programa, Anexo 4, para la aplicación del autoencoder en el espacio de tiempo. Este autoencoder está compuesto por 13 capas ocultas, una capa de entrada y otra

de salida con 1201 neuronas. El número de dimensiones en las capas del autoencoder sigue el orden: 1201, 400, 300, 256, 128, 64, 32, *nn*, 32, 64, 128, 256, 300, 400, 1201.

Todas las neuronas tienen funciones de activación tipo *relu*, excepto las de la capa latente y salida que tienen funciones de activación lineales. Las neuronas en la capa latente y salida tienen funciones de activación lineales ya que los datos pueden tomar valores positivos y negativos.

El entrenamiento se realiza con el 50% de los eventos provenientes del Sangay para cada estación, con una función de pérdida *mse* y optimizador con el algoritmo Adam. Además se toma un *batch_size* de 16, *epochs* de 40 y un conjunto de validación del 30% con respecto al conjunto de entrenamiento.

Se implementa un autoencoder por cada estación dado que las series temporales llegan de distinta forma según la estación. Así, al implementar un autoencoder por cada estación este puede replicar de mejor forma el comportamiento de las señales detectadas por el Sangay.

Finalmente, se inicializan dos funciones *get_latent_layer_output* y *get_output_layer* para extraer los datos de las series temporales en la capa latente (datos a los que se transformó la serie temporal) y capa de salida (serie temporal reconstruida), respectivamente. Con estas funciones se guardan cuatro ficheros, en formato binario, por estación. Estos corresponden a las series temporales transformadas en la capa latente y a las reconstruidas, capa de salida, para eventos y no eventos del Sangay.

2.3.4. Autoencoder en el espacio de frecuencias

Como las series temporales son el registro de una señal en función del tiempo, este tiene frecuencias características que nos puede ayudar a distinguir un evento o no evento del Sangay.

2.3.4.1. Densidad Espectral de Potencia

La densidad de potencia espectral (PSD) es una cantidad que describe el contenido espectral de una señal o serie temporal al cuantificar frecuencias características de un sistema [74]. Existen diferentes métodos

para estimar la PSD, como el método de Barlett [75, 76] o el método de Welch [77]. Este último es popularmente utilizado en la sismología y vulcanología por su bajo coste computacional [78-80].

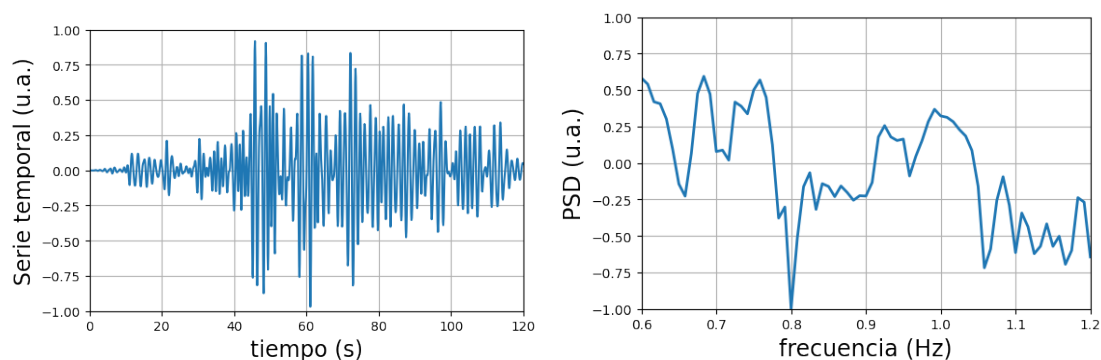


Figura 2.7: Muestra de actividad del Sangay. Izquierda: serie temporal preprocesada (unidades arbitrarias); derecha: densidad de potencia espectral calculada con el método de Welch (unidades arbitrarias).

En la Figura 2.7 observamos cómo la señal de un evento del Sangay tiene frecuencias características, máximos de la PSD, en su rango de filtrado (0.6 a 1.2 Hz). Así, en el espacio de frecuencias podemos caracterizar a la serie temporal en función de otros datos.

2.3.4.2. Extracción de la densidad de potencia espectral

Para el entrenamiento del autoencoder en el espacio de frecuencias se debe extraer la PSD de cada serie temporal preprocesada. Para ello se implementa un programa, Anexo 5, usando el método de Welch.

En este programa se hace un bucle en las señales detectadas y preprocesadas de las siete estaciones, tanto para eventos como para no eventos del Sangay. Primero se realiza una estandarización de las series temporales mediante `StandardScaler()` de `scikit-learn` para que estos tengan varianza unitaria y media cero. Luego se calcula la densidad espectral mediante la función `signal.welch` de `Scipy`. La PSD encontrada se recorta en el rango de interés de 0.6 a 1.2 Hz .

Finalmente, se tienen 73 puntos de datos para cada serie temporal (correspondientes al PSD). Estos son guardados en dos ficheros por cada estación (eventos y no eventos del Sangay) en formato binario.

2.3.4.3. Determinación de la capa latente

De forma análoga al autoencoder en el espacio de tiempo, se determina el número de neuronas (datos) de la capa latente. Este programa, segunda parte del Anexo 3, carga los datos del PSD para eventos del Sangay y entrena un autoencoder con un número variable de neuronas en la capa latente (de tres a veinte).

Como se tienen 73 puntos de datos por cada muestra, el número de dimensiones en las capas de la red neuronal sigue el orden: 73, 64, 32, 25, *nsd*, 25, 32, 64, 73. Donde *nsd* es el número de neuronas en la capa latente. El entrenamiento se realiza con un conjunto de prueba del 30% con respecto al total de muestras de eventos provenientes del Sangay, y es compilado con una función de pérdida *mse*, algoritmo de optimización de Adam, *batch_size* de 16 y un número de *epochs* = 40.

Por último, se entrena un clasificador binario basado en SVM con los mismos parámetros que el implementado para la determinación de dimensiones de la capa latente en el espacio de tiempo. Esto se realiza para determinar la precisión de clasificación binaria en función de la dimensión del espacio latente. Por lo que se tomará un valor de *nsd* donde la precisión se estabilice.

2.3.4.4. Entrenamiento del autoencoder

Con el *nsd* determinado se realiza el entrenamiento del autoencoder en el espacio de frecuencias para cada estación (segunda parte del Anexo 4). El código realiza las siguientes actividades:

- En un bucle para las siete estaciones elegidas de la RENSIG, se cargan los ficheros con las densidades espectrales de cada estación. Para cada muestra se calcula su logaritmo natural y se divide para el máximo absoluto. Esto se realiza para estandarizar los datos entre -1 y 1 . Después, se toma un conjunto de entrenamiento aleatorizado del 50% con respecto al total de muestras provenientes del Sangay. Esto se realiza para que el autoencoder sea capaz de replicar la densidad espectral de los eventos provenientes del Sangay y no de aquellos externos al volcán.

- Se inicializa el autoencoder con los dimensiones de capa latentes: 73, 64, 32, 25, *nsd*, 25, 32, 64, 73. La capa latente y de salida tienen funciones de activación lineales, mientras el resto de capas escondidas tienen funciones tipo *relu*. La red neuronal tiene una función de pérdida de tipo *mse* y optimizador de tipo *Adam*. Además, se toma un *batch_size* de 16, *epochs* de 40 y un conjunto de validación aleatorizado del 30% con respecto al conjunto de entrenamiento.
- Los datos reconstruidos (densidad espectral reconstruida) y de la capa latente (densidad espectral transformada) son guardados en ficheros por cada estación y tipo de evento en formato binario. La extracción se realiza definiendo una función *get_latent_layer_output* para la capa latente y una función *get_capa_final* para la salida reconstruida. De esta forma, se almacena el cómo los autoencoders replican la densidad espectral y el cómo transforma los datos en *nsd* datos para eventos y no eventos del Sangay en cada estación.

2.4. Extracción de estadísticos

Como el entrenamiento de los autoencoders es realizado para los eventos provenientes del Sangay, la reconstrucción de la serie temporal y PSD para no eventos debe diferir de las que tienen los eventos. Sea *est* el nombre de la estación y *ev* 1 o 0, dependiendo de si son eventos o no eventos del Sangay, se calculan diferentes estadísticos (medidas entre la señal original y reconstruida), Anexo 6, de donde se extrae:

- *dme*: valor medio de la diferencia absoluta entre la serie temporal original y reconstruida. Guardado como *est_dmeev.npy*.
- *dst*: desviación estándar de la diferencia de la serie temporal original con la reconstruida. Guardado como *est_dstev.npy*.
- *k*: curtosis de la diferencia de la serie temporal original con la reconstruida. Guardado como *est_kev.npy*.
- *cc*: correlación cruzada entre la serie temporal original con la reconstruida para un desfase 0. Guardado como *est_ccev.npy*.

- `dmpsd`: valor medio de la diferencia absoluta entre la densidad espectral original y reconstruida. Guardado como `est_dmpsddev.npy`.
- `dspsd`: desviación estándar de la diferencia de la densidad espectral original con la reconstruida. Guardado como `est_dspsddev.npy`.

Así, se tienen seis datos adicionales, por muestra, para el entrenamiento del clasificador binario.

Se calcula la correlación cruzada entre la serie temporal original y reconstruida porque se trata de un método estándar para estimar el grado de similitud (semejanza) entre dos series temporales [81, 82].

Sean $\{X_i\}$ y $\{Y_i\}$ (series temporales), donde $i = 1, 2, \dots, N$, se define la correlación cruzada entre las series temporales como:

$$r(d) = \frac{\sum_{i=1}^{N-d} [(X_i - \bar{X})(Y_{i+d} - \bar{Y})]}{\sqrt{\sum_{i=1}^N (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2}} \quad (2.1)$$

donde \bar{X} y \bar{Y} corresponden a las medias de las series temporales y $d = 0, 1, \dots, N - 1$ es el retraso sobre el que se evalúa la correlación cruzada. El valor de $r(d)$ está en el rango de -1 a 1 y es adimensional [82]. Cuando $d = 0$ se trata de la correlación cruzada entre las dos series temporales sin ningún desfase, por lo que el estadístico `cc` cuantifica qué tan similares son las series temporales originales y reconstruidas por el autoencoder.

En el caso que $X = Y$, la correlación cruzada, $r(d)$, es conocida como la función de autocorrelación. Esta técnica la intentamos aplicar en el programa adjunto en el Anexo 2, pero, como se puede observar en el siguiente capítulo de este trabajo, no tuvo buenos resultados.

2.5. Clasificador binario

Por último, es necesario implementar un algoritmo capaz de detectar si se trata de un evento o no del Sangay. En el área de la inteligencia artificial se los conoce como clasificadores binarios, ya que dividen las muestras en dos posibles clases (salidas).

2.5.1. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM) [83, 84] permiten separar un conjunto de muestras en dos clases, eventos o no eventos, al encontrar un hiperplano que maximiza la proyección de los datos de entrenamiento sobre el plano [43].

En el caso de que la separación de los datos no se la pueda realizar por un hiperplano en un espacio euclideo, se cambia la métrica del mismo. Esto se realiza cambiando el producto escalar (kernel) para encontrar una hipersuperficie que sí los separe. El SVM no lineal más utilizado es el que tiene kernel radial [83], donde los datos son transformados de la forma:

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right) \quad (2.2)$$

en un espacio con p datos para las muestras x_i y $x_{i'}$. γ es una constante positiva y un hiperparámetro en el entrenamiento del SVM para la clasificación binaria, por lo que este debe ser definido antes de aplicar el algoritmo. En la Figura 2.8 podemos observar cómo dos tipos de muestras (verdes y rojas) son separados con un kernel radial al pasar del espacio de los datos al espacio con la métrica (kernel) radial (ecuación 2.2). De esta forma, los SVM encuentran una hipersuperficie que separa dos tipos de muestras.

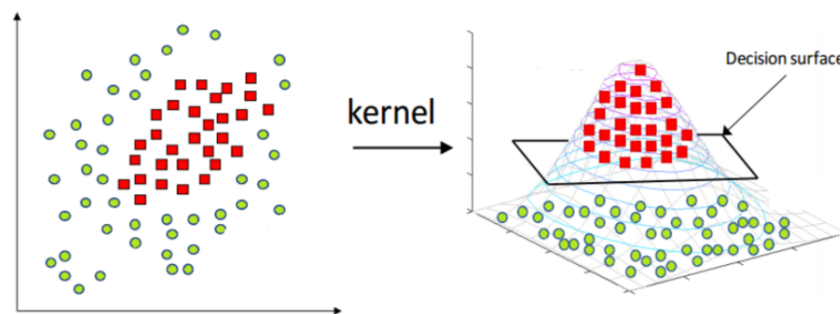


Figura 2.8: Kernel radial aplicado en una Máquina de Soporte Vectorial para clasificación binaria [85].

Si γ es muy grande, se tiene una clasificación susceptible a un sobre-entrenamiento. Por otro lado, un valor pequeño de γ tiene poca varian-

za por lo que dificulta el entrenamiento del clasificador al requerir más muestras para aprender a separar los eventos de los no eventos en una hipersuperficie.

El clasificador binario se implementa con los datos de ambas capas latentes y estadísticos (por muestra) extraídos (Anexo 7). Se tendrá un total de $nn + nsd + 6$ datos por muestra.

El SVM es implementado de la librería scikit-learn, donde el hiperparámetro *gamma* depende del número de muestras: $\gamma = 1/\#muestras$.

Para el entrenamiento del SVM primero se cargan los datos, por muestra, para eventos y no eventos del Sangay en una estación. De igual forma se inicializa un vector de 1 y 0 correspondiendo al objetivo de si es o no un evento. Se toma un conjunto de entrenamiento aleatorizado del 50% con respecto al número total muestras.

Antes del entrenamiento, se estandariza cada dato para tener una media cero y varianza unitaria. Esto se realiza para evitar el efecto de datos atípicos en el clasificador. El clasificador binario será implementado para diferentes combinaciones de los $nn + nsd + 6$ datos para determinar cuál tiene mejor capacidad de detectar un evento en el Sangay.

2.5.2. Validación cruzada

La validación cruzada de k iteraciones (k-fold cross-correlation) es un método estándar para evaluar y comparar algoritmos de inteligencia artificial dividiendo el conjunto de muestras en uno de entrenamiento y otro de validación [86]. En esta técnica se divide el conjunto de entrenamiento en k conjuntos disconexos con el mismo número de muestras. Consecuentemente se realizan k iteraciones de entrenamiento y validación del algoritmo de inteligencia artificial, fijando uno de los conjuntos para la validación y entrenando con el resto de $k - 1$ conjuntos [87].

La validación cruzada permite obtener estimadores del entrenamiento [86]. En particular, en los algoritmos de clasificación binaria se puede estimar la probabilidad de que una muestra se trate de un evento o un no evento.

Por lo tanto, el algoritmo SVM implementado puede obtener la pro-

babilidad en la asignación de clasificación, evento o no evento, mediante validación cruzada de 5 iteraciones. Esto se realiza activando el parámetro *probability=True* en el SVM.

Así, del clasificador binario, además de poder obtener la predicción de si un evento es o no del Sangay, se conoce la probabilidad de que sea o no un evento.

2.5.3. Evaluación del clasificador

Existen diferentes indicadores, métricas, para evaluar la calidad de un modelo de clasificación binaria, como la exactitud, la precisión o la exhaustividad (*recall*) [88]. Todos estos indicadores están relacionados con la matriz de confusión, que indica el número de falsos positivos (FP) y negativos (FN), así como verdaderos positivos (VP) y negativos (VN) al evaluar la predicción del algoritmo de clasificación binaria en el conjunto de prueba.

En la Figura 2.9 observamos una matriz de confusión para la clasificación binaria. Verdadero o Falso significa si una muestra es o no un evento, mientras Positivo o Negativo es cómo el clasificador asigna a la muestra.

	Realmente es positivo	Realmente es negativo
Predicho como positivo	Verdaderos Positivos (VP)	Falsos Positivos (FP)
Predicho como negativo	Falsos Negativos (FN)	Verdaderos Negativos (VN)

Figura 2.9: Matriz de confusión para clasificación binaria [89].

Podemos definir a la exactitud, precisión y exhaustividad como [88]:

$$\text{exactitud} = \frac{VP + VN}{FP + FN + VP + VN} \quad (2.3)$$

$$\text{precisión} = \frac{VP}{VP + FP} \quad (2.4)$$

$$\text{recall} = \frac{VP}{VP + FN} \quad (2.5)$$

La exactitud mide la tasa de eventos clasificados correctamente sobre el total. La precisión mide la capacidad de clasificar eventos positivos en el modelo. Mientras que el recall mide la calidad del modelo para detectar muestras positivas (eventos).

Por esto, como último paso del proyecto se validará la calidad del clasificador binario en cada estación al medir la exactitud, precisión y recall.

Capítulo 3

Resultados, conclusiones y recomendaciones

3.1. Resultados

3.1.1. Métodos fallidos

El análisis de componentes principales no tuvo los resultados esperados. El ángulo correspondiente a la dirección de la primera componente principal con respecto al eje Norte no tiene un cambio significativo de si se trata de eventos o no eventos del Sangay, por lo que no existe una relación entre la dirección del volcán con la estación y este ángulo.

De igual forma, la primera componente principal tiene una baja varianza. Por lo que a pesar de encontrarse una dirección favorable, esta no es significativa a la hora de distinguir de si se trata de un evento o no del volcán. Esto se da para muestras y no muestras del Sangay.

Estos resultados se dan en los sismógrafos de las siete estaciones elegidas de la RENSIG. Por todo lo expuesto anteriormente, no pudimos obtener información entre la dirección de los eventos del sismómetro y la dirección a la que está el Sangay, por lo que se toma únicamente la componente Este de cada muestra dividida por el máximo absoluto entre de las componentes horizontales.

Por otro lado, la autocorrelación de las series temporales también fue un método fallido para intentar caracterizar la diferencia entre un evento

y no evento del Sangay. En la Figura 3.1 observamos la autocorrelación de diferentes eventos y no eventos volcánicos registrados en la estación BULB. No se observan máximos ni mínimos característicos que se repitan en la autocorrelación, por lo que no podemos utilizarla para diferenciar si tenemos o no un evento.

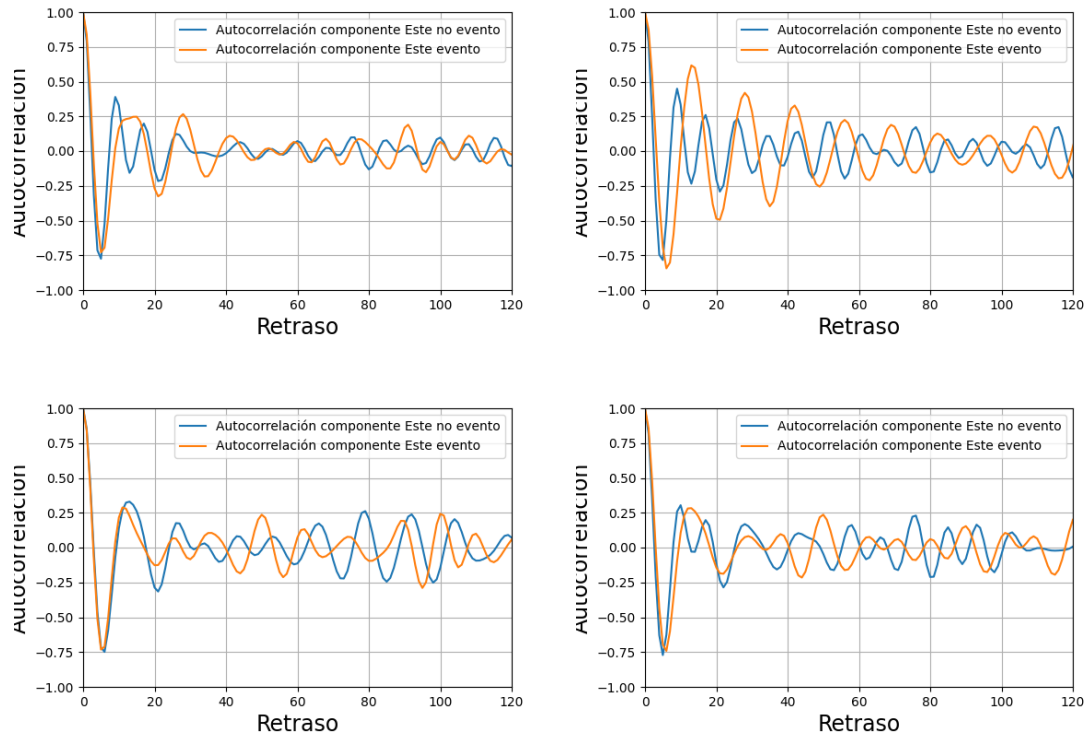


Figura 3.1: Ejemplos de autocorrelación para series temporales en dirección Este para eventos y no eventos del Sangay en la estación BULB.

3.1.2. Métodos exitosos

En la Figura 3.2 se observa cómo el número de neuronas (capa latente) en las redes neuronales de tipo autoencoder afecta a la precisión de clasificación de eventos y no eventos del Sangay.

Tanto para el autoencoder en el espacio de tiempo, línea azul, y en el espacio de frecuencias, línea naranja, se observa que a partir de 10 neuronas la precisión deja de incrementar significativamente y empieza a oscilar. Esto se da por el carácter aleatorio en el entrenamiento de las redes neuronales, dado que cada vez que se entrena el algoritmo se eligen diferentes muestras en el conjunto de prueba y entrenamiento.

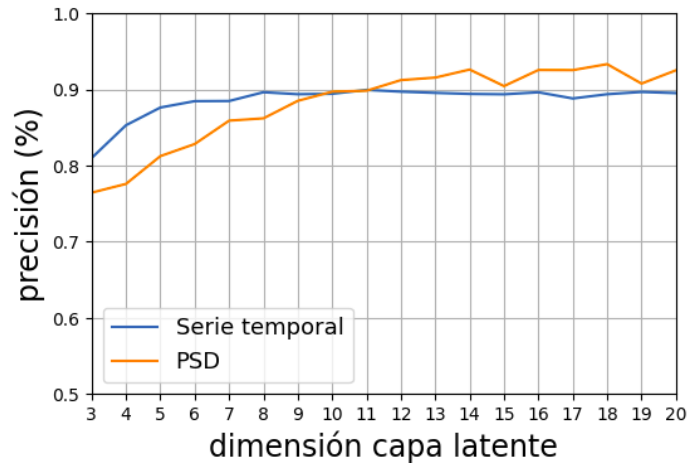


Figura 3.2: Precisión en función de la dimensión de la capa latente para la estación PUYO.

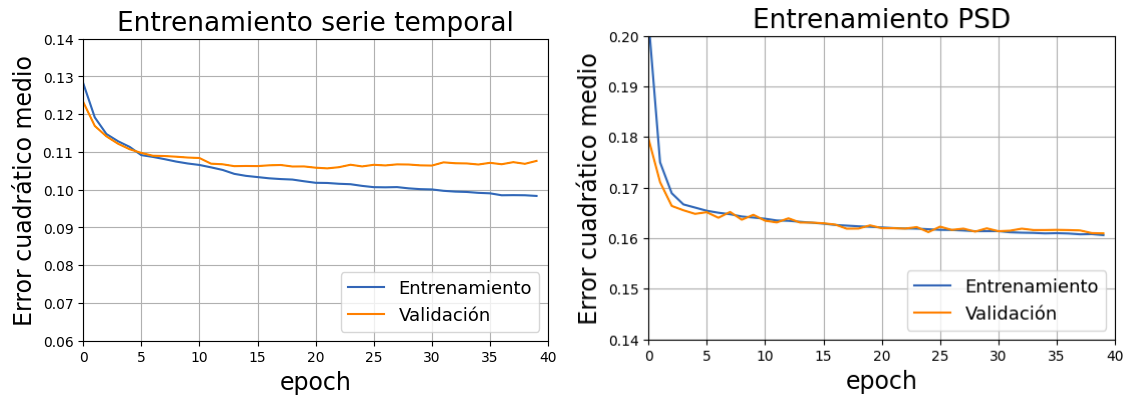


Figura 3.3: Error cuadrático medio para el conjunto de entrenamiento y validación para los autoencoders de la estación PKYU. Izquierda: series temporales; derecha: densidad de potencia espectral.

Como resultado, es suficiente tomar valores de $nn = 10$ y $nsd = 10$ para el entrenamiento de los autoencoders y extracción de estadísticos. Es decir, tanto las series temporales con 1201 datos y las densidades de potencia espectral con 73 datos, son transformadas a 10 datos cada una. El resultado de la Figura 3.2 se da en todas las estaciones sísmicas.

Ya fijada la dimensión del espacio latente para los dos autoencoders, nn y nsd , se realizó el entrenamiento de los autoencoders.

En la Figura 3.3 se observa el error cuadrático medio (mse) en el entrenamiento de los autoencoders con eventos del Sangay para la estación PKYU en función del número de *epochs*. Se observa que los hiperparámetros *batch_size* de 16 y *epochs* de 40 fueron adecuados, ya que las

curvas del error cuadrático medio para el conjunto de validación no aumentan en el intervalo de 0 a 40 epochs. Por lo tanto, no se dio un sobreentrenamiento en las redes neuronales.

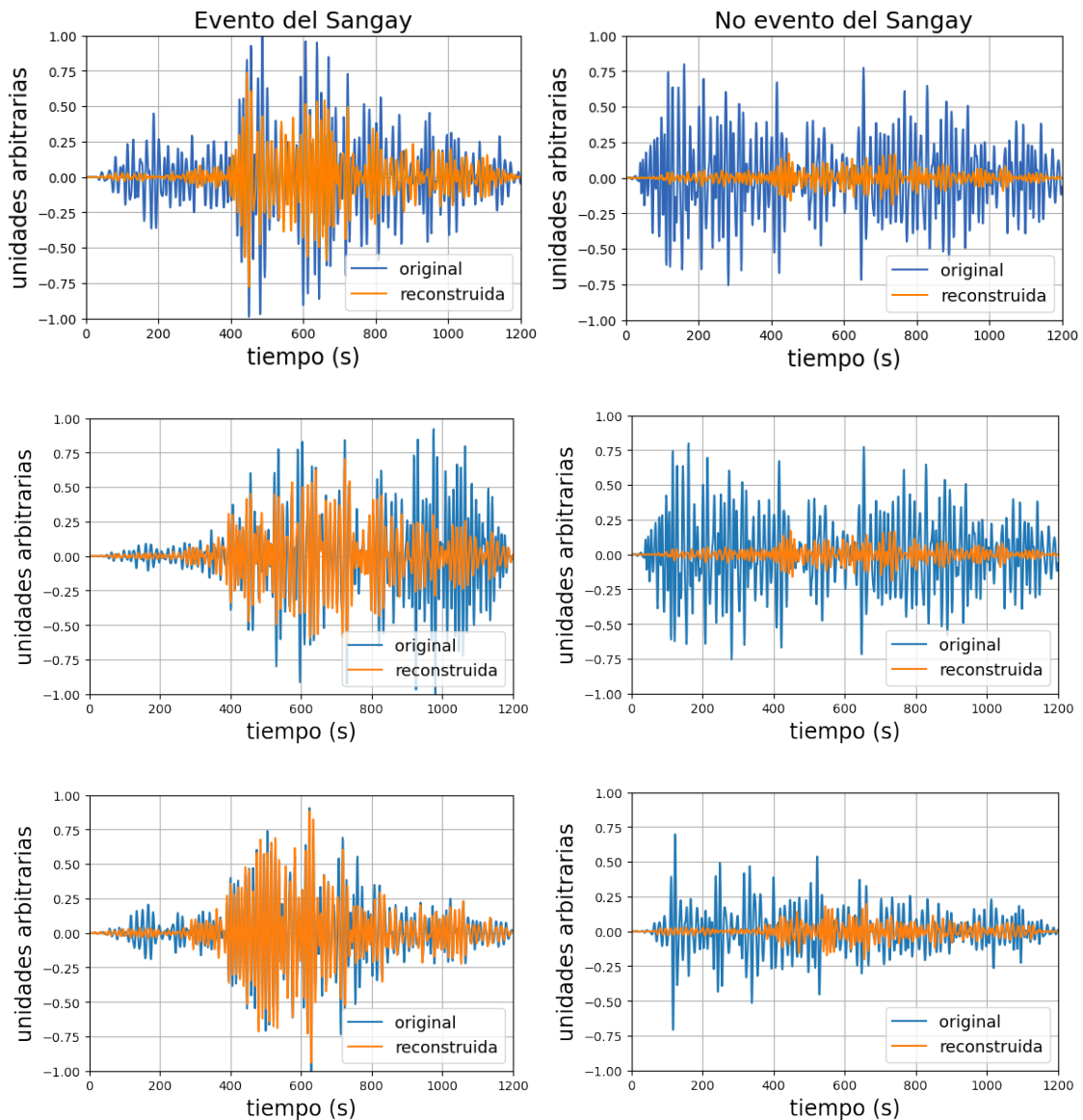


Figura 3.4: Ejemplo de series temporales originales y reconstruidas por el autoencoder para la estación PKYU. Izquierda: evento del volcán Sangay; derecha: no evento del volcán Sangay.

En la Figura 3.4 se observa cómo la serie temporal reconstruida (naranja) puede replicar aproximadamente la serie temporal original (azul) cuando son eventos que provienen del Sangay (Figura 3.4 izquierda), a diferencia de cuando no lo son (Figura 3.4 derecha).

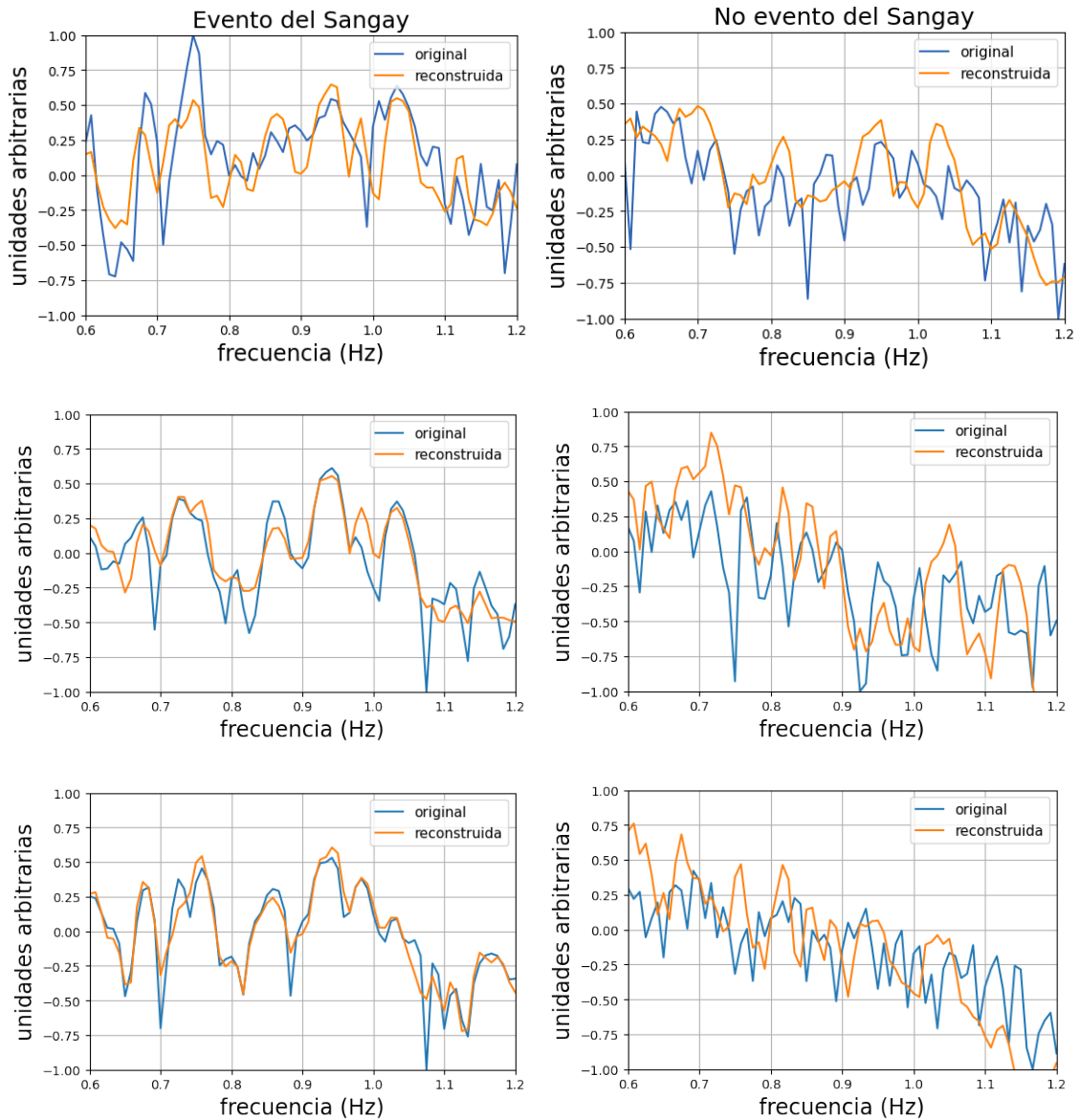


Figura 3.5: Ejemplo de PSDs originales y reconstruidas por el autoencoder de frecuencias en la estación PKYU. Izquierda: evento del volcán Sangay; derecha: no evento del volcán Sangay.

Del mismo modo, el autoencoder en el espacio de frecuencias (Figura 3.5) puede replicar aproximadamente la densidad espectral de eventos provenientes del Sangay. Esto se da porque las curvas reconstruidas de la densidad espectral para eventos (Figura 3.5 izquierda) se acercan más a las PSDs originales que en los no eventos del Sangay (Figura 3.5 derecha).

En resumen, los autoencoders entrenados con eventos del Sangay son capaces de reconstruir aproximadamente las series temporales y PSDs por lo que se observan diferentes distribuciones de los estadísticos extraí-

dos, sección 2.4, entre los eventos y no eventos (Figura 3.6) del volcán.

La correlación cruzada para un desfase 0 (cc) no varía claramente para eventos y no eventos del Sangay (Figura 3.6(a)), a diferencia de otros indicadores como el dme y $dm\psi sd$ (Figura 3.6(b)).

La curtosis (k) es otro indicador cuya distribución difiere entre eventos y no eventos (Figura 3.6(c)). Además, se observa una correlación entre el error medio de la diferencia absoluta (dme y $dm\psi sd$ en la Figura 3.6(b)) con la desviación estándar de ellos (dst y $ds\psi sd$ en la Figura 3.6(d)).

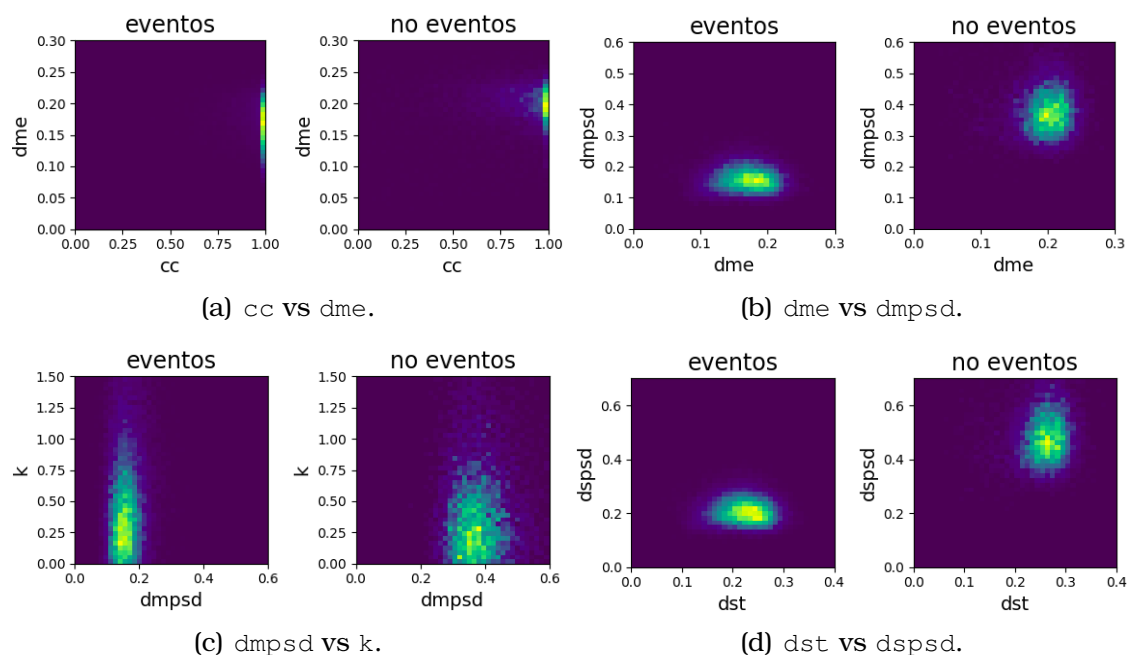


Figura 3.6: Histograma bidimensional de calor para estadísticos entre eventos y no eventos del Sangay en la estación BMAS.

En la Figura 3.7 se observa que usando los 26 datos en total (cuadrados verdes) en el clasificador binario, se tiene la mayor exactitud de predicción con respecto a los otros dos casos. En resumen, se tiene una exactitud de más del 98 % para cada una de las estaciones. Así, entrenar con los datos de las capas latentes y estadísticos es la mejor opción para implementar un detector de actividad volcánica con las señales registradas por el sismómetro de una estación.

Por otra parte, cuando sólo se usan los datos extraídos de la capa latente, se tiene una exactitud aproximada entre el 92 % y 96 % para las señales registradas en las siete estaciones de la red.

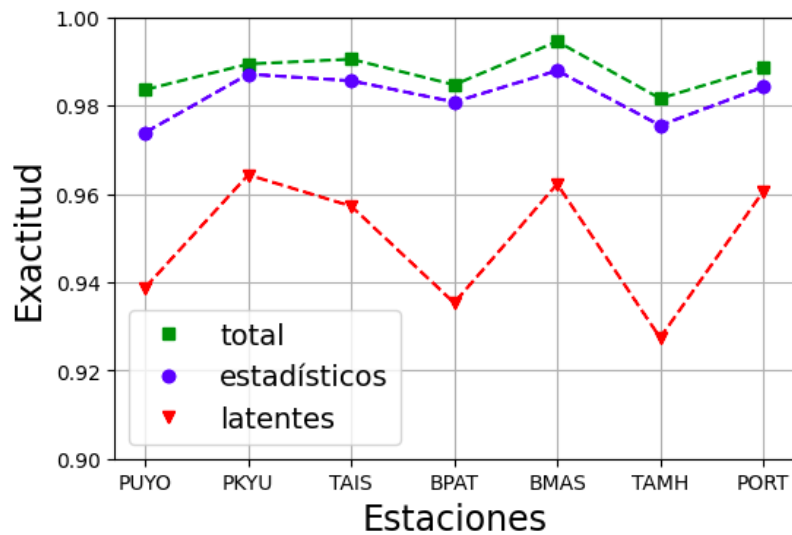


Figura 3.7: Exactitud del detector entrenado con los datos de la capa latente y estadísticos (verde), sólo estadísticos (azul) y sólo de la capa latente (rojo).

Mediante validación cruzada de 5 iteraciones se obtuvo la precisión de clasificación binaria del SVM.

En la Figura 3.8 se observa cómo la probabilidad de clasificación es mayor al 90 % para la mayoría de datos. Esta se obtiene mediante el proceso de validación cruzada de 5 iteraciones señalada en el capítulo anterior.

Las líneas segmentadas son la ubicación del percentil 5 de la distribución en la probabilidad de clasificación para eventos y no eventos del Sangay. Por lo tanto, el 95 % de los no eventos clasificados por el SVM tienen una certeza (probabilidad) del más del 90 %. Mientras que el 95 % de los eventos clasificados por el SVM tienen una probabilidad de clasificación mayor al 95 %. Los resultados presentados en la estación PUYO son similares en las siete estaciones.

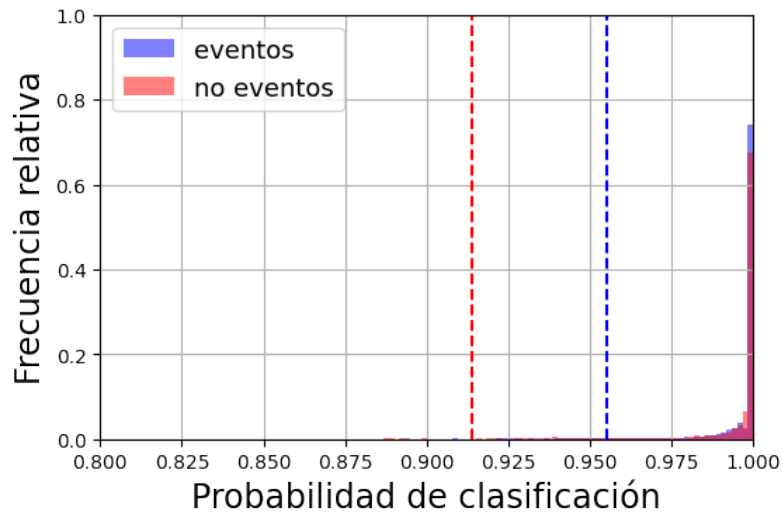


Figura 3.8: Histograma relativo de la probabilidad de clasificación para eventos (azul) y no eventos (rojo) del Sangay en la estación PUYO. Las líneas segmentadas representan el percentil 5 de los datos.

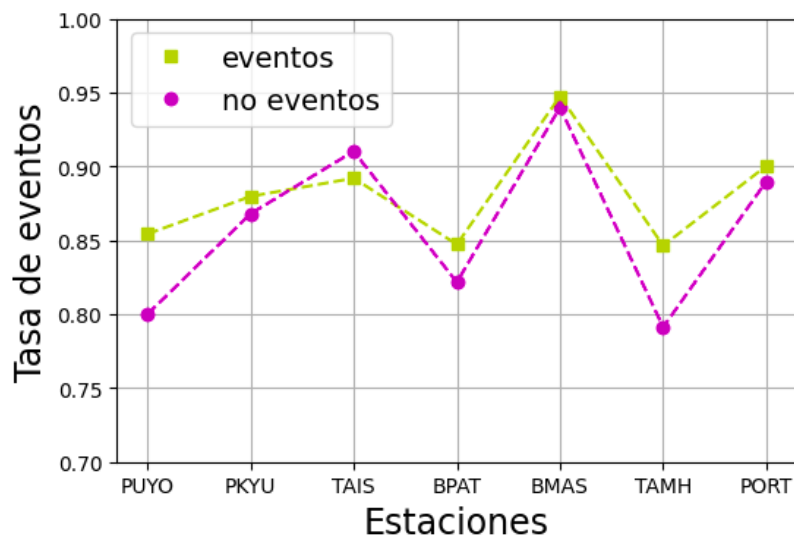


Figura 3.9: Tasa de eventos y no eventos del Sangay clasificados correctamente con una probabilidad mayor al 99% para las estaciones.

Si graficamos la tasa de eventos y no eventos con una probabilidad de clasificación mayor al 99% entre el total de muestras y no muestras del Sangay, respectivamente, observamos que más del 80% de ellos tienen una probabilidad de clasificación mayor al 99%. Esto también se evidencia en la Figura 3.8, donde observamos un pico en la frecuencia relativa cerca de una probabilidad de clasificación de 1.00.

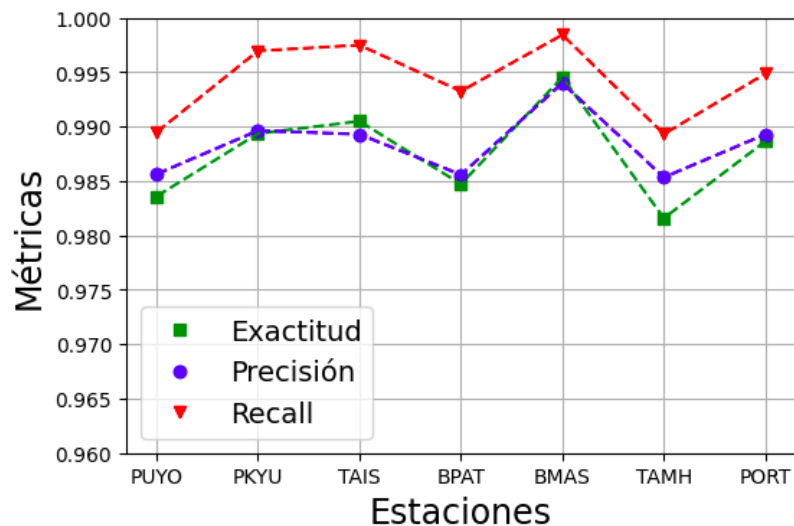


Figura 3.10: Exactitud, precisión y exhaustividad (recall) en el detector entrenado para las estaciones.

Por último, si graficamos la exactitud, precisión y recall en la detección de eventos del Sangay cuando usamos los datos de la capa latente y estadísticos (26 datos por muestra), observamos que: el recall es mayor al 99% en las siete estaciones de la red, datos rojos de la Figura 3.10, mientras la precisión y exactitud es mayor al 98% (datos azules y verdes de la Figura 3.10). En concreto, el detector puede clasificar correctamente un evento o no evento del volcán Sangay.

3.2. Conclusiones y recomendaciones

Fue posible construir un buen clasificador binario de actividad volcánica del Sangay utilizando las componentes horizontales de una única estación mediante redes neuronales tipo autoencoder y SVM. Esto se verifica por cómo la exactitud, precisión y recall de detección es superior al 98% en las siete estaciones elegidas de la RENSIG.

Cuando utilizamos la red neuronal de tipo autoencoder, a partir de 10 datos en la capa latente del espacio de frecuencia y espacio de tiempo se tiene una máxima precisión. Así, la red neuronal es capaz de reconstruir de manera bastante aproximada la serie temporal y su densidad espectral desde la capa latente al transformar no linealmente los datos.

De igual forma, se tiene que más del 80% de eventos clasificados tienen

una probabilidad de ser asignados correctamente mayor al 99% en todas las estaciones.

Este detector tiene una alta capacidad de reconocer eventos del Sangay, recall, en cada estación lejana a pesar que se implementó con los datos de cada estación individualmente.

El desempeño del detector puede mejorar si se consideran datos simultáneos de todas las estaciones, por lo que hay características, como el tiempo de arribo de los eventos y la intensidad de las señales, que están relacionadas a la distancia del volcán a las estaciones. Esto permitiría implementar una red de detección de actividad volcánica en el Sangay utilizando un enjambre de estaciones regionales lejanas a la cumbre del volcán.

A pesar de la utilidad de las redes neuronales de tipo autoencoder en transformar no linealmente los datos a un menor número de ellos, es un algoritmo que funciona como una caja negra debido a la dificultad de interpretar y relacionar los datos del espacio latente con magnitudes físicas. Por lo tanto, es importante entender cómo se relacionan los datos no lineales encontrados con los autoencoders, espacio latente, con las muestras provenientes del Sangay para generar un futuro modelo de detección de eventos y entender la fenomenología de los volcanes.

Vásconez et al. [2] tomaron un filtrado entre 0.6 y 1.2 Hz para la componente vertical del sismógrafo. Se podría estudiar cómo cambiar el poder predictivo de nuestro clasificador binario cuando cambiamos el intervalo de frecuencias del filtro, estudio en espectrogramas de muestras provenientes del Sangay, y en el intervalo de tiempos (ventana de actividad) fijada en el preprocesamiento de las series temporales. Además, cambiar la ventana de actividad a un tiempo menor y mejorar el filtrado de las series temporales podría mejorar la reducción lineal de datos por PCA.

Este detector también puede ser implementado para detectar actividad en otros volcanes como el Reventador, usando datos de otras estaciones lejanas. De igual forma, este clasificador puede ser replicado para ampliar el catálogo sísmico de eventos que no han sido detectados por el Sangay, u otros volcanes como el Tungurahua (en las anteriores décadas).

Otra investigación que se puede realizar a futuro es la implementa-

ción de un clasificador multiclase que determine si hay actividad o no en el volcán, y de qué tipo de evento volcánico puede tratarse usando los datos de las capas latentes y los estimadores estadísticos de las series temporales y densidades espectrales.

Capítulo A

Anexos

A.1. Preprocesamiento de series temporales

shorturl.at/fjoV1

A.2. Aplicación de PCA y Autocorrelación

shorturl.at/abqrt

A.3. Dimensión espacio latente de los autoencoders

shorturl.at/gmWZ0

A.4. Entrenamiento de autoencoders

shorturl.at/awxz1

A.5. Extracción de PSD (Método de Welch)

shorturl.at/bhlAX

A.6. Extracción de estadísticos con los datos originales y reconstruidos

shorturl.at/grV68

A.7. Entrenamiento de clasificador binario

shorturl.at/bCHNO

Referencias bibliográficas

- ¹J. Ordóñez, S. Vallejo, J. Bustillos, M. Hall, D. Andrade, S. Hidalgo y P. Samaniego, *Volcán Sangay. Peligros Volcánicos Potenciales*. Inf. téc. (Instituto Geofísico - Escuela Politécnica Nacional, Quito - Ecuador, 2022), pág. 1.
- ²F. J. Vasconez, S. Hidalgo, J. Battaglia, S. Hernandez, B. Bernard, D. Coppola, S. Valade, P. Ramón, S. Arellano, C. Liorzou, M. Almeida, M. Ortiz, J. Córdova y A. V. Müller, «Linking ground-based data and satellite monitoring to understand the last two decades of eruptive activity at Sangay volcano, Ecuador», *Bulletin of Volcanology* **84**, 49 (2022).
- ³S. Kinoshita, «Low-frequency and trend compensation of broadband seismograms», *Earth, Planets and Space* **64**, e5-e8 (2012).
- ⁴J. D. Hamilton, *Time Series Analysis*, 1st (Princeton University Press, 1994), pág. 25.
- ⁵U.S. Geological Survey, *Seismographs - Keeping Track of Earthquakes* | U.S. Geological Survey, <https://www.usgs.gov/programs/earthquake-hazards/seismographs-keeping-track-earthquakes>, [Online; accedido el 3-Julio-2022], 2022.
- ⁶P. Tosi, P. Sbarra y V. D. Rubeis, «Earthquake sound perception», *Geophysical Research Letters* **39**, 10.1029/2012GL054382 (2012).
- ⁷J. B. Johnson y J. M. Lees, «Plugs and chugs—seismic and acoustic observations of degassing explosions at Karymsky, Russia and Sangay, Ecuador», *Journal of Volcanology and Geothermal Research* **101**, 67-82 (2000).

- ⁸K. I. Konstantinou y C. H. Lin, «Nonlinear Time Series Analysis of Volcanic Tremor Events Recorded at Sangay Volcano, Ecuador», [Pure and Applied Geophysics](#) **161**, 145-163 (2004).
- ⁹J. M. Lees y M. Ruiz, «Non-linear explosion tremor at Sangay, Volcano, Ecuador», [Journal of Volcanology and Geothermal Research](#) **176**, 170-178 (2008).
- ¹⁰H. D. Ortiz, R. S. Matoza, C. Garapaty, K. Rose, P. Ramón y M. C. Ruiz, «Multi-year regional infrasound detection of Tungurahua, El Reventador, and Sangay volcanoes in Ecuador from 2006 to 2013», en (2020), pág. 022003.
- ¹¹S. A. Carn, A. J. Krueger, S. Arellano, N. A. Krotkov y K. Yang, «Daily monitoring of Ecuadorian volcanic degassing from space», [Journal of Volcanology and Geothermal Research](#) **176**, 141-150 (2008).
- ¹²A. M. M. Rivera, F. Amelung y P. Mothes, «Volcano deformation survey over the Northern and Central Andes with ALOS InSAR time series», [Geochemistry, Geophysics, Geosystems](#) **17**, 2869-2883 (2016).
- ¹³X. Liang y C. Wang, «Separating hypersurfaces of SVMs in input spaces», [Pattern Recognition Letters](#) **30**, 469-476 (2009).
- ¹⁴Q. He, Z.-Z. Shi, L.-A. Ren y E. Lee, «A novel classification method based on hypersurface», [Mathematical and Computer Modelling](#) **38**, 395-407 (2003).
- ¹⁵D. Ienco y R. Interdonato, *Deep Multivariate Time Series Embedding Clustering via Attentive-Gated Autoencoder* (2020), págs. 318-329.
- ¹⁶C.-Y. Liou, W.-C. Cheng, J.-W. Liou y D.-R. Liou, «Autoencoder for words», [Neurocomputing](#) **139**, 84-96 (2014).
- ¹⁷Q.-s. Zhang y S.-c. Zhu, «Visual interpretability for deep learning: a survey», [Frontiers of Information Technology & Electronic Engineering](#) **19**, 27-39 (2018).
- ¹⁸S. M. Shankaranarayana y D. Runje, *ALIME: Autoencoder Based Approach for Local Interpretability* (2019), págs. 454-463.

- ¹⁹B. Adhikari, S. Dahal, M. Karki, R. K. Mishra, R. K. Dahal, S. Sasmal y V. Klausner, «Application of wavelet for seismic wave analysis in Kathmandu Valley after the 2015 Gorkha earthquake, Nepal», [Geoenvironmental Disasters](#) **7**, 2 (2020).
- ²⁰B. Jun-Wei, W. Zuo-Ju, W. Zhi-Jia, O. Fang y C. Yuan, «Wavelet Transform and Its Application in Earthquake Engineering», [2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications](#), 1126-1128 (2014).
- ²¹S. Lapins, D. C. Roman, J. Rougier, S. D. Angelis, K. V. Cashman y J. M. Kendall, «An examination of the continuous wavelet transform for volcano-seismic spectral analysis», [Journal of Volcanology and Geothermal Research](#) **389**, 106728 (2020).
- ²²H. Kumagai, M. Nakano, T. Maeda, H. Yepes, P. Palacios, M. Ruiz, S. Arrais, M. Vaca, I. Molina y T. Yamashima, «Broadband seismic monitoring of active volcanoes using deterministic and stochastic approaches», [Journal of Geophysical Research](#) **115**, B08303 (2010).
- ²³P. Segall, «Volcano deformation and eruption forecasting», [Geological Society, London, Special Publications](#) **380**, 85-106 (2013).
- ²⁴R. Lara-Cueva, J. C. Larco, D. S. Benítez, N. Pérez, F. Grijalva y M. Ruiz, «On finding possible frequencies for recognizing microearthquakes at Cotopaxi volcano: A machine learning based approach», [Journal of Volcanology and Geothermal Research](#) **407**, 107092 (2020).
- ²⁵M. Curilem, R. Soto, F. Huenupan, C. S. Martin, L. Franco y C. Cardona, «Hierarchical classification structure based on SVM for volcano seismic events», [2019 IEEE Latin American Conference on Computational Intelligence \(LA-CCI\)](#), 1-6 (2019).
- ²⁶B. Apolloni et al., «Support vector machines and MLP for automatic classification of seismic signals at Stromboli volcano», en Proc. 19th Italian Workshop Neural Nets, vol. 204 (2009), pág. 116.
- ²⁷K. Pearson, «LIII. On lines and planes of closest fit to systems of points in space», [The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science](#) **2**, 559-572 (1901).
- ²⁸H. Hotelling, «Analysis of a complex of statistical variables into principal components.», [Journal of Educational Psychology](#) **24**, 417-441 (1933).

- ²⁹C. Spearman, «"General Intelligence,. objectively Determined and Measured», [The American Journal of Psychology](#) **15**, 201 (1904).
- ³⁰V. Klema y A. Laub, «The singular value decomposition: Its computation and some applications», [IEEE Transactions on Automatic Control](#) **25**, 164-176 (1980).
- ³¹E. Keogh, K. Chakrabarti, M. Pazzani y S. Mehrotra, «Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases», [Knowledge and Information Systems](#) **3**, 263-286 (2001).
- ³²K. F. Tiampo, P. J. González, S. Samsonov, J. Fernández y A. Camacho, «Principal component analysis of MSBAS DInSAR time series from Campi Flegrei, Italy», [Journal of Volcanology and Geothermal Research](#) **344**, 139-153 (2017).
- ³³Z. Lomnicki y S. Zaremba, «On the estimation of autocorrelation in time series», [The Annals of Mathematical Statistics](#) **28**, 140-158 (1957).
- ³⁴P. Baldi, «Autoencoders, unsupervised learning, and deep architectures», en [Proceedings of ICML workshop on unsupervised and transfer learning \(JMLR Workshop y Conference Proceedings, 2012\)](#), págs. 37-49.
- ³⁵D. Bank, N. Koenigstein y R. Giryes, «Autoencoders», [arXiv preprint arXiv:2003.05991](#) (2020).
- ³⁶U. Michelucci, «An Introduction to Autoencoders», [arXiv preprint arXiv:2201.03898](#) (2022).
- ³⁷G. E. Hinton y R. R. Salakhutdinov, «Reducing the Dimensionality of Data with Neural Networks», [Science](#) **313**, 504-507 (2006).
- ³⁸Y. Wang, H. Yao y S. Zhao, «Auto-encoder based dimensionality reduction», [Neurocomputing](#) **184**, 232-242 (2016).
- ³⁹W. Wang, Y. Huang, Y. Wang y L. Wang, «Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction», [2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops](#), 496-503 (2014).
- ⁴⁰N. Tavakoli, S. Siami-Namini, M. A. Khanghah, F. M. Soltani y A. S. Namin, «An autoencoder-based deep learning approach for clustering time series data», [SN Applied Sciences](#) **2**, 937 (2020).

- ⁴¹T. Kieu, B. Yang, C. Guo y C. S. Jensen, «Outlier Detection for Time Series with Recurrent Autoencoder Ensembles.», en IJCAI (2019), págs. 2725-2732.
- ⁴²C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st (Springer, 2006).
- ⁴³G. James, D. Witten, T. Hastie y R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103 (Springer New York, 2013).
- ⁴⁴M. Grandini, E. Bagli y G. Visani, *Metrics for Multi-Class Classification: an Overview*, 2020.
- ⁴⁵L. Krischer, T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron y J. Wassermann, «ObsPy: a bridge for seismology into the scientific Python ecosystem», [Computational Science & Discovery](#) **8**, 014003 (2015).
- ⁴⁶M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr y J. Wassermann, «ObsPy: A Python Toolbox for Seismology», [Seismological Research Letters](#) **81**, 530-533 (2010).
- ⁴⁷Python documentation, 5. *Data Structures*. *Python 3.10.6 documentation*, <https://docs.python.org/3/tutorial/datastructures.html>, [Online; accedido el 5-Agosto-2022], 2022.
- ⁴⁸H. Nyquist, «Certain topics in telegraph transmission theory», *Transactions of the American Institute of Electrical Engineers* **47**, 617-644 (1928).
- ⁴⁹F. Rosenblatt, «The perceptron: a probabilistic model for information storage and organization in the brain.», *Psychological review* **65**, 386 (1958).
- ⁵⁰D. Andina, A. Vega-Corona, J. I. Seijas y J. Torres-García, *Neural Networks Historical Review* (Springer US, 2007), págs. 39-65.
- ⁵¹S. Sharma, S. Sharma y A. Athaiya, «Activation functions in neural networks», *towards data science* **6**, 310-316 (2017).
- ⁵²*Activation Function - AI Wiki*, <https://machine-learning.paperspace.com/wiki/activation-function>, Accessed: 2022-08-28.
- ⁵³S.-C. Wang, *Artificial Neural Network* (Springer US, 2003), págs. 81-100.

- ⁵⁴F. Amherd y E. Rodriguez, «Heatmap-based Object Detection and Tracking with a Fully Convolutional Neural Network», (2021).
- ⁵⁵Y. LeCun, Y. Bengio y G. Hinton, «Deep learning», [Nature](#) **521**, 436-444 (2015).
- ⁵⁶A. Krizhevsky, I. Sutskever y G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks», en [Advances in Neural Information Processing Systems](#), vol. 25, ed. por F. Pereira, C. Burges, L. Bottou y K. Weinberger (2012).
- ⁵⁷K. Y. Tam y M. Y. Kiang, «Managerial applications of neural networks: the case of bank failure predictions», *Management science* **38**, 926-947 (1992).
- ⁵⁸I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville e Y. Bengio, «Generative adversarial nets», *Advances in neural information processing systems* **27** (2014).
- ⁵⁹P. Werbos, «Beyond regression:"new tools for prediction and analysis in the behavioral sciences», Ph. D. dissertation, Harvard University (1974).
- ⁶⁰P. J. Werbos, «Backpropagation through time: what it does and how to do it», *Proceedings of the IEEE* **78**, 1550-1560 (1990).
- ⁶¹R. Hecht-Nielsen, «Theory of the backpropagation neural network», en *Neural networks for perception* (Elsevier, 1992), págs. 65-93.
- ⁶²P. Christoffersen y K. Jacobs, «The importance of the loss function in option valuation», *Journal of Financial Economics* **72**, 291-318 (2004).
- ⁶³D. P. Kingma y J. Ba, «Adam: A method for stochastic optimization», arXiv preprint arXiv:1412.6980 (2014).
- ⁶⁴Y. Nesterov, «A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$ », en *Doklady an ussr*, vol. 269 (1983), págs. 543-547.
- ⁶⁵M. D. Zeiler, «Adadelta: an adaptive learning rate method», arXiv preprint arXiv:1212.5701 (2012).
- ⁶⁶M. Claesen y B. De Moor, «Hyperparameter search in machine learning», arXiv preprint arXiv:1502.02127 (2015).

- ⁶⁷X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao y F. Porikli, «Hyperparameter optimization for tracking with continuous deep q-learning», en Proceedings of the IEEE conference on computer vision and pattern recognition (2018), págs. 518-527.
- ⁶⁸J. Brownlee, *Difference Between a Batch and an Epoch in a Neural Network*, <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>, [Online; accedido el 14-Julio-2022], 2018.
- ⁶⁹Y. LeCun, «PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)», (1987).
- ⁷⁰H. Bourlard e Y. Kamp, «Auto-association by multilayer perceptrons and singular value decomposition», *Biological cybernetics* **59**, 291-294 (1988).
- ⁷¹G. E. Hinton y R. Zemel, «Autoencoders, minimum description length and Helmholtz free energy», *Advances in neural information processing systems* **6** (1993).
- ⁷²R. O'Connor, *Introduction to Variational Autoencoders Using Keras*, <https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/>, [Online; accedido el 11-Julio-2022], 2022.
- ⁷³L. Hollenstein, L. Lichtensteiger, T. Stadelmann, M. Amirian, L. Budde, J. Meierhofer, R. Füchslin y T. Friedli, «Unsupervised Learning and Simulation for Complexity Management in Business Operations», **313-331 (2019)**.
- ⁷⁴R. N. Youngworth, B. B. Gallagher y B. L. Stamper, «An overview of power spectral density (PSD) calculations», en , ed. por H. P. Stahl (ago. de 2005), 58690U.
- ⁷⁵M. S. Bartlett, «Smoothing Periodograms from Time-Series with Continuous Spectra», *Nature* **161**, 686-687 (1948).
- ⁷⁶M. S. Bartlett, «PERIODOGRAM ANALYSIS AND CONTINUOUS SPECTRA», *Biometrika* **37**, 1-16 (1950).
- ⁷⁷P. Welch, «The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms», *IEEE Transactions on Audio and Electroacoustics* **15**, 70-73 (1967).

- ⁷⁸R. A. Lara-Cueva, D. S. Benitez, E. V. Carrera, M. Ruiz y J. L. Rojo-Alvarez, «Automatic Recognition of Long Period Events From Volcano Tectonic Earthquakes at Cotopaxi Volcano», *IEEE Transactions on Geoscience and Remote Sensing* **54**, 5247-5257 (2016).
- ⁷⁹J. R. Evans, F. Followill, C. R. Hutt, R. P. Kromer, R. L. Nigbor, A. T. Ringler, J. M. Steim y E. Wielandt, «Method for Calculating Self-Noise Spectra and Operating Ranges for Seismographic Inertial Sensors and Recorders», *Seismological Research Letters* **81**, 640-646 (2010).
- ⁸⁰A. Avallone, M. Marzario, A. Cirella, A. Piatanesi, A. Rovelli, C. D. Alessandro, E. D'Anastasio, N. D'Agostino, R. Giuliani y M. Mattone, «Very high rate (10 Hz) GPS seismology for moderate-magnitude earthquakes: The case of the Mw 6.3 L'Aquila (central Italy) event», *Journal of Geophysical Research* **116**, B02305 (2011).
- ⁸¹D. W. Boyd, *Stochastic Analysis* (Elsevier, 2001), págs. 211-227.
- ⁸²P. Bourke, «Cross correlation, autocorrelation, 2d pattern identification», (1996).
- ⁸³C. Cortes y V. Vapnik, «Support-vector networks», *Machine Learning* **20**, 273-297 (1995).
- ⁸⁴B. Schölkopf, A. J. Smola, R. C. Williamson y P. L. Bartlett, «New Support Vector Algorithms», *Neural Computation* **12**, 1207-1245 (2000).
- ⁸⁵*Support Vector Machines its not so complicated*, <https://kdagiit.medium.com/support-vector-machines-its-not-so-complicated-5ccf66278159>, Accessed: 2022-08-28.
- ⁸⁶P. Refaeilzadeh, L. Tang y H. Liu, *Cross-Validation* (Springer New York, 2016), págs. 1-7.
- ⁸⁷D. Berrar, *Cross-Validation* (Elsevier, 2019), págs. 542-545.
- ⁸⁸scikit-learn, 3.3. *Metrics and scoring: quantifying the quality of predictions*, https://scikit-learn.org/stable/modules/model_evaluation.html#, [Online; accedido el 18-Julio-2022], 2022.
- ⁸⁹*Capítulo 10 Aprendizaje Supervisado | Data Science con R*, <https://bookdown.org/dparedesi/data-science-con-r/aprendizaje-supervisado.html>, Accessed: 2022-08-28.