



**Creación de algoritmos inteligentes basados en la teoría de Machine Learning tradicional  
para detección de eventos sísmicos en el volcán Llaima**

Castillo Tipantuña, Edwin Alexander

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Telecomunicaciones

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en

Telecomunicaciones

Ing. Lara Cueva, Román Alcides PhD.

25 de agosto del 2023



Plagiarism report

O2\_CastilloEdwin.pdf

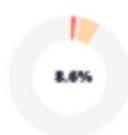
## Scan details

Scan time:  
August 25th, 2023 at 18:56 UTC

Total Pages:  
46

Total Words:  
11374

## Plagiarism Detection



Types of plagiarism		Words
Identical	2%	227
Minor Changes	0.6%	68
Paraphrased	6%	680
Omitted Words	0%	0

## AI Content Detection



Text coverage  
 AI text  
 Human text

## Plagiarism Results: (21)

### Your File

4.1%

javie

No introduction available.

### Your File

2%

Usuario de Windows

No introduction available.

Firma:



ROMAN ALCIDES LARA  
CUEVA

Dr. Lara Cueva, Román Alcides

Director



**Departamento de Eléctrica, Electrónica y Telecomunicaciones**

**Carrera de Telecomunicaciones**

**Certificación**

Certifico que el trabajo de integración curricular: **“Creación de algoritmos inteligentes basados en la teoría de Machine Learning tradicional para detección de eventos sísmicos en el volcán Llaima”** fue realizado por el señor **Castillo Tipantuña, Edwin Alexander**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 25 de agosto de 2023

Firma:



**Dr. Lara Cueva, Román Alcides**

C. C 1713988218



Departamento de Eléctrica, Electrónica y Telecomunicaciones  
Carrera de Telecomunicaciones

**Responsabilidad de Autoría**

Yo, **Castillo Tipantuña, Edwin Alexander**, con cédula de ciudadanía n°1726088337, declaro que el contenido, ideas y criterios del trabajo de integración curricular: **Creación de algoritmos inteligentes basados en la teoría de Machine Learning tradicional para detección de eventos sísmicos en el volcán Llaima** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 25 de agosto de 2023

Firma

**Castillo Tipantuña, Edwin Alexander**

C.C.: 1726088337



**Departamento de Eléctrica, Electrónica y Telecomunicaciones**

**Carrera de Telecomunicaciones**

**Autorización de Publicación**

Yo **Castillo Tipantuña, Edwin Alexander**, con cédula de ciudadanía n°1726088337, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Creación de algoritmos inteligentes basados en la teoría de Machine Learning tradicional para detección de eventos sísmicos en el volcán Llaima** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

**Sangolquí, 25 de agosto de 2023**

Firma

**Castillo Tipantuña, Edwin Alexander**

C.C: 1726088337

## **Dedicatoria**

Quiero dedicar este proyecto a Dios, que durante todo mi camino universitario me dio las fuerzas, sabiduría y persistencia necesaria para seguir adelante y alcanzar mis metas.

Dedico a mis padres Edwin y Alicia por ser mi apoyo inquebrantable durante este largo camino en el que me dieron la motivación e inspiración para seguir adelante, por sus palabras de aliento a nunca rendirme y la confianza depositada en mí.

A mi hermano Ismael, quien ha estado a mi lado dándome su apoyo y respaldo incondicional en todo momento.

A mis abuelitos, tíos, tías, y primos cuya presencia y apoyo constante ha sido un pilar importante en mi vida.

A mis amigos por el apoyo en los buenos y malos momentos atravesados, brindándome su solidaridad y aliento cuando más lo necesitaba.

Edwin Alexander Castillo Tipantuña

## Agradecimiento

Expreso mi sincero agradecimiento a la Universidad de las Fuerzas Armadas ESPE y a la carrera de Telecomunicaciones por brindarme el conocimiento y la oportunidad de formarme como profesional.

Un agradecimiento especial al Dr. Román Lara por toda la paciencia, dedicación y conocimiento aportado que fue indispensable para poder culminar con mi trabajo de titulación.

Quiero expresar mi profundo agradecimiento a mis padres Edwin y Alicia, así como a mi hermano Ismael por brindarme todo su apoyo tanto moral como económico, que ha sido fundamental en este camino. Su sabiduría, paciencia y consejos han sido pilares fundamentales que me han ayudado a crecer como persona y lograr alcanzar el éxito en esta nueva etapa de mi vida.

Un agradecimiento especial a mi amigo Oscar Ortega, que se ha transformado en un hermano que la experiencia académica me brindó. Su apoyo inquebrantable a lo largo de este recorrido, marcado por desafíos y momentos felices, ha sido invaluable.

De igual manera agradezco a mis buenos amigos César, Javier y Xavier por siempre estar presentes en todo momento, por brindarme su tiempo, conocimiento y ayuda durante este largo viaje que llevare siempre presente.

Edwin Alexander Castillo Tipantuña

## Índice de Contenidos

Dedicatoria .....	6
Agradecimiento.....	7
Resumen .....	16
Abstract.....	17
Capítulo I.....	18
Introducción.....	18
Antecedentes .....	19
Justificación e Importancia.....	20
Objetivo General.....	21
Actividades .....	21
Capítulo II .....	22
Metodología .....	22
<i>Materiales</i> .....	22
<i>Volcán Llaima</i> .....	22
<i>Monitorización del volcán Llaima</i> .....	23
<i>Metodología</i> .....	24
<i>Base de Datos Volcán Llaima</i> .....	25
<i>Preprocesamiento</i> .....	31
<i>Procesamiento</i> .....	35
<i>Entrenamiento</i> .....	41
Métricas de desempeño .....	44
Capítulo III.....	46
Resultados .....	46
<i>Filtrado y Normalización</i> .....	46
<i>Generación de Ruido y combinación con señal sísmica</i> .....	48
<i>Segmentación y Etiquetado</i> .....	49
<i>Balanceo de Datos</i> .....	51
<i>Partición de base de datos</i> .....	52
<i>Procesamiento de señales</i> .....	54
<i>Entrenamiento de modelos de ML</i> .....	58
<i>Test a mejor modelo ML</i> .....	83
<i>Interfaz Gráfica</i> .....	86

Capítulo IV.....	91
Conclusiones .....	91
Trabajos Futuros .....	93
Referencias .....	94

## Índice de Tablas

<b>Tabla 1</b> Clasificación de base de datos del volcán Llaima.....	26
<b>Tabla 2</b> Distribución de la base de datos .....	31
<b>Tabla 3</b> Particiones Hold Out a la base de datos balanceada.....	37
<b>Tabla 4</b> Resumen de características.....	39
<b>Tabla 5</b> Fórmulas para calcular las métricas de desempeño .....	45
<b>Tabla 6</b> Resultado de aplicar las 3 particiones tipo Hold Out.....	53
<b>Tabla 7</b> Tabla de la extracción de las 84 Características de las primeras 5 ventanas de 15 segundos.....	56
<b>Tabla 8</b> Resultados con la partición Hold Out 80 – 20 en los 3 modelos de ML en el dominio temporal.....	59
<b>Tabla 9</b> Resultados con la partición Hold Out 80–20 en los 3 modelos de ML en el dominio frecuencial .....	61
<b>Tabla 10</b> Resultados con la partición Hold Out 80–20 en los 3 modelos de ML, con las 84 características .....	63
<b>Tabla 11</b> Resultados con la partición Hold Out 70 – 30 en los 3 modelos de ML en el dominio temporal.....	65
<b>Tabla 12</b> Resultados con la partición Hold Out 70 – 30 en los 3 modelos de ML en el dominio frecuencial .....	67
<b>Tabla 13</b> Resultados con la partición Hold Out 70–30 en los 3 modelos de ML, con las 84 características .....	69
<b>Tabla 14</b> Resultados con la partición Hold Out 50 – 50 en los 3 modelos de ML en el dominio temporal.....	71

<b>Tabla 15</b> Resultados con la partición Hold Out 50–50 en los 3 modelos de ML en el dominio frecuencial .....	73
<b>Tabla 16</b> Resultados con la partición Hold Out 50–50 en los 3 modelos de ML, con las 84 características .....	75
<b>Tabla 17</b> Resultados con la partición Hold Out 20 – 80 en los 3 modelos de ML en el dominio temporal .....	77
<b>Tabla 18</b> Resultados con la partición Hold Out 20 – 80 en los 3 modelos de ML en el dominio frecuencial .....	79
<b>Tabla 19</b> Resultados con la partición Hold Out 20–80 en los 3 modelos de ML, con las 84 características .....	81
<b>Tabla 20</b> Test de Clasificación del Modelo DT .....	83
<b>Tabla 21</b> Test en Detección del Modelo DT .....	84

## Índice de Figuras

<b>Figura 1</b> Volcán Llaima.....	23
<b>Figura 2</b> Estaciones sísmicas del Volcán Llaima, la estación LAV marcada con un triángulo rojo. ....	24
<b>Figura 3</b> Diagrama General del Detector de microsismo del volcán Llaima .....	25
<b>Figura 4</b> VT del volcán Llaima en tiempo y frecuencia .....	27
<b>Figura 5</b> LP del volcán Llaima en tiempo y frecuencia.....	28
<b>Figura 6</b> TR del volcán Llaima en tiempo y frecuencia .....	29
<b>Figura 7</b> TC del volcán Llaima en tiempo y frecuencia .....	30
<b>Figura 8</b> Componente de 0.2 Hz presente en LP, TC, TR y VT .....	32
<b>Figura 9</b> Diagrama de bloques para la generación de la señal de ruido .....	34
<b>Figura 10</b> Diagrama de bloques de balanceo de datos.....	37
<b>Figura 11</b> Estructura del árbol de decisión .....	41
<b>Figura 12</b> Representación gráfica del modelo SVM .....	43
<b>Figura 13</b> Matriz de Confusión .....	44
<b>Figura 14</b> LP, TC, TR y VT en frecuencia al aplicar el filtro pasa banda.....	46
<b>Figura 15</b> LP, TC, TR y VT en tiempo al aplicar el filtro pasa banda .....	47
<b>Figura 16</b> LP, TC, TR y VT normalizados con z-score.....	47
<b>Figura 17</b> Señal de Ruido de 20 minutos generada .....	48
<b>Figura 18</b> Combinación microsismos - ruido .....	49
<b>Figura 19</b> (a) Ventana con microsismo (b) Ventana con señal de Ruido .....	49
<b>Figura 20</b> Segmentación y Etiquetado de un registro de 20 minutos .....	50
<b>Figura 21</b> Acercamiento al registro etiquetado y segmentado de 20 minutos .....	51
<b>Figura 22</b> Resultado de Balanceo de datos.....	51

<b>Figura 23</b> Distribución equitativa de etiquetas para entrenamiento en partición 80–20, 70-30, 50-50 y 20-80.....	53
<b>Figura 24</b> Distribución equitativa de etiquetas para validación en partición 80 - 20, 70 - 30, 50-50, 20-80 .....	54
<b>Figura 25</b> Base de datos en el dominio temporal .....	55
<b>Figura 26</b> PSD de ventanas con: (a) Microsismo (b) Señal de ruido .....	55
<b>Figura 27</b> Característica 16 – Entropía en frecuencia.....	57
<b>Figura 28</b> Normalización de las 84 características .....	58
<b>Figura 29</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo.....	59
<b>Figura 30</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM Tiempo.....	60
<b>Figura 31</b> Curva de desempeño k-NN:(a)A, P, R, S vs k Vecinos (b)BER vs k Vecinos	60
<b>Figura 32</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia.....	61
<b>Figura 33</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia .....	62
<b>Figura 34</b> Curva de desempeño A,P,R,S vs k Vecinos – k-NN Frecuencia.....	62
<b>Figura 35</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT 84 características .....	63
<b>Figura 36</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características	64
<b>Figura 37</b> Curva de desempeño A, P, R, S vs k Vecinos – k-NN 84 características .....	64
<b>Figura 38</b> Curva de desempeño BER vs MinLeafSize – DT Tiempo .....	65
<b>Figura 39</b> Curva de desempeño BER vs BoxConstraint – SVM Tiempo.....	66
<b>Figura 40</b> Curva de desempeño k-NN Tiempo (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos .....	66
<b>Figura 41</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia.....	67
<b>Figura 42</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia .....	68
<b>Figura 43</b> Curva de desempeño A, P, R, S vs k Vecinos – k-NN Frecuencia .....	68

<b>Figura 44</b> Curva de desempeño BER vs MinLeafSize – DT 84 características .....	69
<b>Figura 45</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características	70
<b>Figura 46</b> Curva de desempeño A, P, R, S vs k Vecinos – k-NN 84 características .....	70
<b>Figura 47</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo .....	71
<b>Figura 48</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM Tiempo.....	72
<b>Figura 49</b> Curva de desempeño A, P, R, S vs k Vecinos – k-NN Tiempo.....	72
<b>Figura 50</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia.....	73
<b>Figura 51</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia .....	74
<b>Figura 52</b> Curva de desempeño BER vs k Vecinos – k-NN Frecuencia.....	74
<b>Figura 53</b> Curva de desempeño BER vs MinLeafSize – DT 84 características .....	75
<b>Figura 54</b> Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características	76
<b>Figura 55</b> Curva de desempeño k-NN 84 características: (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos .....	76
<b>Figura 56</b> Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo.....	77
<b>Figura 57</b> Curva de desempeño BER vs BoxConstraint – SVM Tiempo.....	78
<b>Figura 58</b> Curva de desempeño BER vs k Vecinos – k-NN Tiempo .....	78
<b>Figura 59</b> Curva de desempeño BER vs MinLeafSize – DT Frecuencia.....	79
<b>Figura 60</b> Curva de desempeño BER vs BoxConstraint – SVM Frecuencia .....	80
<b>Figura 61</b> Curva de desempeño: (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos – k-NN Frecuencia.....	80
<b>Figura 62</b> Curva de desempeño BER vs MinLeafSize – DT 84 características .....	81
<b>Figura 63</b> Curva de desempeño BER vs BoxConstraint – SVM 84 características .....	82
<b>Figura 64</b> Curva de desempeño: (a)A, P, R, S vs BER (b) BER vs k Vecinos – k-NN 84 características.....	82

<b>Figura 65</b> Árbol de decisión del modelo DT con 84 características .....	83
<b>Figura 66</b> Detección a registro con 8 microsismos reales .....	84
<b>Figura 67</b> Detección a registro con 3 microsismos sintéticos.....	85
<b>Figura 68</b> Detección a registro del volcán Cotopaxi .....	85
<b>Figura 69</b> Detección a segundo registro del volcán Cotopaxi.....	86
<b>Figura 70</b> Presentación del programa de detección y clasificación de microsismos .....	86
<b>Figura 71</b> Pestaña de configuración de parámetros.....	87
<b>Figura 72</b> Gráfico del registro de 20 minutos .....	88
<b>Figura 73</b> Proceso de detección de los registros de 20 minutos .....	89
<b>Figura 74</b> Tabla con el resultado del proceso de detección y clasificación.....	90

## Resumen

Los fenómenos geológicos, como las erupciones volcánicas, son uno de los responsables de causar grandes pérdidas de vidas humanas. Por lo tanto, es crucial adoptar medidas preventivas, como la monitorización para reducir los impactos negativos de estos eventos naturales devastadores. En este contexto, en Chile se encuentra el volcán Llaima que es uno de los 4 volcanes más activos de América del Sur, por este motivo la Red Nacional de Vigilancia Volcánica de Chile (RNVV) a través de expertos del Observatorio Vulcanológico De los Andes del Sur (OVDAS) monitorizan y analizan la actividad sísmica de los volcanes, con la misión de proveer información oportuna respecto a la actividad volcánica y precautelar vidas en presencia de este tipo de desastres, sin embargo el análisis e interpretación de los datos recolectados son en grandes cantidades y su análisis tiene un nivel de complejidad alto por esta razón se buscan nuevos métodos que permitan realizar este proceso de una manera más rápida y eficiente. El trabajo de titulación presenta un algoritmo inteligente basado en *Machine Learning* tradicional que permite detectar microsismos del volcán Llaima en registros de 20 minutos. La base de datos utilizada consta de 3592 microsismos recopilados de la estación LAV durante los años 2010 y 2016. El detector se compone de 3 etapas, en la primera, realiza un preprocesamiento que incluye filtrado, normalización, generación de ruido y combinación Microsismo - Ruido. La segunda etapa implica el procesamiento que incluye técnicas de segmentación, etiquetado y balanceo. Por último, se lleva a cabo la etapa de entrenamiento de los algoritmos *k* Vecinos Cercanos, Máquina de Vector de Soporte y Árbol de Decisión con datos en el dominio temporal, dominio frecuencial y la extracción de las 84 características. El detector alcanza en clasificación un: 97.41% Exactitud, 92.73% Precisión, 93.97% Sensibilidad, 98.23% Especificidad, 0.038 en BER. En la fase de detección: 99.66% Exactitud, 99.26% Precisión, 99.26% Sensibilidad, 99.37% Especificidad, 0.006 en BER.

*Palabras Clave:* Microsismos, Machine Learning, Detección, monitorización, Llaima

## Abstract

Geological phenomena, such as volcanic eruptions, are among those responsible for causing great loss of human life. Therefore, it is crucial to adopt preventive measures such as monitoring to reduce the negative impacts of these devastating natural events. In this context, Chile has the Llaima volcano which is one of the 4 most active volcanoes in South America, for this reason the National Volcanic Surveillance Network of Chile (RNVV) through experts of the Southern Andes Volcanological Observatory (OVDAS) monitors and analyzes the seismic activity of volcanoes, However, the analysis and interpretation of the collected data are in large quantities and their analysis has a high level of complexity. For this reason, new methods are being sought to perform this process in a faster and more efficient way. The degree work presents an intelligent algorithm based on traditional Machine Learning that allows detecting micro-earthquakes of the Llaima volcano in records of 20 minutes. The database used consists of 3592 microseisms collected from the LAV station during the years 2010 and 2016. The detector is composed of 3 stages, in the first one, it performs a preprocessing that includes filtering, normalization, noise generation and Micro-Noise combination. The second stage involves processing that includes segmentation, labeling and balancing techniques. Finally, the training stage of the k-Nearest Neighbors, Support Vector Machine and Decision Tree algorithms is carried out with data in the time domain, frequency domain and the extraction of the 84 features. The detector achieves in classification a: 97.41% Accuracy, 92.73% Accuracy, 93.97% Sensitivity, 98.23% Specificity, 0.038 in BER. In the detection phase: 99.66% Accuracy, 99.26% Precision, 99.26% Sensitivity, 99.37% Specificity, 0.006 in BER.

*Keywords:* Microseismic, Machine Learning, Detection, monitoring, Llaima

## Capítulo I

### Introducción

La vulcanología estudia aspectos fundamentales de los volcanes, como su origen, estructura y composición, así como la evolución de su actividad eruptiva a lo largo del tiempo y los fenómenos vinculados con las erupciones volcánicas (Bordino, 2022). En este sentido, el cinturón de fuego del Pacífico, constituye una extensa zona sísmica y volcánica que se extiende a lo largo de 40 000 km alrededor del océano Pacífico y atraviesa alrededor de 29 países, destaca por ser la más extensa y activa de todo el planeta, al tener el 75% de los volcanes de todo el mundo (Equipo editorial Etecé, 2023).

Entre los volcanes que alberga el cinturón de fuego del Pacífico se encuentra el volcán Llaima que es uno de los volcanes más activos y voluminosos que tiene Sudamérica, esto ha llevado a instituciones como el Observatorio Vulcanológico de los Andes del Sur (OVDAS), ubicada en Chile a monitorizar el volcán de manera continua 24/7 los 365 días del año, donde un grupo de expertos se encarga de interpretar datos sobre la actividad sísmica que son obtenidos en grandes cantidades de instrumentos con un nivel de precisión alto como: sismómetros, sensores de radar, inclinómetros, cámaras de vigilancia, etc (SERNAGEOMIN, 2023).

En este contexto, la detección temprana y precisa de microsismos en volcanes es esencial para la gestión del riesgo ante una posible erupción volcánica, para lo cual se ha recurrido a sistemas de reconocimiento automático que permiten analizar grandes cantidades de datos y conviertan este proceso más rápido y efectivo. Existen propuestas de sistemas de reconocimiento automático que se basan en técnicas como Aprendizaje de Máquina (ML, del inglés *Machine Learning*) o Aprendizaje Profundo (DL, del inglés *Deep Learning*) para realizar las dos etapas fundamentales que componen a un sistema de reconocimiento que son la detección y clasificación de microsismos. Al tener bases de datos de microsismos limitadas,

mediante DL se ha recurrido a la generación de señales artificiales a partir de señales reales mediante la técnica de *Data Augmentation* (Awan, 2022), esto favorece al entrenamiento de los modelos de ML o DL debido a que se tiene disponible un conjunto más representativo de señales.

### **Antecedentes**

La detección automática de microsismos ha sido un campo de investigación activo y en constante evolución durante las últimas décadas. Numerosos investigadores han abordado el desafío de desarrollar sistemas capaces de identificar microsismos de manera eficiente y precisa. En este contexto, la técnica basada en deconvolución homomórfica y promedio a corto plazo sobre promedio a largo plazo (STA/LTA, del inglés *Short Time Average over Long Time Average*), es la que se utiliza comúnmente en la detección automática de microsismos. En este sentido, Lara et al. (2022), en su investigación desarrolla un sistema de detección de microsismos del volcán Cotopaxi basado en STA/LTA, en su estudio pusieron a prueba este sistema con diversos conjuntos de señales y con un registro de aproximadamente 5 días que contenía un total de 350 microsismos. Los resultados alcanzados fueron un 98 % de probabilidad de detección y 99 % de precisión.

Varios investigadores han propuesto técnicas distintas a STA/LTA para la detección automática de microsismos, como el de Rosero, K. (2020) que emplea la técnica de Detección por Actividad de la Voz (VAD, del inglés *Voice Activity Detection*), para la detección de microsismos del volcán Cotopaxi, donde obtiene resultados con una precisión del 100% y una exactitud del 99 %, por otra parte Bhatti et al. (2016), en su investigación propone un sistema automático de detección de microsismos del volcán Llaima mediante la técnica del Modelo de Markov Oculto (HMM, del inglés *Hidden Markov Model*) donde alcanza una exactitud de 94%.

Métodos como ML o DL de igual forma se han aplicado tanto en detección como en clasificación de microsismos. Altamirano, B. (2021), presenta un sistema que detecta y

clasifica de manera automática microsismos del volcán Cotopaxi, mediante ML tradicional, entre sus métricas de desempeño alcanzadas en detección son: 98% en exactitud, 95% en precisión y 0.017 en BER. R. A. Lara-Cueva et al. (2016) en su trabajo presenta un sistema de reconocimiento automático donde destaca el uso de la extracción de características en tiempo, frecuencia y escala, su detector arroja un 99% de precisión y sensibilidad. Canário et al. (2020), por su parte proporciona una nueva arquitectura para clasificación de microsismos del volcán Llaima denominada *SeismicNet* que está basada en Redes Neuronales Profundas (DNN, del inglés *Deep Neural Network*), donde obtiene tasas de clasificación superiores al 90%.

### **Justificación e Importancia**

Las erupciones volcánicas pueden llegar a producir daños irreparables, en este caso si un volcán entra en proceso eruptivo se puede tener consecuencias en las personas como: lesiones traumáticas, quemaduras, asfixia, problemas respiratorios e incluso hasta la muerte. (OPS, n.d.)

En Chile grupos de expertos del OVDAS o en Ecuador grupos de expertos del Instituto Geofísico de la Escuela Politécnica Nacional, se encargan de analizar individualmente cada microsismo y clasificarlo. Los microsismos son recolectados en grandes cantidades y su análisis tiene un nivel de complejidad alto (SERNAGEOMIN, 2023), además que requiere de una gran cantidad de tiempo y está expuesto a errores, en situaciones cuando el volcán entre en actividad se puede tener un rango de error más alto debido a que la cantidad de microsismos recibidos incrementa sustancialmente. Por ello disponer de un sistema de reconocimiento de microsismos automático tiene gran importancia debido a que se puede realizar este procedimiento de una manera más rápida y eficiente, lo que permite prevenir y salvaguardar la vida de la población de estas localidades, así como a reducir daños y pérdidas materiales.

**Objetivo General**

Crear algoritmos inteligentes basados en la teoría de Machine Learning tradicional para detección de eventos sísmicos en el volcán Llaima

**Actividades**

1. Identificación de las propiedades y variables del problema a tener en cuenta
2. Reunión de la base de datos representativa, con datos provistos por el observatorio Observatorio Volcanológico de los Andes del Sur.
3. Creación de una estructura de datos (temporales, espaciales, otros.) con un soporte común.
4. Identificación de la información a priori a incluir en los algoritmos, y creación de algoritmos de Machine Learning tradicional adaptados al problema.
5. Pruebas y evaluación del desempeño.

## Capítulo II

### **Metodología**

#### ***Materiales***

Para el desarrollo del presente trabajo de titulación se hace uso del software Matlab® versión R2022b, en un computador con las siguientes características: procesador Intel Core i3-4150T de tercera generación, 8 GB de memoria RAM.

#### ***Volcán Llaima***

La cordillera de los Andes es una cadena montañosa que se extiende a lo largo de Sudamérica, desde el oeste de Venezuela hasta la isla de Tierra de Fuego, la cual forma parte de la geología del conocido Cinturón de Fuego del Pacífico (Editorial Etecé, 2023).

El Cinturón de Fuego del Pacífico acoge al volcán Llaima (véase en la Figura 1), es uno de los volcanes más activos y voluminosos que tiene Sudamérica, se encuentra ubicado en la región de la Araucanía (Chile). Su registro eruptivo se encuentra documentado entre el año 1640 y 2009 con 48 eventos, durante los cuales se observaron la emisión de flujos de lava, formación de lahares, donde las principales comunidades amenazadas son Melipeuco, Captrén y Cherquenco (SERNAGEOMIN, n.d.).

**Figura 1**

*Volcán Llaima*



*Nota:* Recuperado de (Edmundo & Cox, 2003)

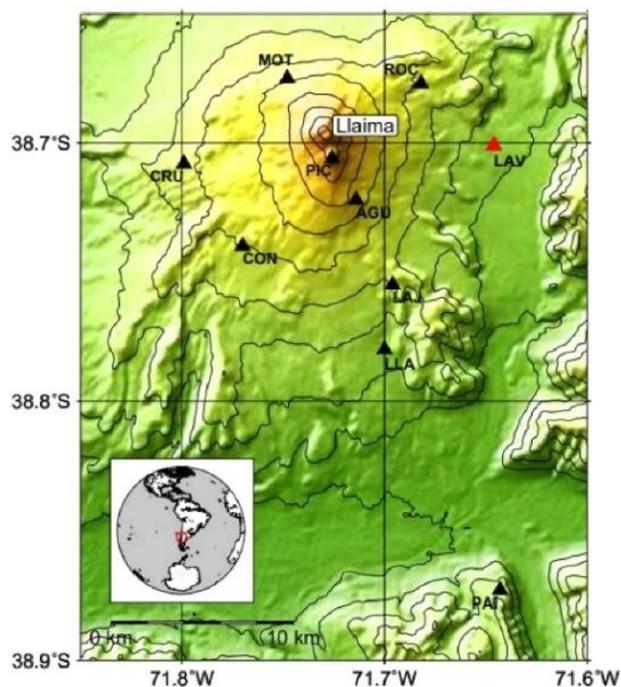
***Monitorización del volcán Llaima***

La monitorización de los 42 volcanes que posee Chile la realiza el OVDAS, donde la monitorización del volcán Llaima se realiza mediante 9 estaciones que recolectan datos de actividades sísmicas 24/7 (Curilem et al., 2018).

La Figura 2 presenta la ubicación y nombre de cada una de las 9 estaciones, entre ellas tenemos a la estación: CRU, MOT, ROC, CON, PIC, AGU, LAJ, LLA y la estación LAV resaltada en rojo, que es la estación de la cual se obtiene la base de datos.

## Figura 2

Estaciones sísmicas del Volcán Llaima, la estación LAV marcada con un triángulo rojo.



*Nota:* Recuperado de (Curilem et al., 2018)

## Metodología

El diagrama de bloques presentado en la Figura 3 ilustra las 3 etapas principales que componen el funcionamiento del detector. En la primera etapa, se realiza el preprocesamiento de la base de datos del volcán Llaima. Esto incluye operaciones como el filtrado y normalización. Además, durante esta fase se genera una señal de ruido, que posteriormente será combinada con los microsismos de la base de datos.

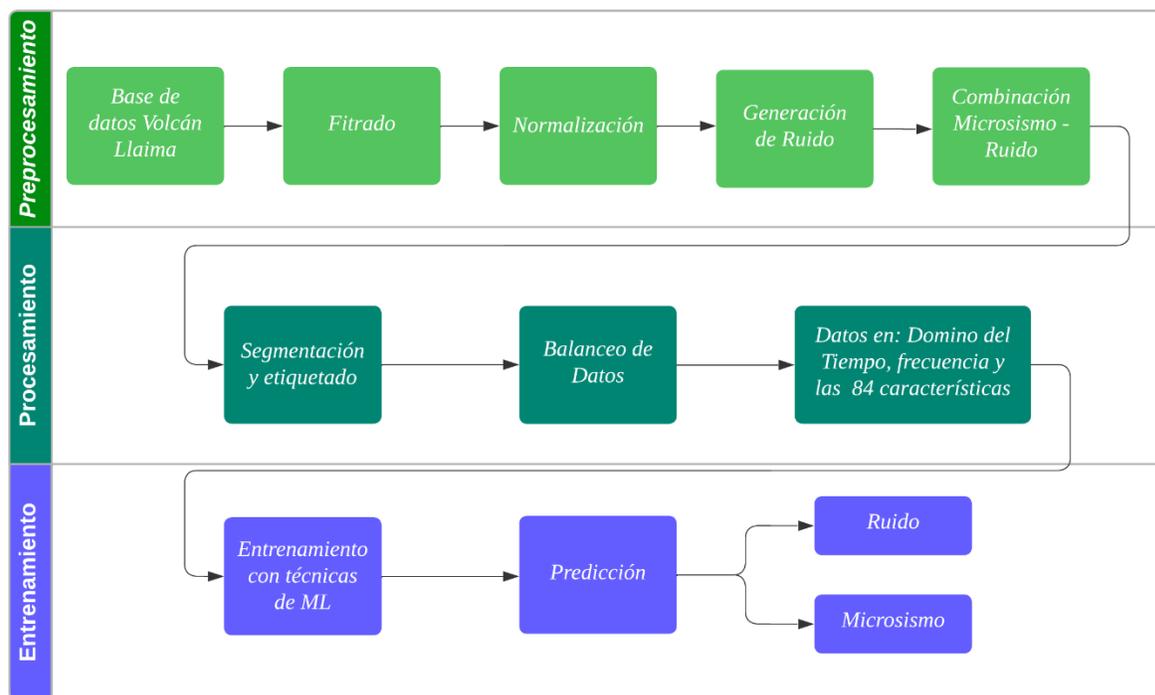
La segunda etapa, conocida como la etapa de procesamiento, se lleva a cabo la segmentación y etiquetado. Esto implica dividir el registro en ventanas a las cuales se les etiquetan como señal o ruido. Para asegurar la igualdad de la cantidad de datos correspondientes a señales y ruido, se realiza un balanceo de datos. A continuación, se crean 3 bases de datos distintas: una en el dominio del tiempo, otra en el dominio de la frecuencia

mediante la densidad espectral de potencia (PSD del inglés, *Power Spectral Density*) y la última que incluye la extracción de las 84 características.

La última etapa consiste en el entrenamiento de los modelos de ML. Durante esta fase se entrenan los modelos con las 3 bases de datos creadas y se selecciona el modelo que obtenga el mejor desempeño, dicho modelo es utilizado para predecir nuevos datos y detectar microsismos en registros de 20 minutos.

### Figura 3

Diagrama General del Detector de microsismo del volcán Llaima



### Base de Datos Volcán Llaima

La base de datos es obtenida de Canário et al., (2020) y es un conjunto de 3592 microsismos obtenidos de la estación LAV, que es una de las nueve estaciones de monitorización.

La base de datos fue recopilada durante los años 2010 y 2016 y los 3592 microsismos fueron clasificados de acuerdo con la Tabla 1.

**Tabla 1**

*Clasificación de base de datos del volcán Llaima*

<b>Tipo</b>	<b>Cantidad de señales</b>
VT	304
LP	1310
TR	490
TC	1488
<b>Total</b>	<b>3592</b>

La explicación a cada tipo de microsismo se presenta a continuación:

### **Volcano Tectónicos**

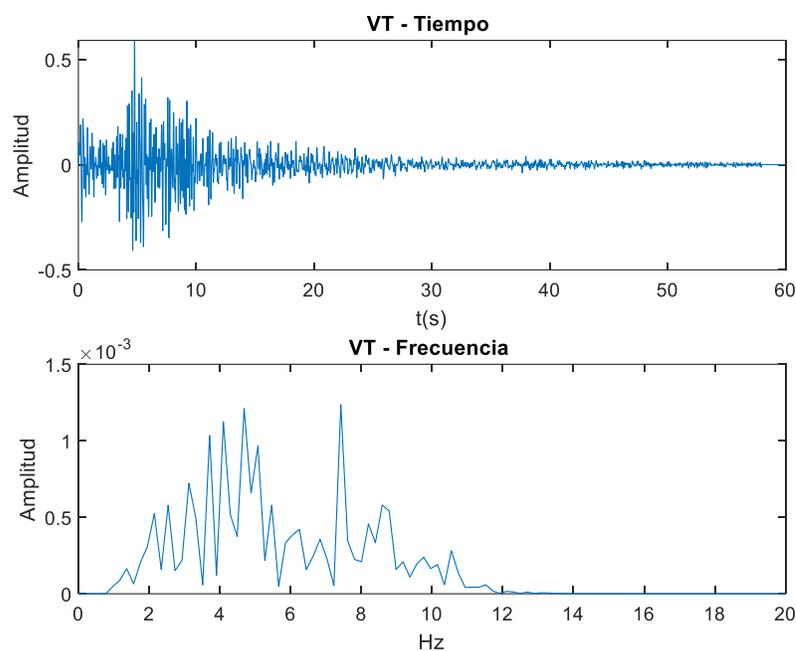
Los Volcano Tectónicos (VT, del inglés *Volcano-Tectonic*), contienen en su señal ondas tipo primaria (P) y tipo secundaria (S), que son usualmente las que indican el inicio de un VT (Ibañez & Carmona, 2000), las ondas se producen debido a la ruptura de rocas en los conductos volcánicos y sus componentes en frecuencia son hasta 10 Hz (Curilem et al., 2018) .

La duración de este tipo de microsismos no es fija, sino va desde los pocos segundos para microsismos pequeños hasta varios minutos para microsismos grandes (Ibañez & Carmona, 2000).

La Figura 4 presenta un ejemplo de un VT del volcán Llaima en tiempo como su espectro en frecuencia.

## Figura 4

VT del volcán Llaima en tiempo y frecuencia



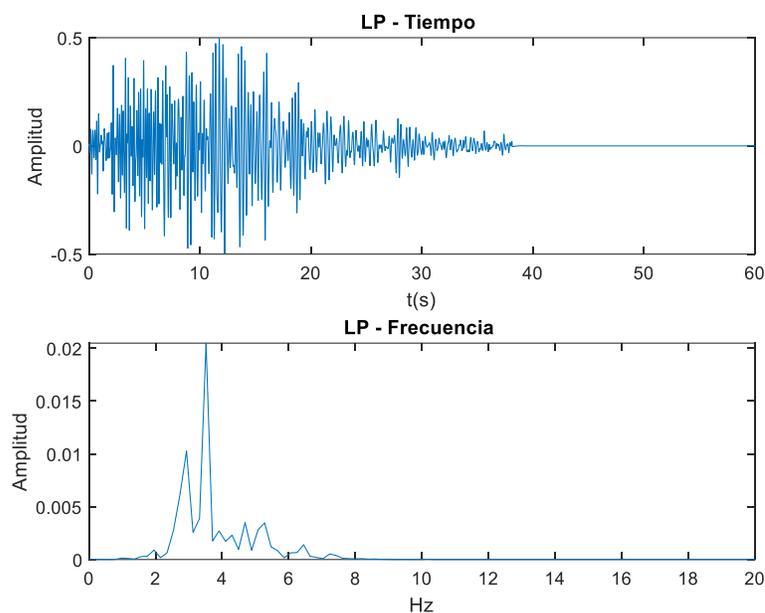
### Largo Periodo

Los Largo Periodo (LP, del inglés *Long Period*) son las más comunes dentro de un ambiente volcánico y suelen tener una duración que varía entre unos pocos segundos hasta poco más de un minuto (Ibañez & Carmona, 2000), se producen debido a un cambio temporal de presión que afecta al fluido dentro del conducto volcánico, lo que produce vibraciones que disminuyen y generan resonancia con frecuencias principales. Los LP del volcán Llaima contiene sus frecuencias principales dentro del rango de 0.5 y 5 Hz (Curilem et al., 2018).

La Figura 5 presenta un LP del volcán Llaima en tiempo como su espectro en frecuencia.

## Figura 5

*LP del volcán Llaima en tiempo y frecuencia*



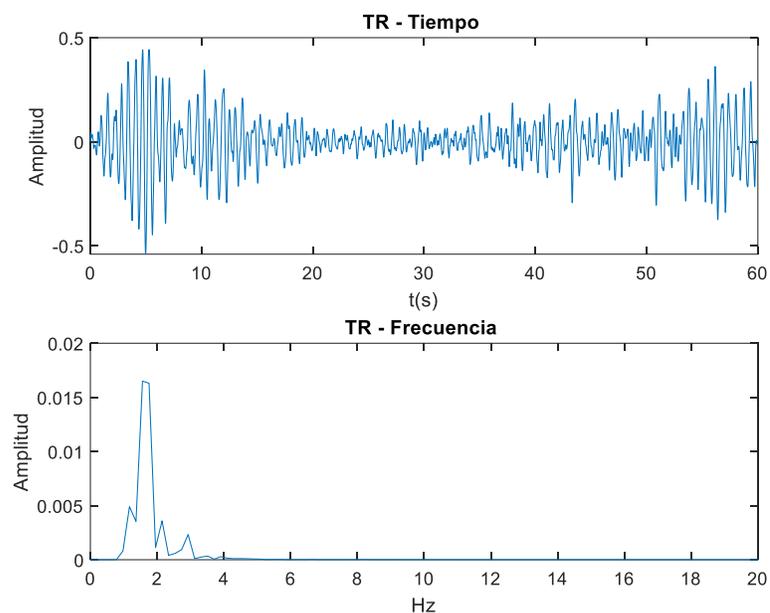
## Tremor

Los Tremor (TR, del inglés *Tremor*) a diferencia de los LP tienen una amplitud constante con una duración de tiempo entre minutos u horas (Ibañez & Carmona, 2000), se producen debido a una alteración constante de presión en el fluido magmático e hidrotermal. Los TR contiene sus frecuencias principales dentro del rango de 0.5 y 3 Hz (Curilem et al., 2018).

La Figura 6 presenta un TR del volcán Llaima en tiempo como su espectro en frecuencia, donde se observa que efectivamente sus picos de frecuencia están dentro del rango de 0.5 y 3 Hz.

## Figura 6

*TR del volcán Llaima en tiempo y frecuencia*



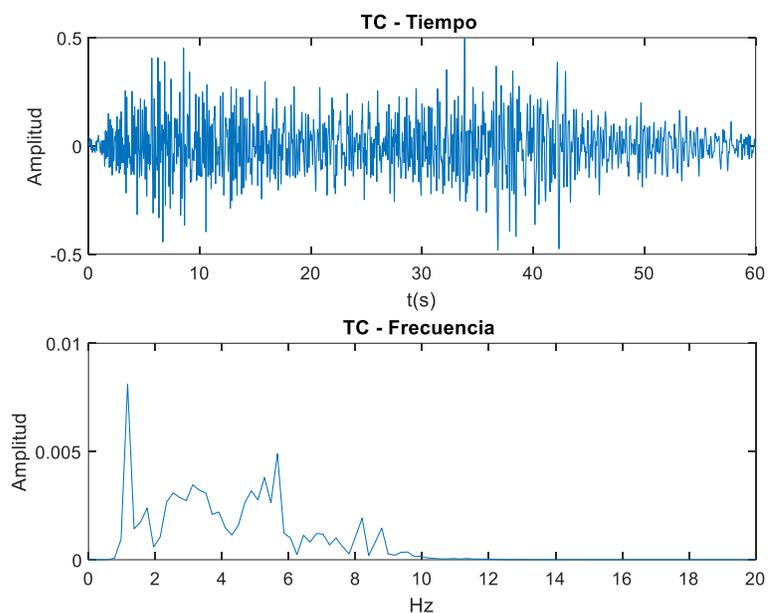
### Tectónicos

Tectónicos (TC, del inglés *Tectonic*), son aquellos que no son producidos por el volcán, sino que están vinculados con fallas geológicas, la forma en cómo se diferencia los VT de los TC es por el análisis en el tiempo de llegada de las ondas P y S (Curilem et al., 2018).

La Figura 7 presenta un TC del volcán Llaima en tiempo como su espectro en frecuencia.

## Figura 7

*TC del volcán Llaima en tiempo y frecuencia*



En la base de datos cada microsismo tiene una duración de 60 segundos y una frecuencia de muestreo de 100 Hz. La Tabla 2, presenta la distribución de la base de datos donde: “*Network*” indica el país de origen de la señal, “*Station*” indica el nombre de la estación de monitoreo, “*SampleRate*” es la frecuencia de muestreo del microsismo, “*Year*” el año o rango de año en el cual fue recogido el microsismo, “*Type*” es el tipo de evento como LP, TC, TR o VT, “*Duration*” hace referencia al tiempo que dura cada microsismo, “*StartPoint*” y “*EndPoint*” indican el punto de inicio y fin de la señal y por último “*Data*” es la columna que contiene al microsismo.

**Tabla 2***Distribución de la base de datos*

Network	Station	Sample Rate	Component	Year	Month	Type	Duration	StartPoint	EndPoint	Data
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	38.1100	1	3812	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	39.6700	1	3968	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	47.5300	1	4754	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	30.2500	1	3026	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	22.9900	1	2300	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	34.3200	1	3433	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	27.2200	1	2723	1x6000 double
"Chile"	"LAV"	100	"SHZ"	"2010-2016"	1	"LP"	37.9400	1	3795	1x6000 double

*Nota:* Se presenta un ejemplo de los 8 primeros microsismos de la base de datos que corresponden a microsismos del tipo LP

### **Preprocesamiento**

#### **Filtrado**

El proceso de filtrado de los microsismos es requerido debido a que al realizar el análisis espectral de cada uno de los microsismos del volcán Llaima se encuentra la presencia de un pico permanente en 0.2 Hz que de acuerdo con Kennett. (2001), se produce por microsismos generados por el choque del océano con el continente. En la Figura 8 se presenta la componente de 0.2 Hz.

La forma matemática para representar el proceso de filtrado es mediante el operador  $h_M\{s_M\}$  que es aplicado a cada microsismo  $s_M$  de la matriz  $S$ , donde  $M$  varía de 1 a 3592.

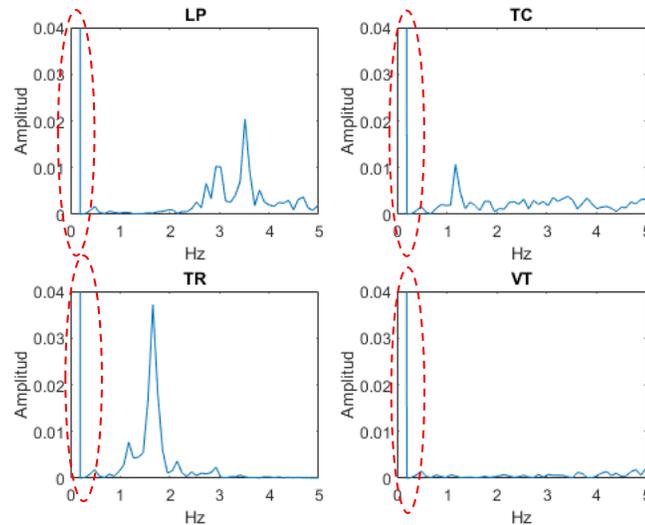
$$S = \{s_1^T, s_2^T, s_3^T, \dots, s_M^T\}^T, \quad (1)$$

El resultado se almacena en la matriz  $H$ , donde  $h_M$  corresponde al microsismo filtrado.

$$H = \{h_1^T, h_2^T, h_3^T, \dots, h_M^T\}^T, \quad (2)$$

### Figura 8

Componente de 0.2 Hz presente en LP, TC, TR y VT



### Normalización de base de datos

La normalización de la base de datos se utiliza porque es una técnica de preprocesamiento con gran importancia en ML, debido a que permite tener un mejor rendimiento y precisión de los modelos, tal como señala Paez. (2022), algunos entrenamientos de modelos de ML se realizan con el descenso de gradiente, donde se requiere que todas las señales estén en la misma escala, para un entrenamiento adecuado.

El método para estandarizar los microsismos consiste en remover la media y tener la desviación estándar igual a 1 (Altamirano, 2021), se representa mediante el operador matemático  $z_{S_M}\{h_M\}$  que es aplicado a cada microsismo filtrado  $h_M$  de la matriz  $H$ .

La ecuación (3) describe la estandarización z-score, donde,  $\mu_M$ ,  $\sigma_M$  y  $h_M$  corresponde a la media, desviación estándar y señal de un microsismo filtrado respectivamente.

$$z_{S_M} = \frac{h_M - \mu_M}{\sigma_M}, \quad (3)$$

El resultado se almacena en la matriz  $ZS$  donde  $z_{S_M}$  representa un microsismo estandarizado.

$$ZS = \{z_{S_1}^T, z_{S_2}^T, z_{S_3}^T, \dots, z_{S_M}^T\}^T, \quad (4)$$

El proceso de normalización se define mediante la ecuación (5), se representa mediante el operador matemático  $z1_M\{z_{S_M}\}$  que es aplicado a cada microsismo estandarizado  $z_{S_M}$  de la matriz  $ZS$ , donde  $\max |z_{S_M}|$  corresponde al valor máximo absoluto de un microsismo estandarizado.

$$z1_M = \frac{z_{S_M}}{\max |z_{S_M}|}, \quad (5)$$

El resultado se almacena en la matriz  $Z1$ , donde  $z1_M$  representa un microsismo normalizado dentro del rango deseado que es -1 y 1

$$Z1 = \{z1_1^T, z1_2^T, z1_3^T, \dots, z1_M^T\}, \quad (6)$$

### **Generación de ruido**

Para la detección se requieren registros de 20 minutos que estén compuestos de la combinación entre uno o varios microsismos y ruido, en este sentido la base de datos contiene únicamente microsismos de 60 segundos, por lo tanto, se crea una señal de ruido  $\eta$  de 20 minutos con el objetivo de combinar ambas señales y obtener los registros de 20 minutos.

**Figura 9**

Diagrama de bloques para la generación de la señal de ruido



La Figura 9 presenta el proceso para obtener la señal de ruido, se elige una señal de un microsismo de manera aleatoria entre el rango de 1 a 1500 que es en las cuales se puede visualizar mayor presencia de ruido, de cada una de ellas se extraen los primeros 0.2 segundos o las 20 primeras muestras. Estas muestras son almacenadas en un vector el cual acumula de 20 muestras en 20 muestras, ciclo que se repite hasta obtener un vector de 20 minutos o 120 000 muestras.

### Combinación Microsismo - Ruido

Con el fin de obtener registros de 20 minutos, se lleva a cabo el proceso de combinar ambos tipos de señales previamente obtenidas, lo cual se define en la ecuación (7).

$$Y = z1_M + \eta, \quad (7)$$

donde:

- $Y$ .- Es el registro de 20 minutos
- $z1_M$ .- Es el microsismo de 60 segundos normalizado
- $\eta$ .- Es la señal de ruido de 20 minutos.

Se generan 3592 señales de ruido con el objetivo de combinar cada uno de los microsismos, con un desplazamiento de 30 muestras que es acumulativo en cada combinación. Es decir, el primer microsismo se posiciona en la muestra 1 de la señal de ruido, el segundo

microsismo se desplaza 30 muestras y se coloca en la posición 31 en la segunda señal de ruido, el tercer microsismo de desplaza 30 muestras y se coloca en la posición 61 en la tercera señal de ruido.

Este proceso de combinación y desplazamiento de las señales permite obtener los registros finales de 20 minutos que están compuestos por la combinación adecuada de los microsismos y el ruido correspondiente.

## **Procesamiento**

### **Segmentación y Etiquetado**

Los registros se segmentan en ventanas, donde se adopta el criterio empleado en la investigación de R. A. Lara-Cueva et al., (2016). En su investigación, emplearon ventanas de varios tamaños, que variaron entre 5 y 200 segundos. El valor óptimo encontrado fue de 15 segundos, por lo tanto, se sigue la misma pauta y se emplean ventanas de 15 segundos para segmentar los registros de 20 minutos. Por cada registro  $Y$  segmentado se obtiene un total de 80 ventanas que son almacenados en el vector  $v_i$ , como se presenta en la ecuación (8).

$$v_i = \{v_{i,1}^T, v_{i,2}^T, v_{i,3}^T, \dots, v_{i,80}^T\}^T, \quad (8)$$

donde  $v_{i,j}$  representa un segmento de 15 segundos de la señal  $Y$ . Cada vector es almacenado en la matriz  $V$ , donde  $v_M$  es un vector de cada registro segmentado.

$$V = \{v_1^T, v_2^T, v_3^T, \dots, v_M^T\}^T, \quad (9)$$

El etiquetado de las ventanas se lleva a cabo en función de la relación señal-ruido (SNR, del inglés *Signal to Noise Ratio*). Para ello, se calcula la SNR de cada ventana de 15 segundos respecto a un vector de ruido de prueba de la misma longitud. El criterio utilizado para la etiquetación es el siguiente: Si la SNR es mayor o igual a 3 dB, se etiqueta la ventana como "+1", lo cual indica que contiene un microsismo. Por otro lado, si la SNR es menor a 3 dB,

la ventana se etiqueta como “-1” lo cual indica que se trata de una señal de ruido. Las etiquetas  $l_{i,j}$  de cada segmento  $v_{i,j}$  son almacenada en el vector  $L_i$ .

$$L_i = \{l_{i,1}, l_{i,2}, l_{i,3}, \dots, l_{i,80}\} \quad , \quad (10)$$

Una vez que se obtiene cada vector  $L_i$ , este se almacena en la matriz  $L$ , como se presenta en la ecuación (9).

$$L = \{L_1^T, L_2^T, L_3^T, \dots, L_M^T\}^T \quad , \quad (11)$$

donde  $L_M$  es un vector de etiquetas de cada registro segmentado.

### **Balanceo de Datos**

El balanceo de datos es un procedimiento importante para obtener mejores resultados en las métricas de desempeño de un modelo de ML, al tener un equilibrio en los datos el modelo no presenta un favoritismo hacia una clase (Hvilshoj, 2022).

La Figura 10 presenta el procedimiento de 5 pasos para el balanceo de datos, se realiza debido a que la etiqueta mayoritaria corresponde a las ventanas de ruido, es decir, las etiquetas con “-1”. Su procedimiento consiste en primer lugar, identificar la ubicación y el número de etiquetas con “+1”, ya que es la etiqueta con minoría de datos. Posterior se busca el mismo número de etiquetas “+1” en etiquetas con “-1”, se unen ambas etiquetas y se buscan las ventanas de acuerdo con la posición identificada de las etiquetas. Por último, se mezclan estas ventanas de manera aleatoria.

**Figura 10**

Diagrama de bloques de balanceo de datos



### Partición de datos

Se implementa el método de partición aleatoria de validación cruzada *Hold Out* a los datos balanceados. El proceso implica dividir la base de datos en dos conjuntos distintos: un conjunto de entrenamiento y un conjunto de validación, en donde se debe mantener la proporción balanceada de etiquetas “+1” y “-1” en cada partición. La Tabla 3 presenta las 4 particiones a realizar a la base de datos balanceada.

**Tabla 3**

Particiones *Hold Out* a la base de datos balanceada

Partición	Entrenamiento %	Validación %
1	80	20
2	70	30
3	50	50
4	20	80

### Procesamiento de señales

Cada conjunto de datos de entrenamiento y validación, son utilizados para generar 3 bases de datos distintas cada una de ellas con características particulares que son utilizadas para la construcción de los modelos.

### ***Base de datos en el dominio temporal***

La primera base de datos se mantiene en su forma original es decir en el dominio temporal.

### ***Base de datos en el dominio frecuencial***

La segunda base de datos se obtiene al aplicar el método de la PSD de Welch a las ventanas de 15 segundos, se implementa debido a que en la en la investigación de R. Lara-Cueva et al., (2020), fue el método que con una resolución media de NFFT = 512 presentó los mejores resultados en comparación con los métodos de Yule – Walker y Burg.

Por cada segmento  $v_{i,j}$  que se aplica la PSD Welch se obtiene un total de 257 muestras que son almacenados en el vector  $p_M$ , y almacenados en la matriz P como se presenta en la ecuación (8).

$$P = \{p_1^T, p_2^T, p_3^T, \dots, p_M^T\}^T, \quad (12)$$

### ***Base de datos mediante la extracción de las 84 características***

La tercera base de datos se genera mediante el cálculo de las 84 características que fueron obtenidas de Pérez et al. (2020), se utiliza debido a que fue el método con el que Altamirano, B. (2021) obtuvo uno de los modelos óptimos empleados en su sistema de votación. Por cada ventana de 15 segundos se obtienen las 84 características más relevantes entre tiempo (13 características), frecuencia (21 características) y escala (50 características).

Por cada segmento  $v_{i,j}$  que se aplica la extracción de las 84 características se obtiene un total de 84 muestras que son almacenados en el vector  $c_M$ , y almacenados en la matriz C como se presenta en la ecuación (8).

$$C = \{c_1^T, c_2^T, c_3^T, \dots, c_M^T\}^T, \quad (13)$$

Las 84 características se presentan en la Tabla 4 .

Tabla 4

## Resumen de características

Dominio del tiempo		Dominio de la frecuencia		Dominio de la escala	
ID	Característica	ID	Característica	ID	Característica
f1	Media	f29	Potencia	f57	Porcentaje de energía para D1
f2	Desviación Estándar	f30	Densidad de pico por encima del valor RMS	f58	Porcentaje de energía para D2
f3	Varianza	f31	2do valor pico más alto	f59	Porcentaje de energía para D3
f4	Entropía	f32	Freq. del 2do pico más alto	f60	Porcentaje de energía para D4
f5	Kurtosis	f33	3er valor pico más alto	f61	Porcentaje de energía para D5
f6	Entropía Multiescala (MSE)	f34	Freq. del 3er pico más alto	f62	Porcentaje de energía para D6
f7	Tiempo para alcanzar el pico máximo	<b>Dominio de la escala</b>		f63	A6 RMS en el dominio del tiempo
f8	Valor RMS	f35	A6 Máx. pico en el dominio de la freq.	f64	D1 RMS en el dominio del tiempo
f9	Valor de Pico a Pico	f36	D1 Máx. pico en el dominio de la freq.	f65	D2 RMS en el dominio del tiempo
f10	Relación pico – RMS	f37	D2 Máx. pico en el dominio de la freq.	f66	D3 RMS en el dominio del tiempo
f11	Energía	f33 8	D3 Máx. pico en el dominio de la freq.	f67	D4 RMS en el dominio del tiempo
f12	Tasa de cruce por cero	f39	D4 Máx. pico en el dominio de la freq.	f68	D5 RMS en el dominio del tiempo
f13	Densidad de pico por encima del valor RMS	f40	D5 Máx. pico en el dominio de la freq.	f67	D6 RMS en el dominio del tiempo
<b>Dominio de la frecuencia</b>		f41	D6 Máx. pico en el dominio de la freq.	f70	A6 Pico a Pico en el dominio del tiempo
f14	Frecuencia de pico máximo	f42	A6 Freq de pico máx.	f71	D1 Pico a Pico en el dominio del tiempo
f15	Ancho de banda del 90% de energía	f43	D2 Freq de pico máx.	f72	D2 Pico a Pico en el dominio del tiempo

<b>Dominio del tiempo</b>		<b>Dominio de la frecuencia</b>		<b>Dominio de la escala</b>	
ID	Característica	ID	Característica	ID	Característica
f16	Entropía	f44	D3 Freq de pico máx.	f73	D3 Pico a Pico en el dominio del tiempo
f17	Media	f45	D4 Freq de pico máx.	f74	D4 Pico a Pico en el dominio del tiempo
f18	Desviación Estándar	f46	D5 Freq de pico máx.	f75	D5 Pico a Pico en el dominio del tiempo
f19	Varianza	f47	D6 Freq de pico máx.	f76	D6 Pico a Pico en el dominio del tiempo
f20	Energía	f48	A6 Media en el dominio de la freq	f77	A6 Relación pico a RMS en el dominio del tiempo
f21	Kurtosis	f49	D1 Media en el dominio de la freq	f78	D1 Relación pico a RMS en el dominio del tiempo
f22	Entropía Multiescala	f50	D2 Media en el dominio de la freq	f79	D2 Relación pico a RMS en el dominio del tiempo
f23	Pico máximo en la banda de 10-20 Hz	f51	D3 Media en el dominio de la freq	f80	D3 Relación pico a RMS en el dominio del tiempo
f24	Freq de pico máx. en la banda de 10 – 20 Hz	f52	D4 Media en el dominio de la freq	f81	D4 Relación pico a RMS en el dominio del tiempo
f25	Pico máximo en la banda de 20-30 Hz	f53	D5 Media en el dominio de la freq	f82	D5 Relación pico a RMS en el dominio del tiempo
f26	Freq de pico máx. en la banda de 20 – 30 Hz	f54	D6 Media en el dominio de la freq	f83	D6 Relación pico a RMS en el dominio del tiempo
f27	Valor RMS	f55	Energía media de los componentes	f84	Energía media de los coeficientes wavelet
f28	Relación pico – RMS	f56	Porcentaje de energía para A6		

*Nota:* Recuperado de (Pérez et al., 2020)

## Entrenamiento

Para cada una de las bases de datos propuestas, se implementan tres modelos de ML supervisado. La búsqueda de la configuración óptima se realiza a través de la variación de uno o varios parámetros dentro de cada modelo.

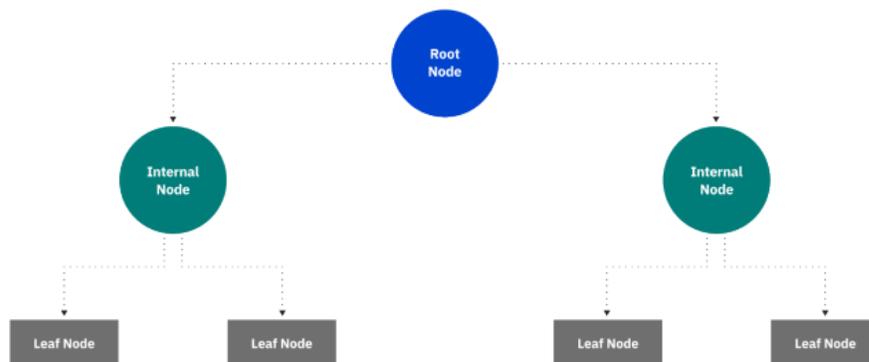
En cada caso, se elige el valor del parámetro que maximiza las métricas de desempeño.

## Árbol de Decisión

El algoritmo de árbol de decisión (DT, del inglés *Decision Tree*), es un algoritmo de aprendizaje supervisado que es utilizado tanto para clasificación como para regresión, su estructura es la de un árbol como se muestra en la Figura 11 y está conformado por: un nodo raíz, nodos de decisión y nodos terminales u hojas.

**Figura 11**

*Estructura del árbol de decisión*



*Nota:* Recuperado de (IBM, n.d.)

El algoritmo crea un árbol de decisiones mediante el proceso de división recursiva, donde en cada paso el algoritmo divide los datos en subgrupos más pequeños en función de una característica relevante. La eficiencia de la predicción se basa en reducir la impureza de los nodos del árbol, es decir el algoritmo determina una característica idónea y busca dividir los

datos de manera que los subgrupos resultantes sean lo más homogéneos posible en términos de su clase o categoría. La impureza de un nodo se calcula mediante la entropía que está definida en la ecuación (14), a través de la entropía el algoritmo puede evaluar diferentes características, donde su elección se inclina por la característica que resulte con el menor valor (Moreno, 2021), o mediante el índice Gini definido en la ecuación (15).

$$Entropía(S) = \sum_{i=1}^n -p_i \cdot \log_2(p_i), \quad (14)$$

donde:

- $S$ .- Conjunto de datos de un nodo
- $p_i$ .- Es la proporción de instancias en  $S$  que pertenecen a la clase  $i$
- $n$ .- Es el número de clases diferentes en  $S$

$$Gini(p) = 1 - \sum_{i=1}^c (p_i)^2 \quad (15)$$

donde:

- $Gini(p)$ .- índice de Gini para el nodo
- $p_i$ .- Es la proporción de ejemplos pertenecientes a la clase  $i$  en el nodo
- $c$ .- Es el número de clases en el nodo

En Matlab® se varía el parámetro “*minleafsize*”. Este parámetro indica el número mínimo de observaciones por hoja del árbol. El resto de parámetros se utilizan por defecto.

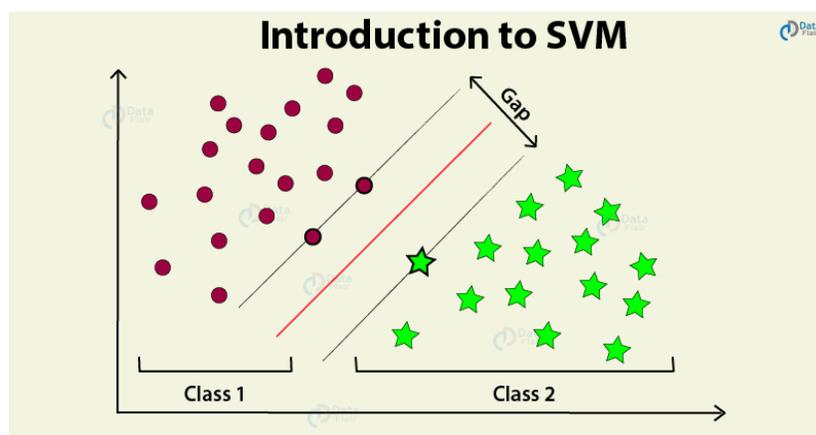
### **Máquina de Vector de Soporte**

El algoritmo Máquina de Vector de Soporte (SVM del inglés *Support Vector Machine*) es un algoritmo de aprendizaje supervisado que al igual que DT es utilizado tanto para problemas de clasificación como para regresión. El modelo SVM ubica el conjunto de datos en el espacio y los separa en dos clases mediante un hiperplano de separación denominado vector de soporte

esto se presenta en la Figura 12, al llegar nuevas muestras al modelo, debido a la separación en los dos espacios se podrá ubicar y saber a qué clase pertenece esta muestra (Mendoza, 2020).

### Figura 12

Representación gráfica del modelo SVM



Nota: Recuperado de (DataFlair, n.d.)

El parámetro variable dentro de este modelo es *BoxConstraint*. El resto de parámetros se utilizan por defecto.

### **k Vecinos Cercanos**

El algoritmo *k Vecinos Cercanos* (*k*-NN, del inglés *k Nearest Neighbors*) es un algoritmo de aprendizaje supervisado que al igual que DT y SVM es utilizado tanto para problemas de clasificación como para regresión. Este método se centra en encontrar los *k* vecinos más cercanos a un punto específico. En su forma más general el funcionamiento consiste en calcular la distancia entre cada elemento y el punto de interés, la métrica de distancia puede ser de tipo euclídeana. Luego se ordena en orden ascendente y se seleccionan los *k* elementos con las distancias más pequeñas. Por último, se genera un sistema de votación y se asigna al nuevo dato la etiqueta más común entre ellos cuando se trata de clasificación (Díaz, n.d.).

Los parámetros variables dentro de este modelo son:” *NumNeighbors*” y “*Distance*”. Se utilizan las métricas de distancia “*euclidean*” y “*cosine*”. El resto de parámetros se utilizan por defecto.

## Métricas de desempeño

Una de las decisiones más importantes a las que se enfrenta al resolver problemas de clasificación, como en este caso un programa de detección basado en clasificación, es determinar qué modelo entrenado es el más idóneo a las necesidades del problema propuesto. La matriz de confusión es la herramienta que nos ayuda en esta elección de modelo, en la Figura 13, se presenta la matriz de confusión para un modelo de dos clases que corresponde a una matriz de 2x2, está compuesta por Verdaderos Positivos (TP, del inglés *True Positive*), Verdaderos Negativos (TN, del inglés *True Negative*), Falso Negativo (FN, del inglés *False Negative*), Falso Positivo (FP, del inglés *False Positive*).

### Figura 13

*Matriz de Confusión*

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

*Nota:* Recuperado de (Mohajon, 2020)

Esta matriz nos brinda una visión de qué tan preciso es nuestro modelo al realizar la clasificación. En ella se registran tanto los aciertos como los errores cometidos para cada una de las categorías (Sotaquirá, 2022).

A partir de la matriz de confusión se puede hallar las métricas de desempeño que son utilizadas en el presente trabajo como Exactitud (A, del inglés *Accuracy*), Precisión (P, del inglés *Precision*), Sensibilidad (R, del inglés *Recall*), Especificidad (S, del inglés *Specificity*) y la Tasa de Error de Bit (BER, del inglés *Bit Error Rate*), la Tabla 5 presenta las ecuaciones para calcular cada una de estas métricas que fueron extraídas de (Altamirano, 2021).

**Tabla 5**

*Fórmulas para calcular las métricas de desempeño*

<b>Métrica de Desempeño</b>	<b>Fórmula</b>
Exactitud (%)	$A = \frac{TP + TN}{TP + TN + FP + FN} \times 100$
Precisión (%)	$P = \frac{TP}{TP + FP} \times 100$
Sensibilidad (%)	$R = \frac{TP}{TP + FN} \times 100$
Especificidad (%)	$S = \frac{TN}{TN + FP} \times 100$
BER	$BER = 1 - \frac{R + S}{200}$

## Capítulo III

### Resultados

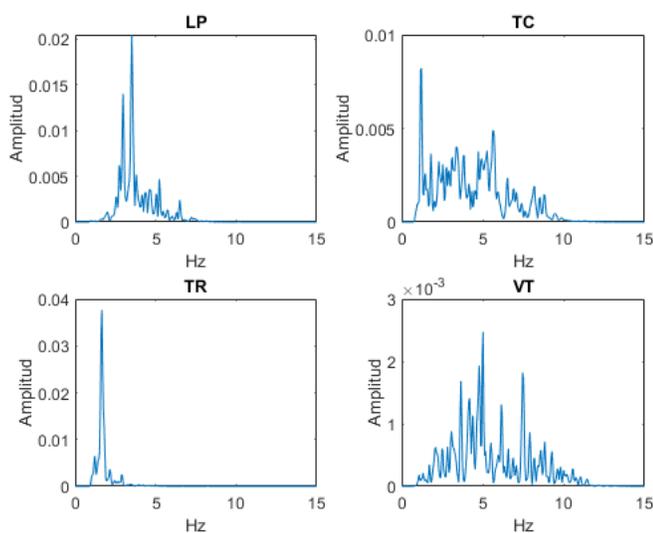
En esta sección se exponen los resultados alcanzados al utilizar la metodología descrita en cada uno de los puntos del tercer capítulo.

#### **Filtrado y Normalización**

Se aplica un filtro pasa banda de tipo FIR de orden 256, con  $f_{min} = 0.7 \text{ Hz}$  y  $f_{sup} = 49,5 \text{ Hz}$ , a los 3592 microsismos del volcán Llaima, el resultado de aplicar el filtro se presenta en la Figura 14, se observa un ejemplo por cada tipo evento, la componente de 0.2 Hz mostrada en la Figura 8 se ha eliminado de forma correcta.

#### **Figura 14**

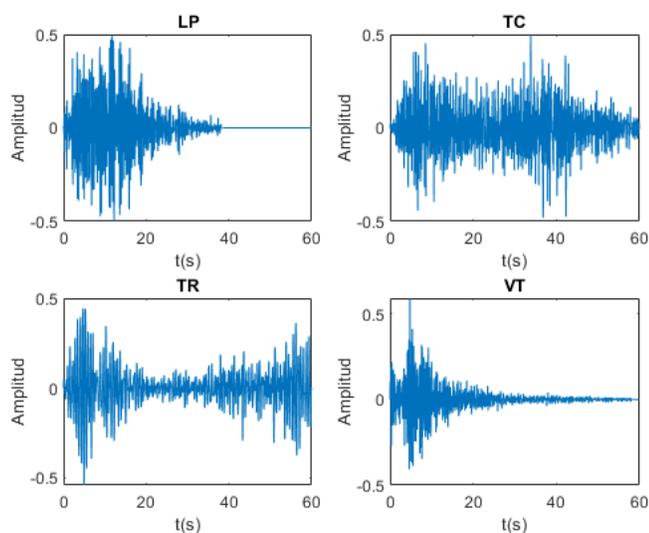
*LP, TC, TR y VT en frecuencia al aplicar el filtro pasa banda*



En el dominio del tiempo, podemos observar que al eliminar la componente de 0.2 Hz, los microsismos se centraron aproximadamente en 0, esto se presenta en la Figura 15.

**Figura 15**

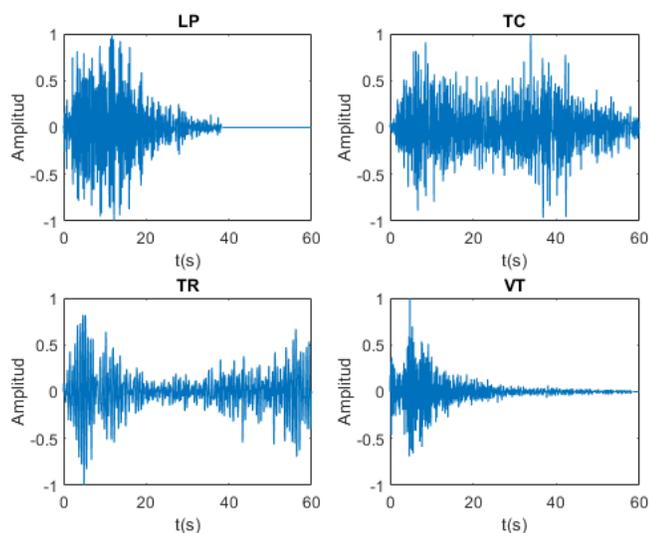
*LP, TC, TR y VT en tiempo al aplicar el filtro pasa banda*



Dentro del proceso de normalización de los microsismos se obtiene el resultado de la Figura 16, se observa que el rango de amplitud en el cual se encuentra cada microsismo está entre -1 y 1.

**Figura 16**

*LP, TC, TR y VT normalizados con z-score*

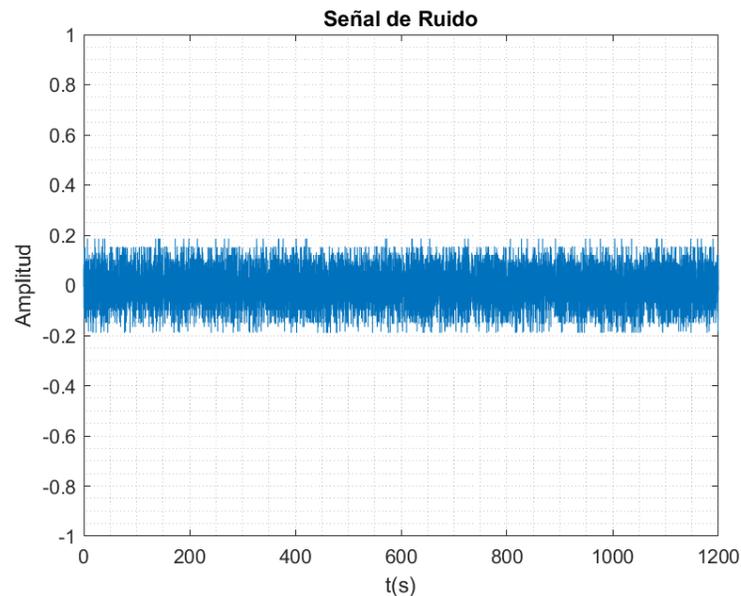


### **Generación de Ruido y combinación con señal sísmica**

La señal de ruido generada se presenta en la Figura 17, se aplica el procedimiento presentado en la Figura 9, la elección de los microsismos fue de manera aleatoria mediante la función *rand* en Matlab®, esto se realiza hasta obtener los 20 minutos o las 120 000 muestras.

#### **Figura 17**

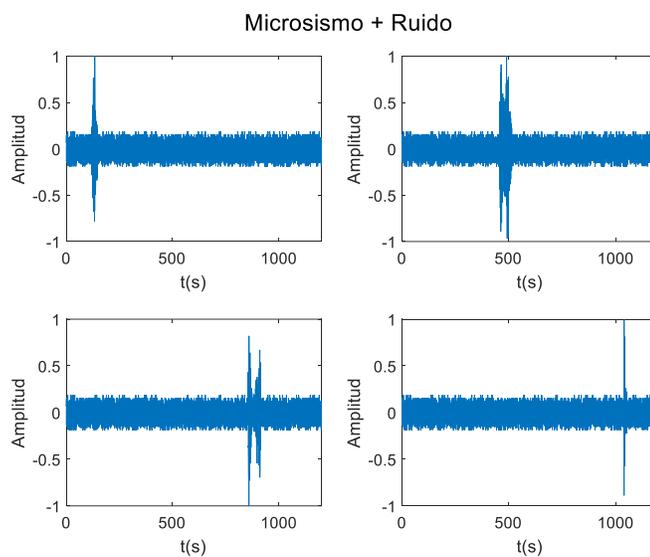
*Señal de Ruido de 20 minutos generada*



A continuación, se realiza una copia de la señal de ruido por cada uno de los microsismos, es decir se obtienen 3592 señales de ruido. El resultado de combinar ambas señales se presenta en la Figura 18, como se puede observar los microsismos están desplazados dentro de la señal de ruido, lo que genera el registro de 20 minutos requerido presentado en la ecuación (7).

## Figura 18

*Combinación microsismos - ruido*

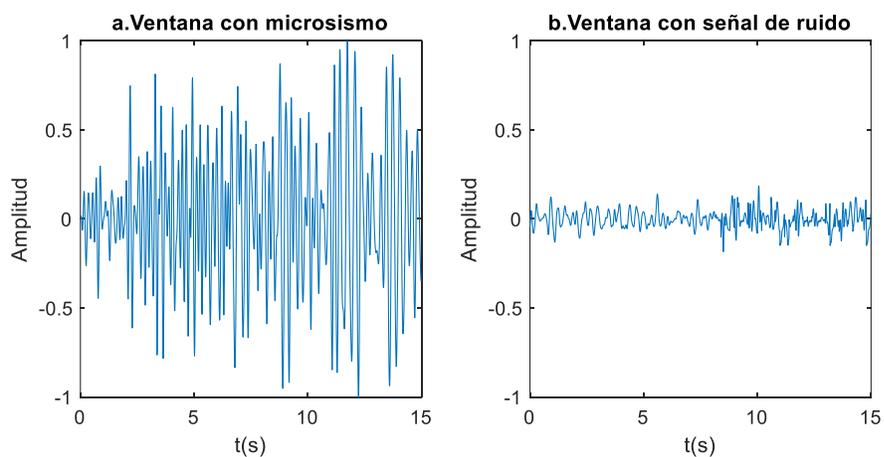


## Segmentación y Etiquetado

A cada uno de los registros, se realiza el proceso de segmentación en ventanas de 15 segundos, la Figura 19 presenta dos ventanas de 15 segundos, una que contiene señal de un microsismo y la otra señal de ruido.

## Figura 19

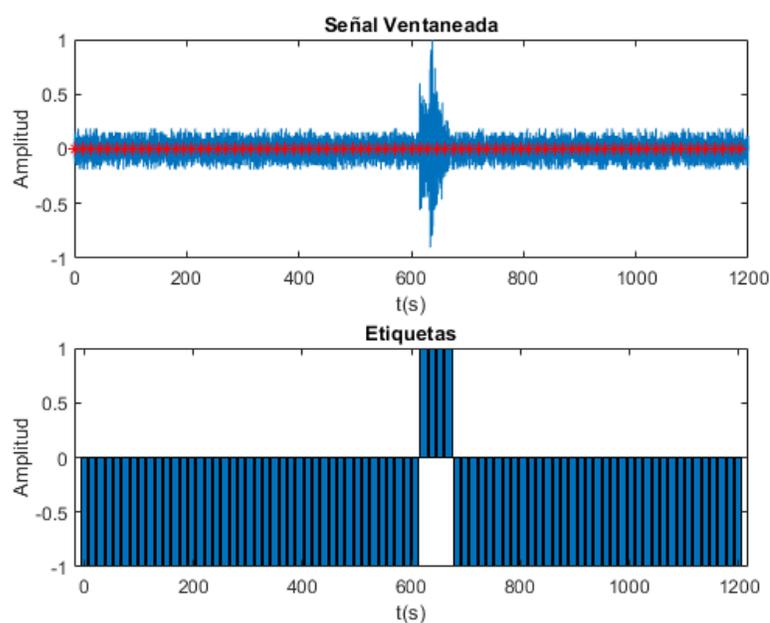
*(a) Ventana con microsismo (b) Ventana con señal de Ruido*



La Figura 20 presenta el proceso de segmentación y etiquetado de un registro. Para este registro se obtiene un total de 80 ventanas cada una de 15 segundos, a cada una de ellas se realiza el proceso de etiquetado donde se considera el criterio de la SNR propuesto, se obtiene un total de 4 etiquetas con “+1” que indican que esas ventanas contienen señal de un microsismo y 76 etiquetas con “-1” que indican que esas ventanas contienen señal de ruido.

### Figura 20

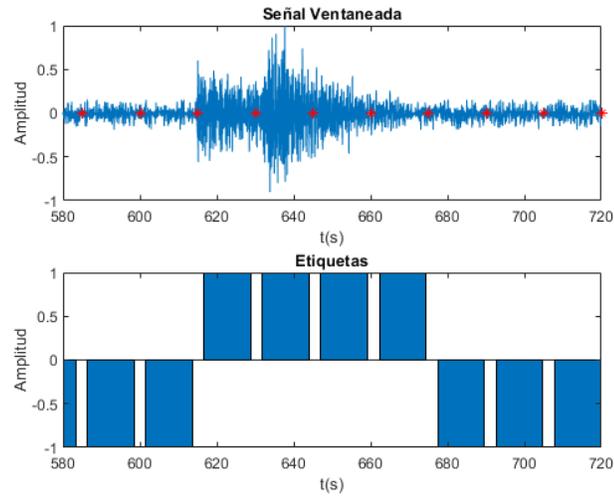
*Segmentación y Etiquetado de un registro de 20 minutos*



Un acercamiento a la Figura 20 se presenta en la Figura 21, donde se aprecia el etiquetado correcto del registro de 20 minutos, que contiene únicamente 4 ventanas con señal de un microsismo. Esto permite afirmar que el criterio propuesto de la relación SNR es válido.

## Figura 21

Acercamiento al registro etiquetado y segmentado de 20 minutos



Una vez segmentados y etiquetados los 3592 registros se obtiene un total de 287 360 ventanas con sus respectivas etiquetas.

## Balanceo de Datos

La Figura 22 presenta los resultados obtenidos al aplicar los 5 pasos para realizar el balanceo de datos.

## Figura 22

Resultado de Balanceo de datos

Workspace	
Name ▲	Value
DataBalan	23936x1500 double
Datamez	23936x1501 double
EtiqBalan	23936x1 double
findRui	11968x1 double
findSen	11968x1 double
Posiciones	23936x1 double

La explicación a cada variable de la Figura 22 es la siguiente:

1. **Identificación de etiquetas +1:** Se encuentra 12 423 posiciones de etiquetas que contienen "+1", esto se observa en la variable findSen.
2. **Identificación de etiquetas -1:** Se busca y encuentra un total de 12 423 posiciones de etiquetas que contienen "-1", esto se observa en la variable findRui. Esta longitud es debido a que se debe igualar el número de etiquetas "+1" con las de "-1".
3. **Unión de etiquetas:** La variable Posiciones contiene la unión de las dos búsquedas en un solo vector, y se obtiene un total de 24 846 posiciones de etiquetas entre "+1" y "-1".
4. **Búsqueda de ventanas:** De acuerdo con las posiciones almacenadas en la variable Posiciones, se busca las ventanas correspondientes y se almacenan en una matriz, la cual tiene un tamaño de 24846 x 1500. Se replica el mismo procedimiento para encontrar las etiquetas, el tamaño del vector de etiquetas es 24846x 1. Las variables que contienen estos resultados son DataBalan y EtiqBalan respectivamente.
5. **Mezcla de datos:** Se mezcla la matriz de forma aleatoria, el resultado se almacena en la variable Datamez.

### ***Partición de base de datos***

La base de datos balanceada es utilizada para realizar las tres particiones *Hold Out* para obtener el conjunto de datos de entrenamiento y el conjunto de datos de validación, el resultado de cada partición se presenta en la Tabla 6.

**Tabla 6**

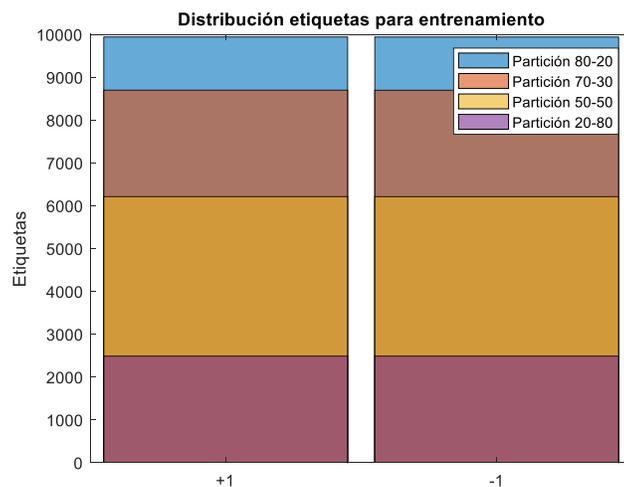
*Resultado de aplicar las 3 particiones tipo Hold Out*

Partición	Entrenamiento %	Train	Validación %	Validation	Total
1	80	19877	20	4969	24846
2	70	17393	30	7453	24846
3	50	12423	50	12423	24846
4	20	4970	80	19876	24846

La Figura 23 presenta el histograma de las etiquetas “+1” y “-1” del conjunto de datos para entrenamiento, donde se garantiza la partición equitativa en cada una de las particiones.

**Figura 23**

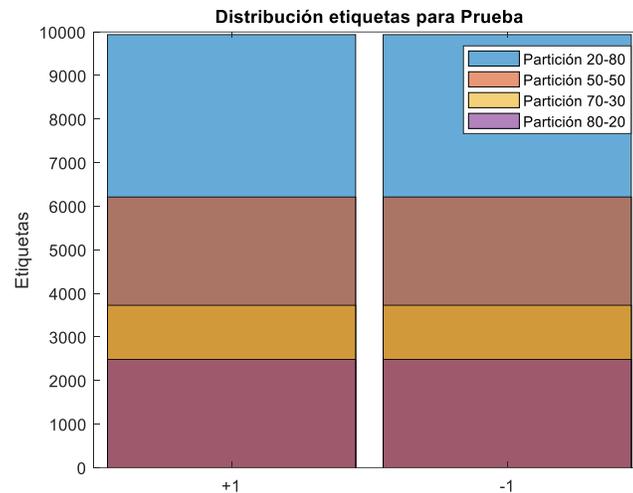
*Distribución equitativa de etiquetas para entrenamiento en partición 80–20, 70-30, 50-50 y 20-80*



La Figura 24 presenta el histograma de las etiquetas “+1” y “-1” del conjunto de datos para validación, donde se garantiza la partición equitativa en cada una de las particiones.

## Figura 24

*Distribución equitativa de etiquetas para validación en partición 80 - 20, 70 - 30, 50-50, 20-80*



### **Procesamiento de señales**

A partir de cada partición se genera las tres bases de datos requeridas para entrenar los modelos de ML.

#### **Base de datos en el dominio del tiempo**

La base de datos en el dominio del tiempo, no presenta modificación alguna, como presenta la Figura 25, aquí se encuentran ventanas de 15 segundos de señales que pertenece a un microsismo y a señales de ruido.

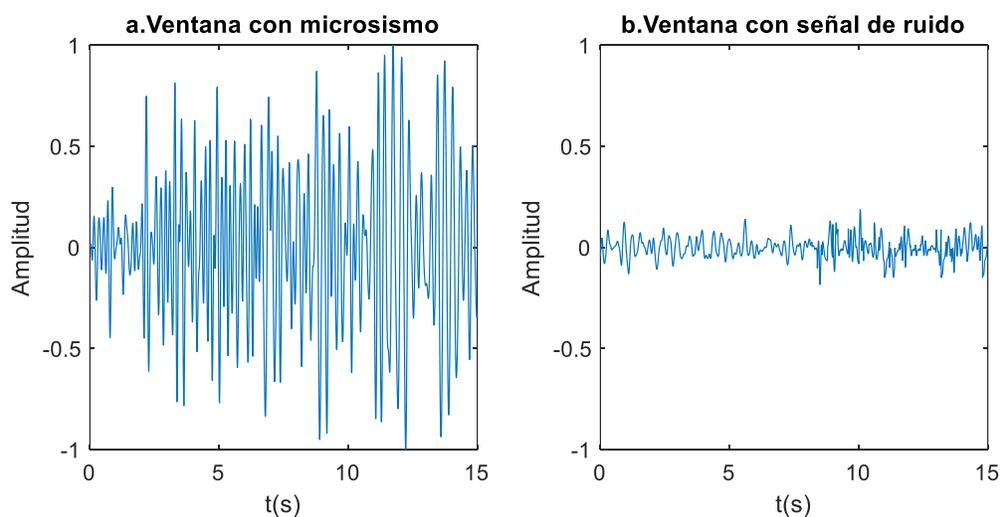
#### **Base de datos en el dominio de la frecuencia**

Se genera a partir de aplicar la técnica de la PSD Welch a cada ventana de 15 segundos. Por cada ventana se obtendrán 257 puntos debido a que se aplica una NFFT de 512. La Figura 26a, presenta el resultado de aplicar la PSD Welch a una ventana que contiene microsismo, la Figura 20b, presenta el resultado de aplicar la PSD Welch a una ventana que contiene ruido. Como se observa la diferencia en amplitud es significativa por lo cual no se

normaliza debido a que mediante prueba y error los mejores resultados se obtienen con los datos sin normalizar.

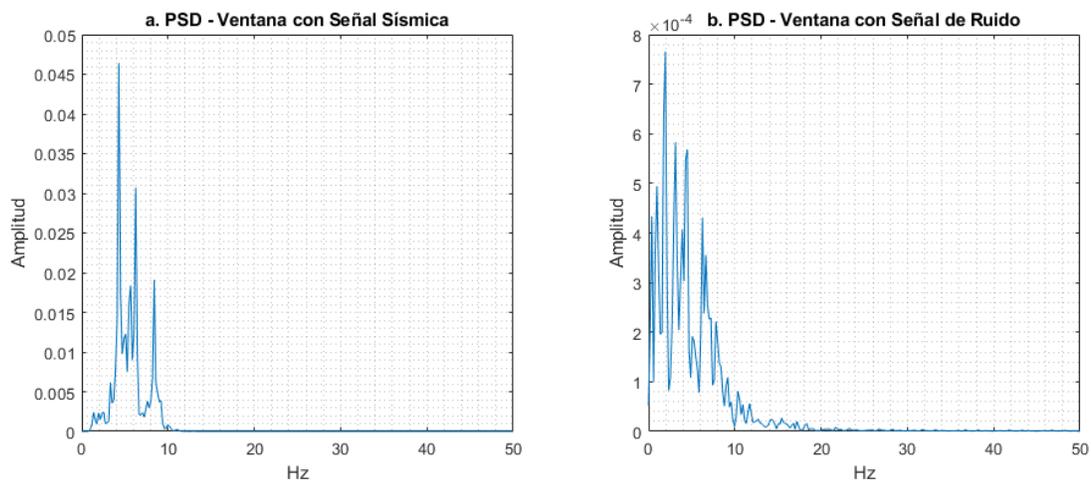
## Figura 25

*Base de datos en el dominio temporal*



## Figura 26

*PSD de ventanas con: (a) Microsismo (b) Señal de ruido*



### Base de datos mediante la extracción de las 84 características

Se calculan las 84 características de la base de datos en el dominio del tiempo, es decir por cada ventana de 15 segundos se obtiene las 84 características más relevantes entre tiempo, frecuencia y escala. La Tabla 7 presenta el resultado y la distribución de encontrar las 84 características. Las primeras 13 columnas corresponde a características en tiempo, entre la columna 14 y 54 se encuentran las características de frecuencia y en las últimas 30 columnas las características de escala.

**Tabla 7**

*Tabla de la extracción de las 84 Características de las primeras 5 ventanas de 15 segundos*

t_mean	t_std	t_var	t_entropy	t_kurtosis	t_multiscale	t_time2	t_rms	t_peak2	t_peak2
					Entropy	peak		peak	rms
-0.0203	0.2806	0.0787	191.3447	3.6875	2.0699	13.9900	0.2812	1.8318	3.5556
0.0054	0.2669	0.0712	177.0200	4.2550	2.0419	4.5900	0.2669	1.8101	3.7473
-0.0026	0.2166	0.0469	149.0047	4.1321	2.0630	12.3900	0.2166	1.8660	4.6177
-9.1188e-04	0.2344	0.0549	161.9117	4.6743	2.0998	14.9700	0.2343	1.9231	4.2676
0.0013	0.2079	0.0432	134.0521	6.0442	2.0572	1.1800	0.2078	1.9829	4.8124

*Nota:* Se presentan las 10 primeras características de las primeras 6 ventanas que corresponden a características de tiempo

Al extraer las 84 características, se observa que la característica 16, que corresponde a la entropía en frecuencia, muestra una amplitud considerablemente superior en comparación con las otras 83 características, esto se presenta en la Figura 27. Esto impide que el entrenamiento de los modelos sea el correcto, debido a que solo tomará esta característica para separar ruido de un microsismo, esta es la razón por la que se elige normalizar la base de datos por característica con el método de mínimos y máximos mediante la ecuación (16) que fue extraída de Amesquita, (2022).

$$z_M = \frac{c_M - \min(c_M)}{\max(c_M) - \min(c_M)}, \quad (16)$$

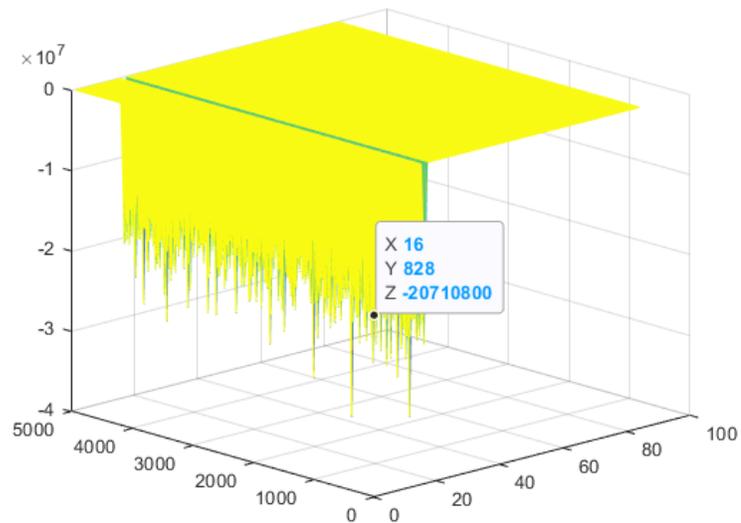
donde:

- $c_M$ .- representa las características que se desean normalizar
- $z_M$ .- representa las características normalizadas
- $\min(c_M)$ .- representa el valor mínimo del conjunto de características a normalizar
- $\max(c_M)$ .- representa el valor máximo del conjunto de características a normalizar

La Figura 28 presenta el resultado de normalizar con el método de mínimos y máximos, como se observa ya no existe una característica que sobresale respecto al resto.

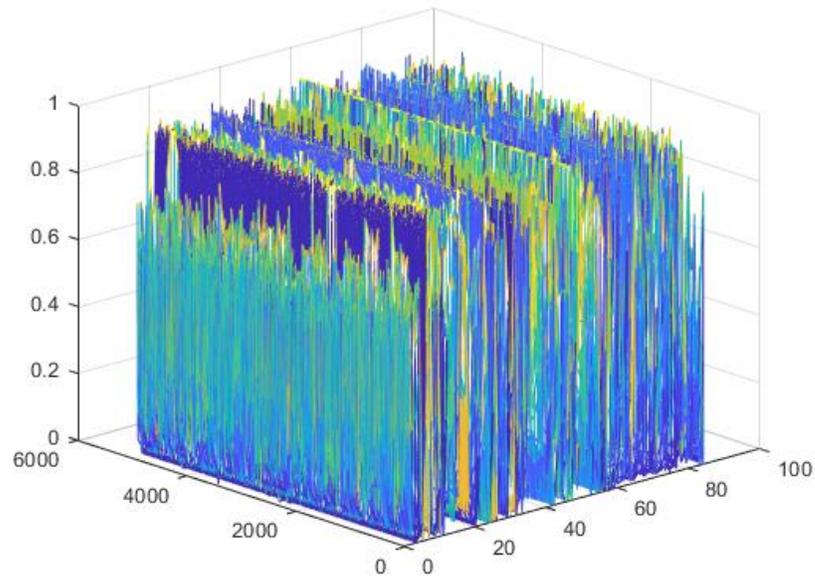
### **Figura 27**

*Característica 16 – Entropía en frecuencia*



**Figura 28**

*Normalización de las 84 características*

***Entrenamiento de modelos de ML***

Por cada modelo de ML se elige el valor del parámetro que minimiza el valor del BER o maximiza las métricas de desempeño A, P, R, S en cada una de las particiones con la base de datos en el dominio del tiempo, de la frecuencia y en la extracción de las 84 características.

**Partición 80 – 20*****Tiempo***

La Tabla 8 presenta los resultados obtenidos al entrenar los tres modelos de ML supervisado en partición de datos 80 – 20 en el dominio temporal.

**Tabla 8**

Resultados con la partición Hold Out 80 – 20 en los 3 modelos de ML en el dominio temporal

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 4 Prune = On SplitCriterion = gdi	99.51	99.51	99.51	99.51	0.0048
<b>SVM</b>	KernelFunction = linear BoxConstraint = 250	98.12	99.58	96.65	99.59	0.0187
<b>k-NN</b>	k = 2 Distance = cosine	98.04	99.95	96.13	99.95	0.0195

La configuración óptima se elige según la Figura 29 para DT, Figura 30 para SVM, Figura 31 para k-NN.

**Figura 29**

Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo

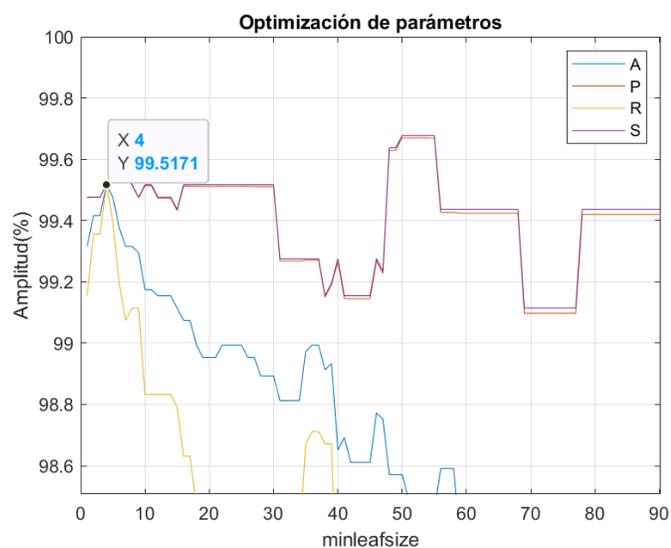


Figura 30

Curva de desempeño A, P, R, S vs BoxConstraint – SVM Tiempo

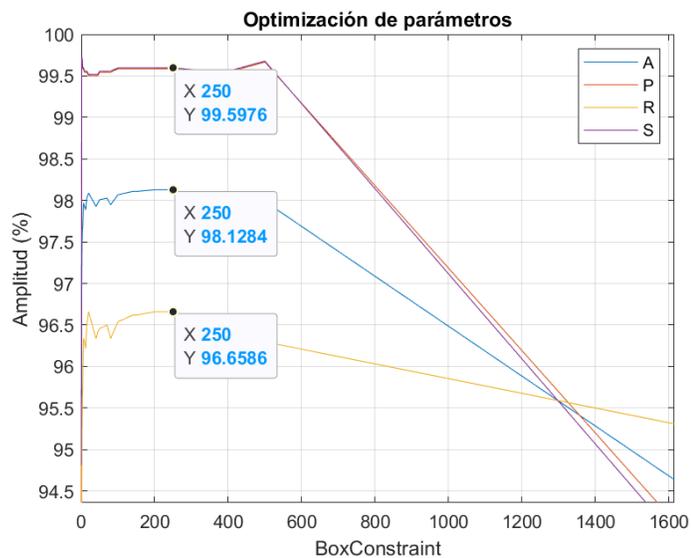
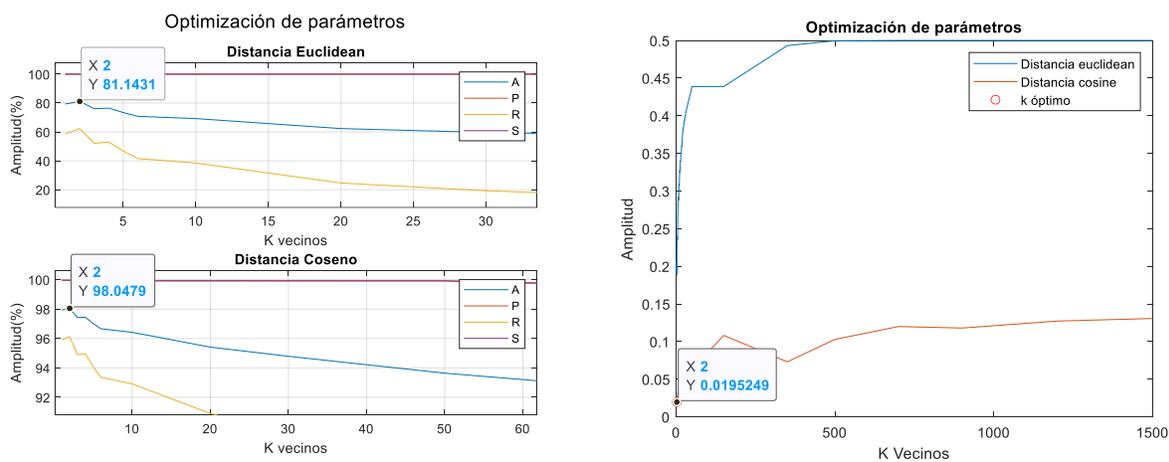


Figura 31

Curva de desempeño k-NN:(a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos



## Frecuencia

La Tabla 9 presenta los resultados obtenidos al entrenar los 3 modelos de ML con la partición de datos 80 – 20, en el dominio frecuencial.

**Tabla 9**

*Resultados con la partición Hold Out 80–20 en los 3 modelos de ML en el dominio frecuencial*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 4 Prune = On SplitCriterion = gdi	98.93	99.51	98.34	99.51	0.0106
<b>SVM</b>	KernelFunction = linear BoxConstraint = 2000	97.34	100	94.68	100	0.0260
<b>k-NN</b>	k = 2 Distance = cosine	98.26	98.34	98.18	98.35	0.0173

La configuración óptima se elige según la Figura 32 para DT, Figura 33 para SVM, Figura 34 para k-NN.

**Figura 32**

*Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia*

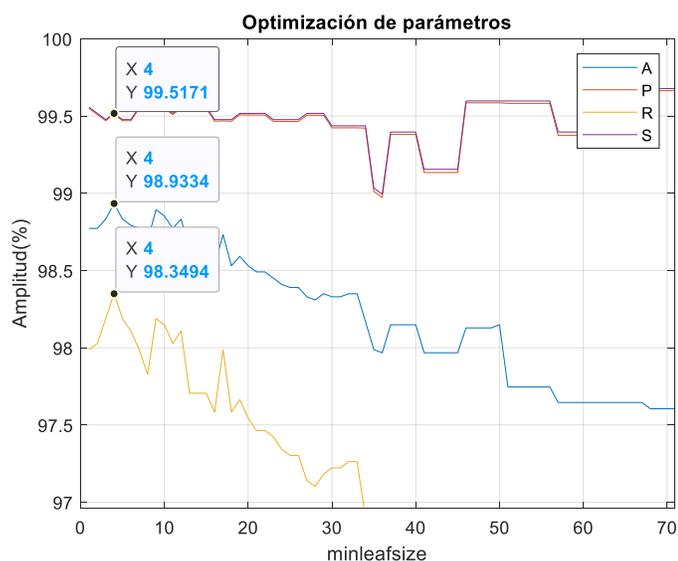


Figura 33

Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia

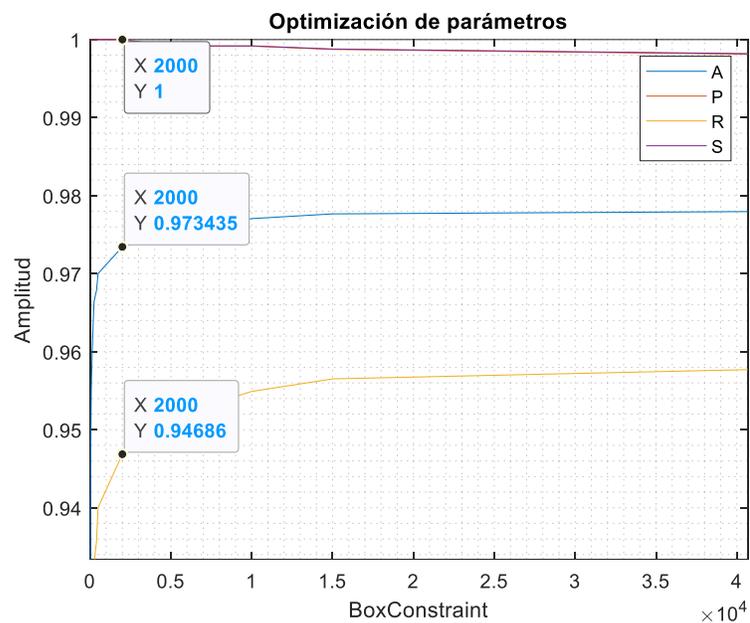
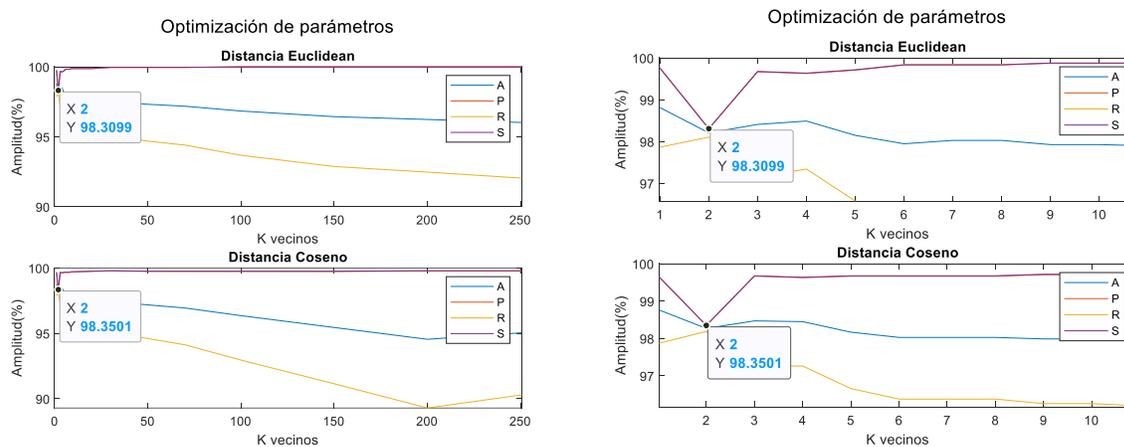


Figura 34

Curva de desempeño A,P,R,S vs k Vecinos – k-NN Frecuencia



### 84 características

La Tabla 10 presenta los resultados obtenidos al entrenar los 3 modelos de ML con la partición de datos 80 – 20, con la extracción de las 84 características.

**Tabla 10**

*Resultados con la partición Hold Out 80–20 en los 3 modelos de ML, con las 84 características*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 1 Prune = On SplitCriterion = gdi	100	100	100	100	0
<b>SVM</b>	KernelFunction = linear BoxConstraint = 15	99.95	99.95	99.95	99.95	0.0040
<b>k-NN</b>	k = 13 Distance = euclidean	99.89	99.87	99.91	99.87	0.0010

La configuración óptima se elige según la Figura 35 para DT, Figura 36 para SVM, Figura 37 para k-NN.

**Figura 35**

*Curva de desempeño A, P, R, S vs MinLeafSize – DT 84 características*

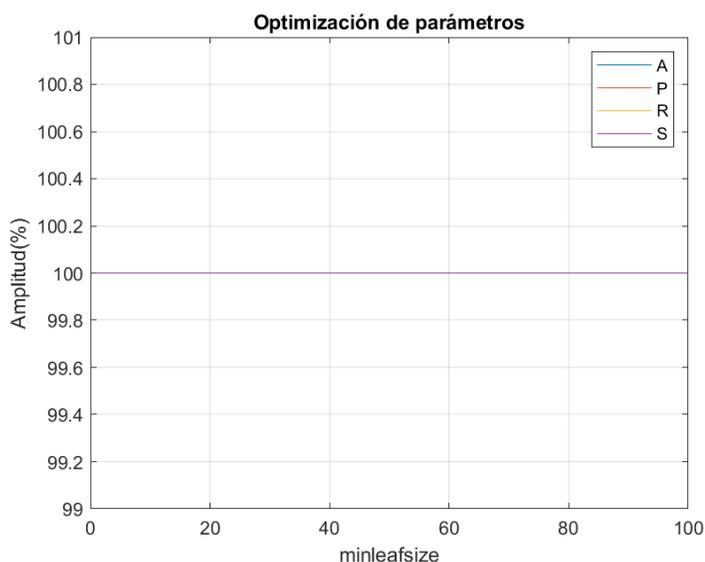


Figura 36

Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características

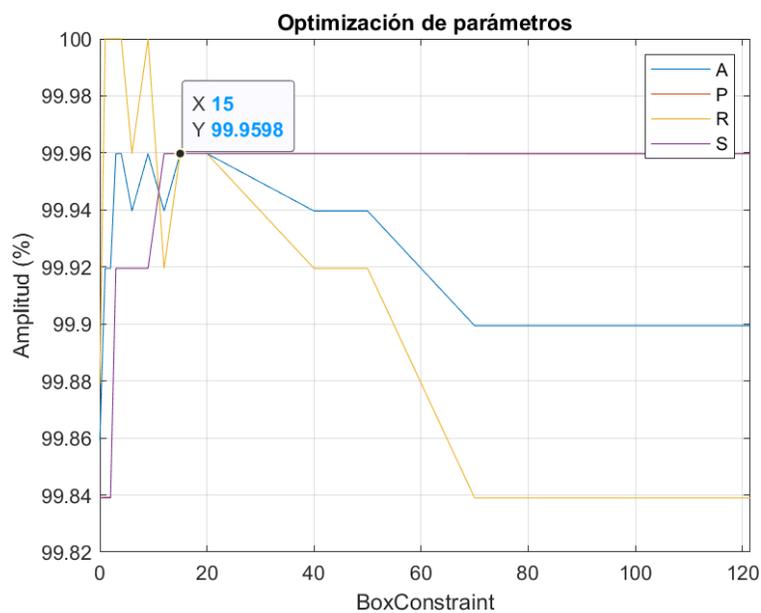
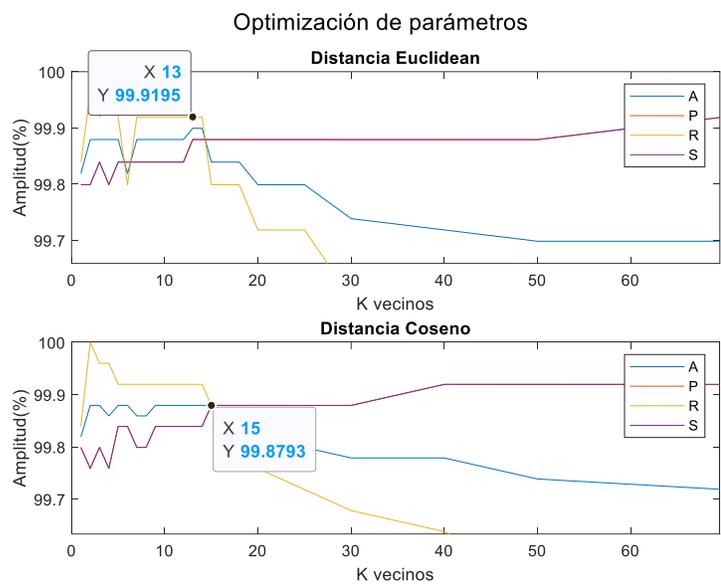


Figura 37

Curva de desempeño A, P, R, S vs k Vecinos – k-NN 84 características



## Partición 70 – 30

### Tiempo

La Tabla 11, presenta los resultados obtenidos al entrenar los 3 modelos de ML con la partición de datos 70 – 30 en el dominio temporal.

**Tabla 11**

*Resultados con la partición Hold Out 70 – 30 en los 3 modelos de ML en el dominio temporal*

Modelo	Configuración	A %	P %	R%	S %	BER
DT	MinleafSize = 9	99.43	99.54	99.32	99.54	0.0056
	Prune = On					
	SplitCriterion = gdi					
SVM	KernelFunction = linear BoxConstraint = 40	97.79	99.58	96.00	99.59	0.0220
k-NN	k = 700 Distance = cosine	89.21	91.57	86.36	92.05	0.1078

La configuración óptima se elige según la Figura 38 para DT, Figura 39 para SVM, Figura 40 para k-NN.

### Figura 38

*Curva de desempeño BER vs MinLeafSize – DT Tiempo*

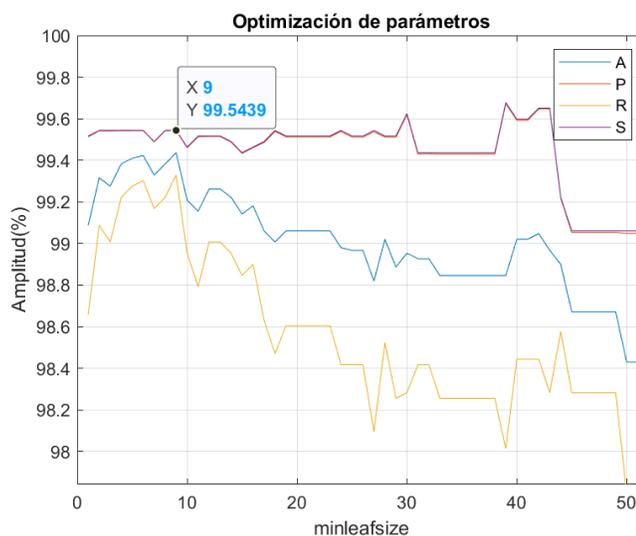


Figura 39

Curva de desempeño BER vs BoxConstraint – SVM Tiempo

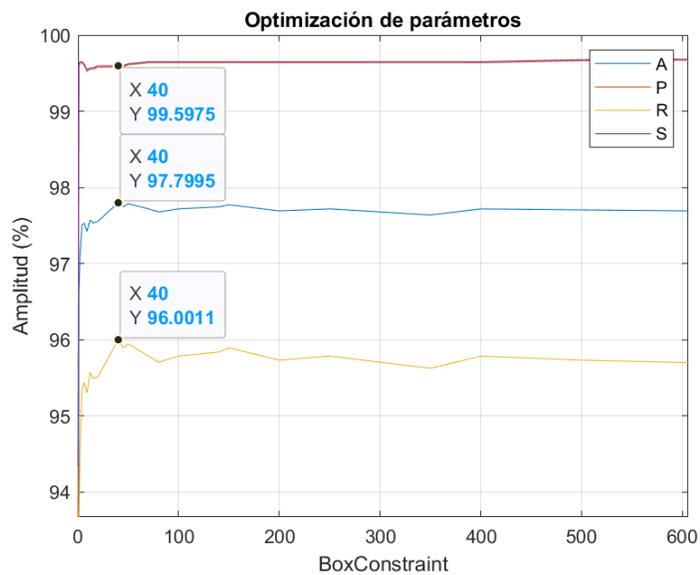
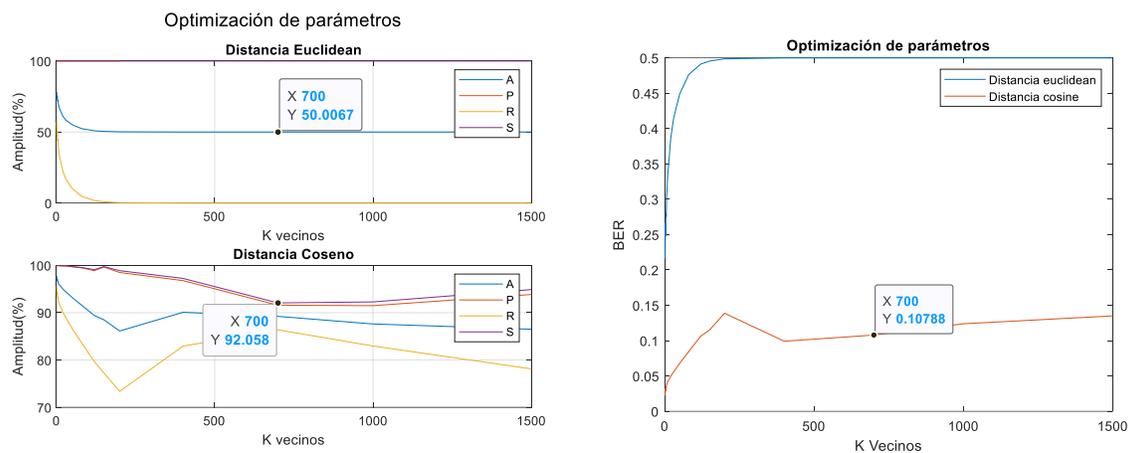


Figura 40

Curva de desempeño k-NN Tiempo (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos



## Frecuencia

La Tabla 12, presenta los resultados obtenidos al entrenar los 3 modelos de ML supervisado con la partición de datos 70 – 30, en el dominio frecuencial.

**Tabla 12**

*Resultados con la partición Hold Out 70 – 30 en los 3 modelos de ML en el dominio frecuencial*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 4 Prune = On SplitCriterion = gdi	98.69	99.48	97.90	99.49	0.0130
<b>SVM</b>	KernelFunction = linear BoxConstraint = 400	96.52	99.94	93.10	99.94	0.0347
<b>k-NN</b>	k = 3 Distance = cosine	98.25	99.64	96.85	99.65	0.0174

La configuración óptima se elige según la Figura 41 para DT, Figura 42 para SVM, Figura 43 para k-NN.

**Figura 41**

*Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia*

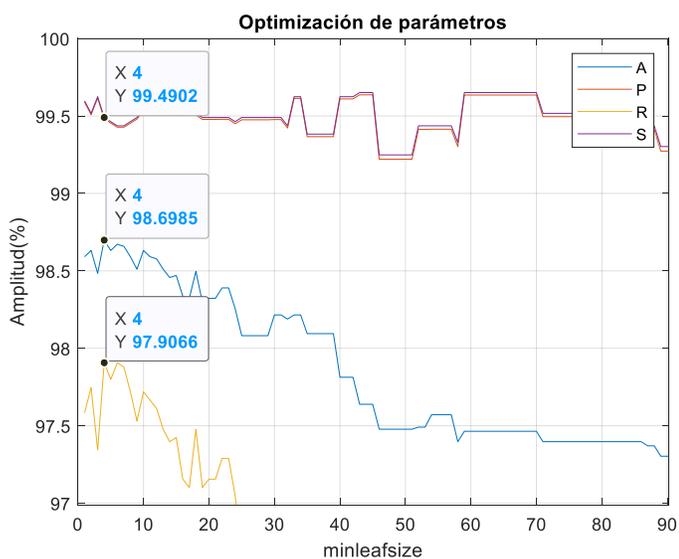


Figura 42

Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia

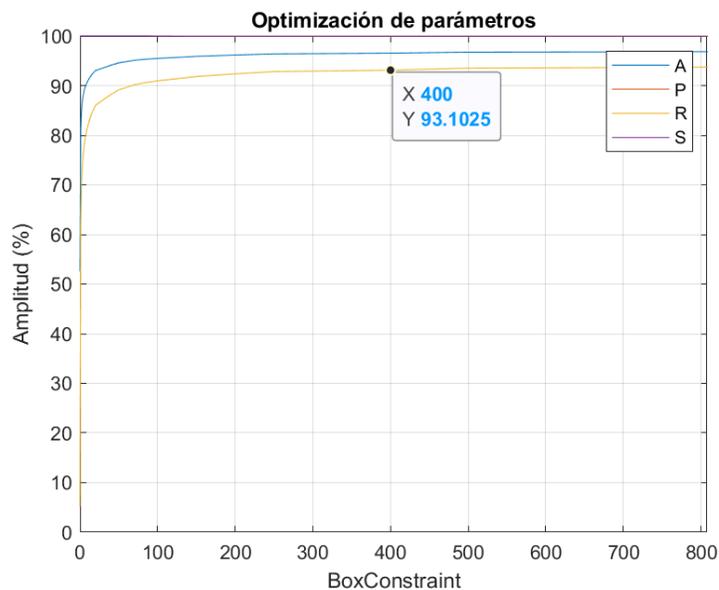
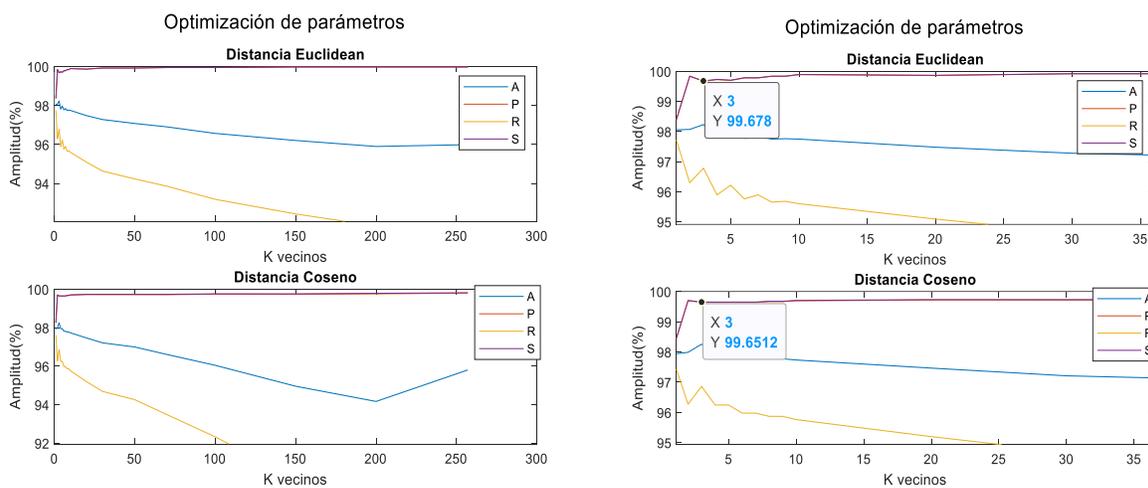


Figura 43

Curva de desempeño A, P, R, S vs k Vecinos – k-NN Frecuencia



### 84 características

La Tabla 13 presenta los resultados obtenidos al entrenar los 3 modelos de ML supervisado con la partición de datos 70 – 30, con la extracción de las 84 características.

**Tabla 13**

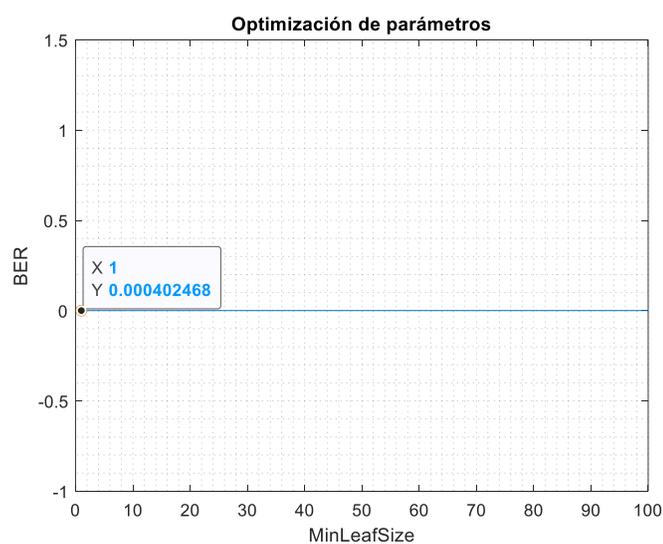
*Resultados con la partición Hold Out 70–30 en los 3 modelos de ML, con las 84 características*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 1 Prune = On SplitCriterion = gdi	99.95	99.91	100	99.91	0.0004
<b>SVM</b>	KernelFunction = linear BoxConstraint = 50	99.91	99.91	99.91	99.91	0.0080
<b>k-NN</b>	k = 3 Distance = euclidean	99.87	99.81	99.94	99.81	0.0012

La configuración óptima se elige según la Figura 44 para DT, Figura 45 para SVM, Figura 46 para k-NN.

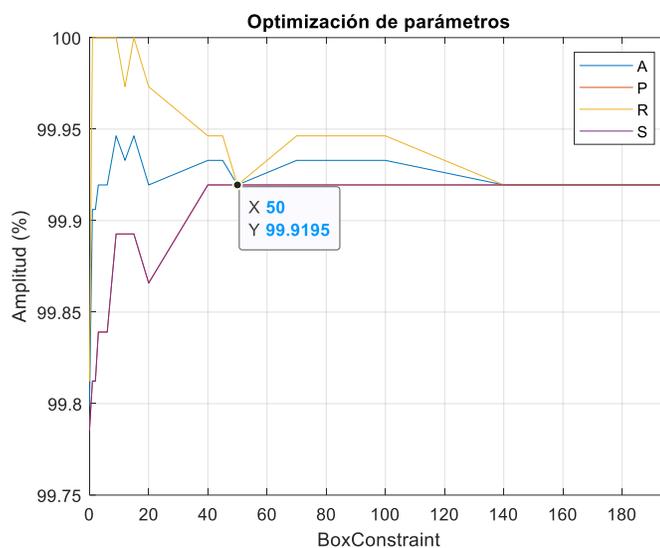
**Figura 44**

Curva de desempeño BER vs MinLeafSize – DT 84 características

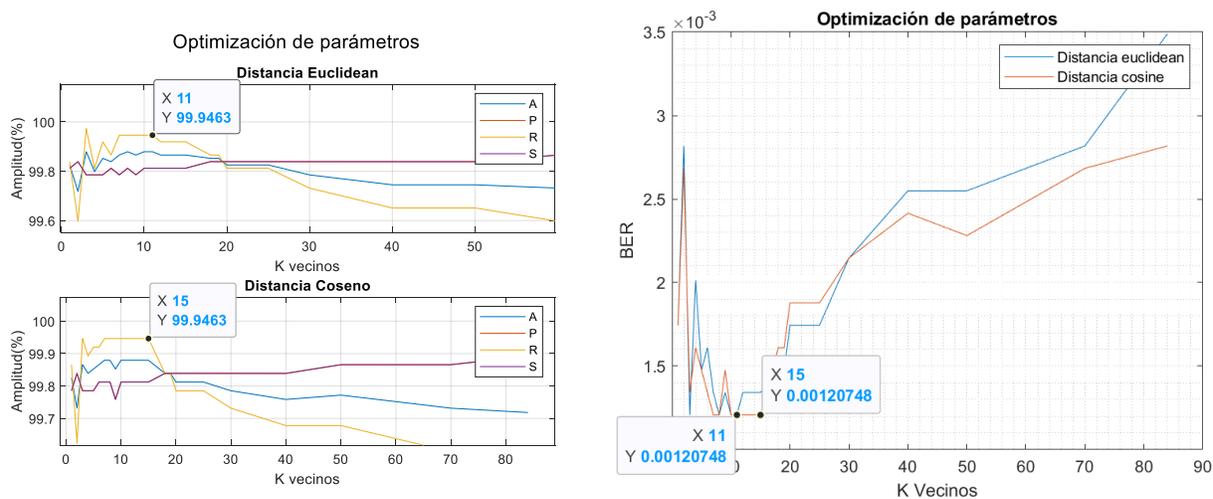


**Figura 45**

Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características

**Figura 46**

Curva de desempeño A, P, R, S vs  $k$  Vecinos –  $k$ -NN 84 características



*Nota:* Para cualquiera de los  $k$  vecinos mencionados en cualquier distancia los resultados de las métricas de desempeño son los mismos.

## Partición 50 – 50

### Tiempo

La Tabla 14 presenta los resultados obtenidos al entrenar los tres modelos de ML supervisado en partición de datos 50 – 50 en el dominio temporal.

**Tabla 14**

*Resultados con la partición Hold Out 50 – 50 en los 3 modelos de ML en el dominio temporal*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 6 Prune = On SplitCriterion = gdi	99.33	99.56	99.09	99.56	0.0066
<b>SVM</b>	KernelFunction = linear BoxConstraint = 20	97.27	99.64	94.89	99.66	0.0272
<b>k-NN</b>	k = 200 Distance = cosine	93.05	96.00	89.84	96.26	0.0694

La configuración óptima se elige según la Figura 47 para DT, Figura 48 para SVM, Figura 49 para k-NN.

### Figura 47

*Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo*

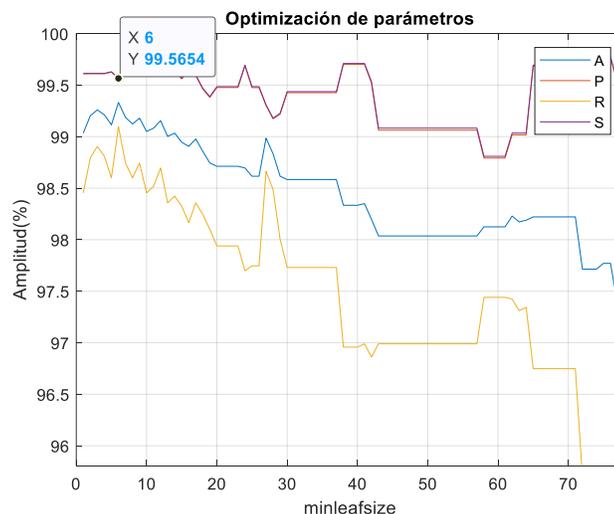


Figura 48

Curva de desempeño A, P, R, S vs BoxConstraint – SVM Tiempo

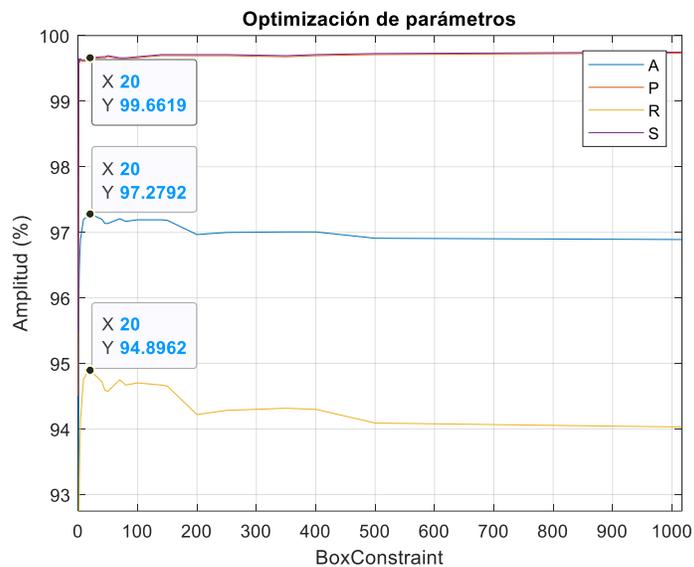
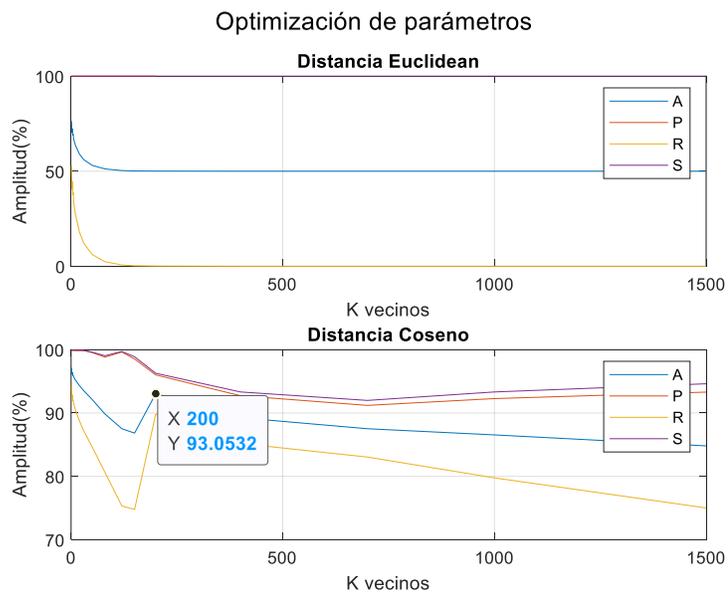


Figura 49

Curva de desempeño A, P, R, S vs k Vecinos – k-NN Tiempo



## Frecuencia

La Tabla 15 presenta los resultados obtenidos al entrenar los 3 modelos de ML con la partición de datos 50 – 50, en el dominio frecuencial.

**Tabla 15**

*Resultados con la partición Hold Out 50–50 en los 3 modelos de ML en el dominio frecuencial*

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 7 Prune = On SplitCriterion = gdi	98.29	99.63	96.94	99.64	0.0170
<b>SVM</b>	KernelFunction = linear BoxConstraint = 500	96.22	99.98	92.46	99.98	0.0377
<b>k-NN</b>	k = 1 Distance = cosine	98.17	99.61	96.71	99.62	0.0182

La configuración óptima se elige según la Figura 50 para DT, Figura 51 para SVM, Figura 52 para k-NN.

**Figura 50**

*Curva de desempeño A, P, R, S vs MinLeafSize – DT Frecuencia*

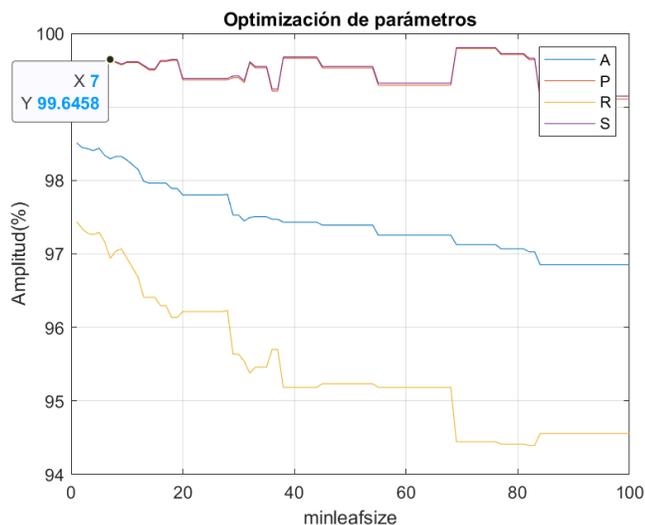


Figura 51

Curva de desempeño A, P, R, S vs BoxConstraint – SVM Frecuencia

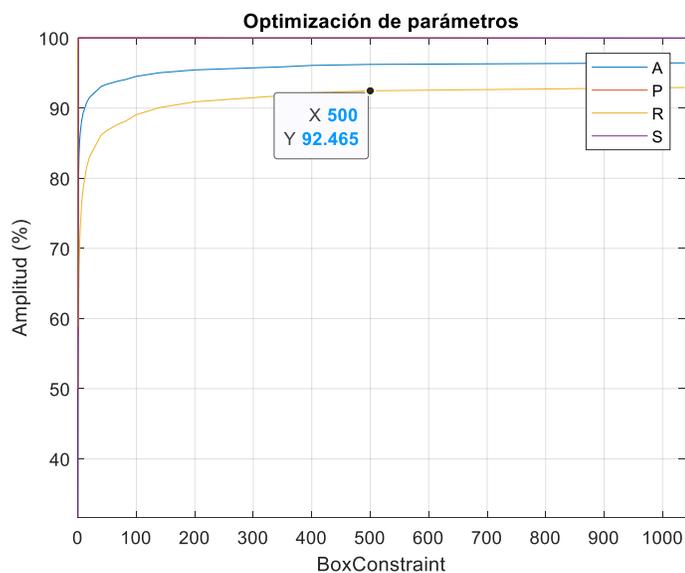
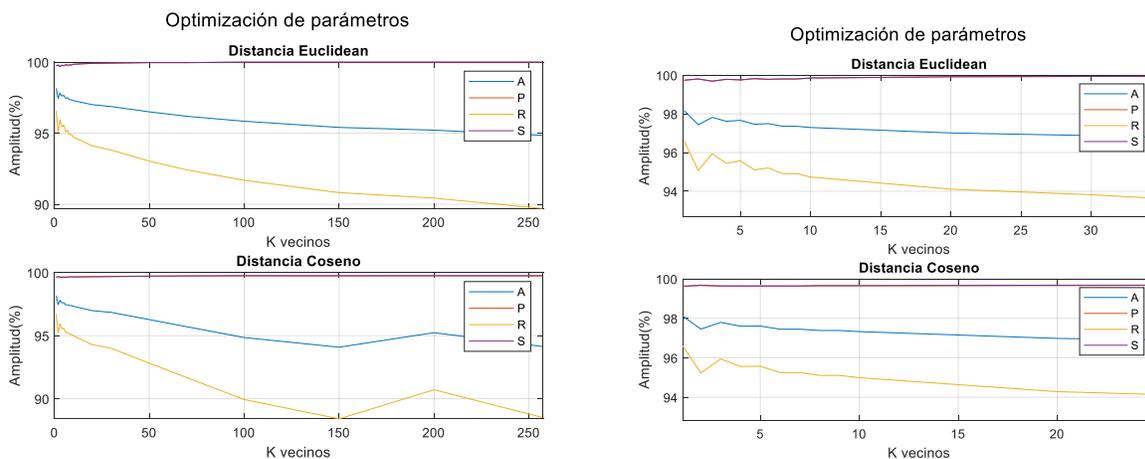


Figura 52

Curva de desempeño BER vs k Vecinos – k-NN Frecuencia



### 84 características

La Tabla 16 presenta los resultados obtenidos al entrenar los 3 modelos de ML con la partición de datos 50 – 50, con la extracción de las 84 características.

Tabla 16

Resultados con la partición Hold Out 50–50 en los 3 modelos de ML, con las 84 características

Modelo	Configuración	A %	P %	R%	S %	BER
DT	MinleafSize = 1 Prune = On SplitCriterion = gdi	99.09	100	98.18	100	0.0090
SVM	KernelFunction = linear BoxConstraint = 140	99.91	99.98	99.85	99.98	0.0008
k-NN	k = 3 Distance = euclidean	99.83	99.74	99.93	99.74	0.0016

La configuración óptima se elige según la Figura 53 para DT, Figura 54 para SVM, Figura 55 para k-NN.

Figura 53

Curva de desempeño BER vs MinLeafSize – DT 84 características

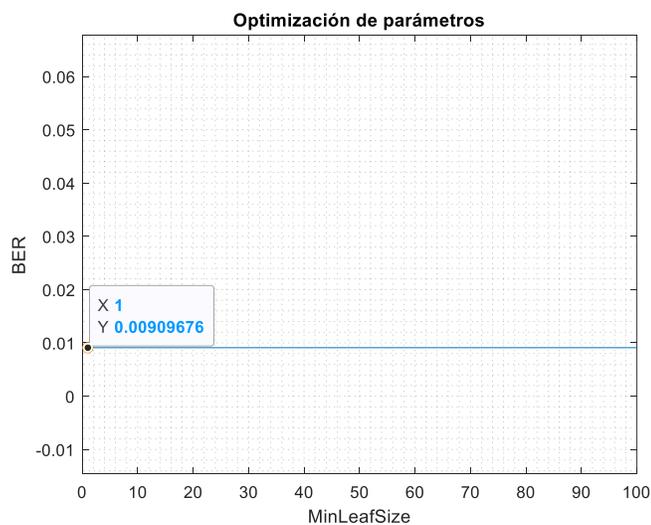


Figura 54

Curva de desempeño A, P, R, S vs BoxConstraint – SVM 84 características

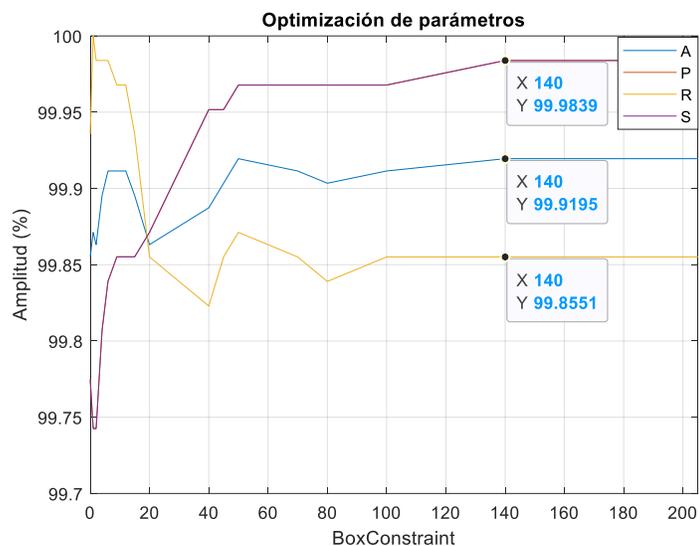
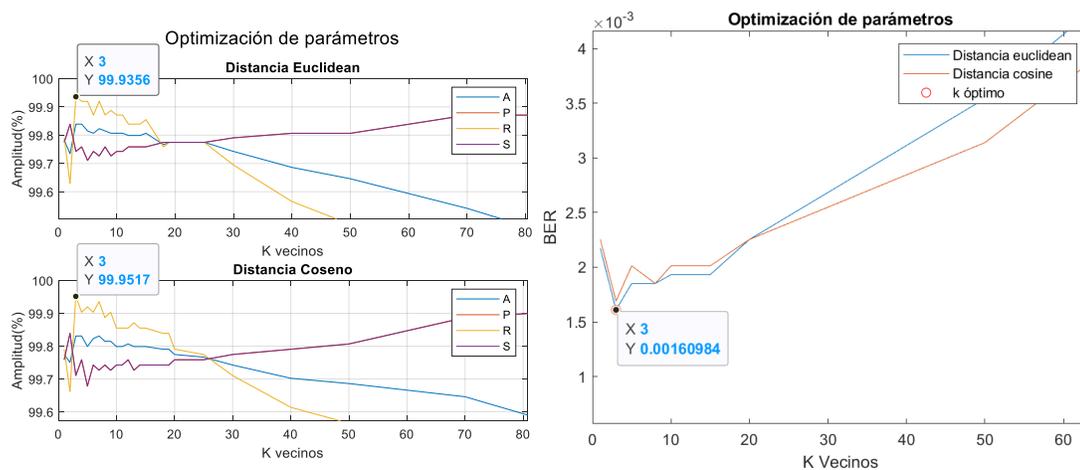


Figura 55

Curva de desempeño k-NN 84 características: (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos



## Partición 20 – 80

### Tiempo

La Tabla 17, presenta los resultados obtenidos al entrenar los 3 modelos de ML supervisado con la partición de datos 20 – 80 en el dominio temporal.

Tabla 17

Resultados con la partición Hold Out 20 – 80 en los 3 modelos de ML en el dominio temporal

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 3 Prune = On SplitCriterion = gdi	99.14	99.55	98.73	99.55	0.0085
<b>SVM</b>	KernelFunction = linear BoxConstraint = 1.5	99.57	99.16	100	99.15	0.0042
<b>k-NN</b>	k = 1 Distance = cosine	96.56	99.79	93.31	99.80	0.0343

La configuración óptima se elige según la Figura 56 para DT, Figura 57 para SVM, Figura 58 para k-NN.

Figura 56

Curva de desempeño A, P, R, S vs MinLeafSize – DT Tiempo

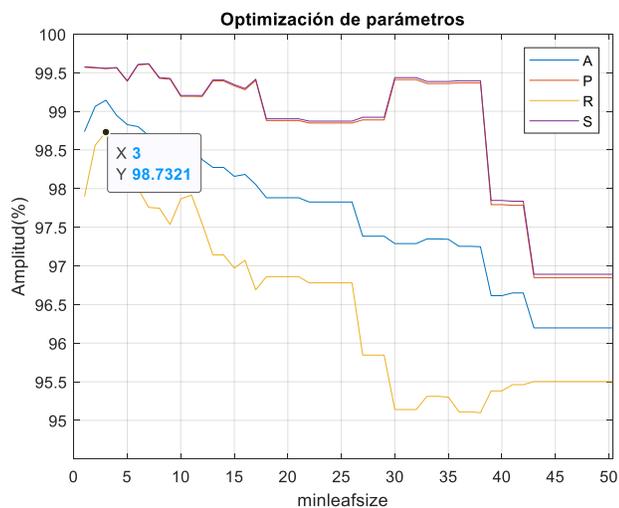


Figura 57

Curva de desempeño BER vs BoxConstraint – SVM Tiempo

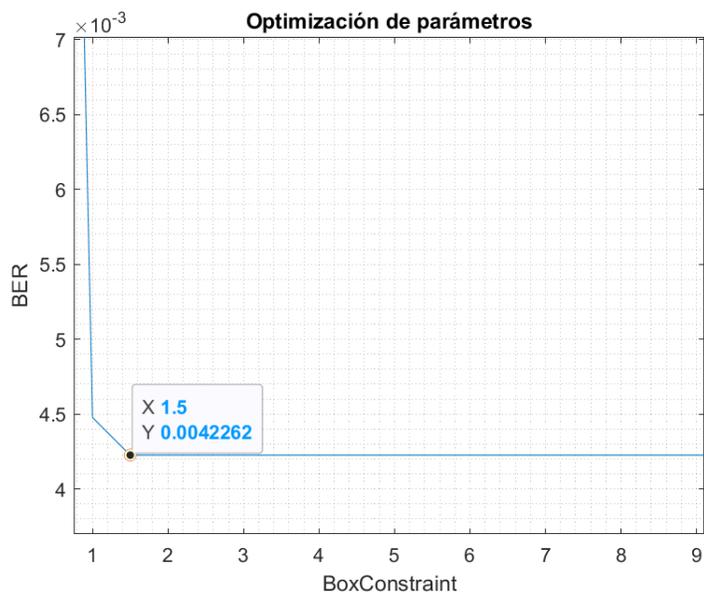
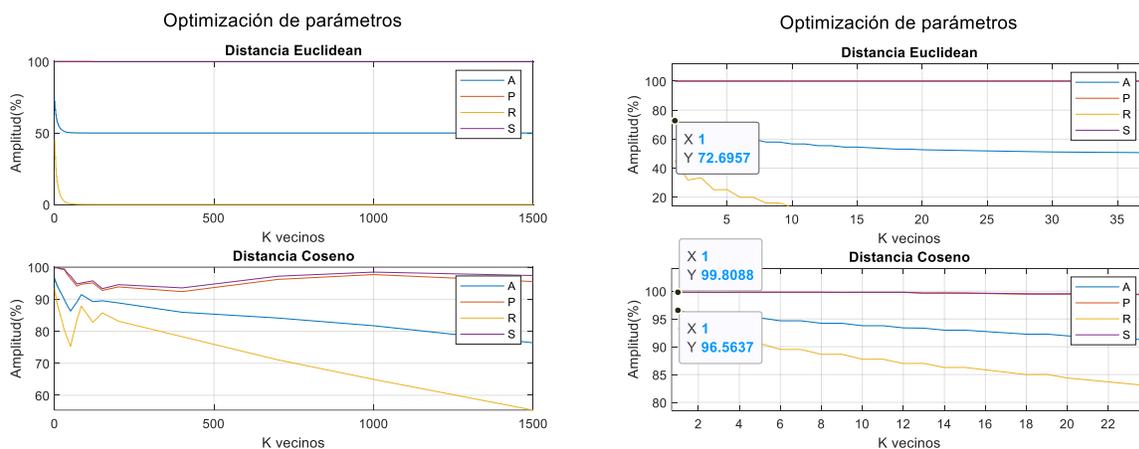


Figura 58

Curva de desempeño BER vs k Vecinos – k-NN Tiempo



### Frecuencia

La Tabla 18, presenta los resultados obtenidos al entrenar los 3 modelos de ML supervisado con la partición de datos 20 – 80, en el dominio frecuencial.

**Tabla 18**

Resultados con la partición Hold Out 20 – 80 en los 3 modelos de ML en el dominio frecuencial

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 1 Prune = On SplitCriterion = gdi	98.38	99.33	97.41	99.34	0.0171
<b>SVM</b>	KernelFunction = linear BoxConstraint = 400	95.41	99.97	90.85	99.97	0.0458
<b>k-NN</b>	k = 1 Distance = euclidean	97.87	99.52	96.20	99.53	0.02128

La configuración óptima se elige según la Figura 59 para DT, Figura 60 para SVM, Figura 61 para k-NN.

**Figura 59**

Curva de desempeño BER vs MinLeafSize – DT Frecuencia

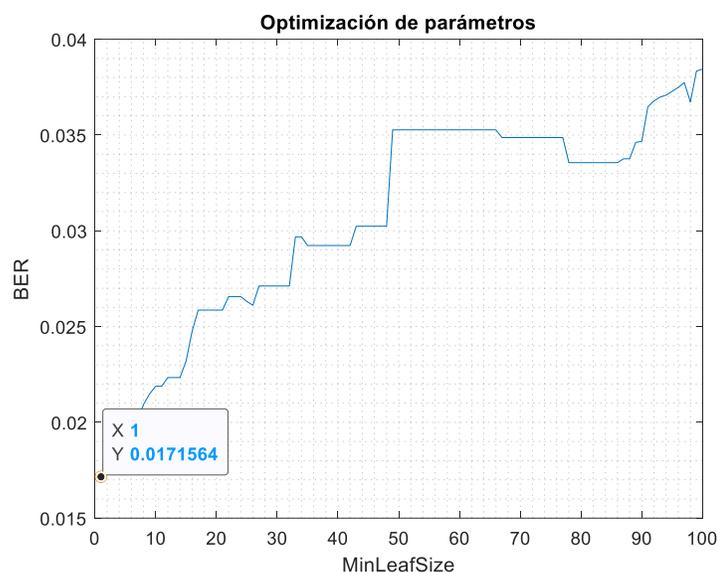


Figura 60

Curva de desempeño BER vs BoxConstraint – SVM Frecuencia

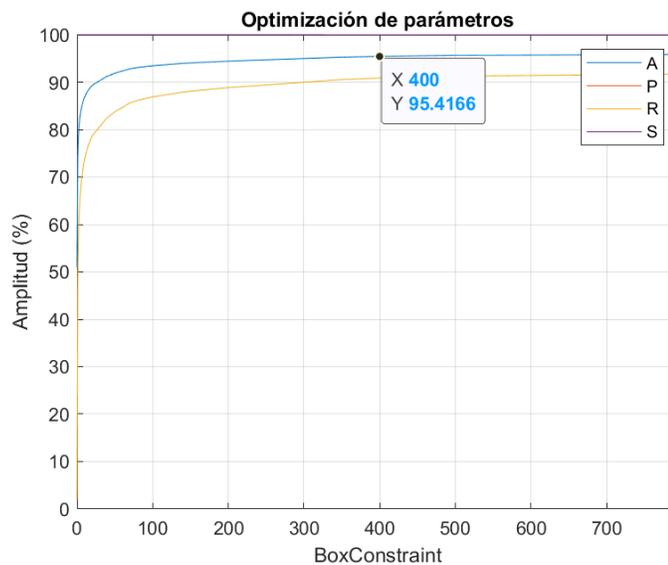
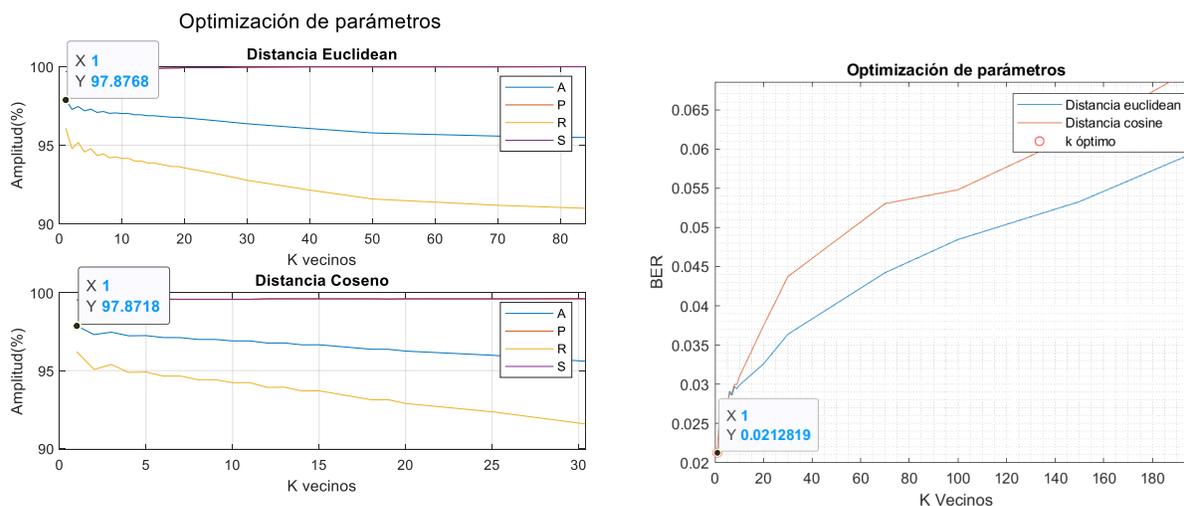


Figura 61

Curva de desempeño: (a) A, P, R, S vs k Vecinos (b) BER vs k Vecinos – k-NN Frecuencia



### 84 características

La Tabla 19, presenta los resultados obtenidos al entrenar los 3 modelos de ML supervisado con la partición de datos 20 – 80, con la extracción de las 84 características.

**Tabla 19**

Resultados con la partición Hold Out 20–80 en los 3 modelos de ML, con las 84 características

Modelo	Configuración	A %	P %	R%	S %	BER
<b>DT</b>	MinleafSize = 1 Prune = On SplitCriterion = gdi	99.97	99.95	100	99.95	0.0002
<b>SVM</b>	KernelFunction = linear BoxConstraint = 4	99.82	99.74	99.90	99.74	0.0017
<b>k-NN</b>	k = 5 Distance = cosine	99.75	99.59	99.90	99.59	0.0024

La configuración óptima se elige según la Figura 62 para DT, Figura 63 para SVM, Figura 64 para k-NN.

**Figura 62**

Curva de desempeño BER vs MinLeafSize – DT 84 características

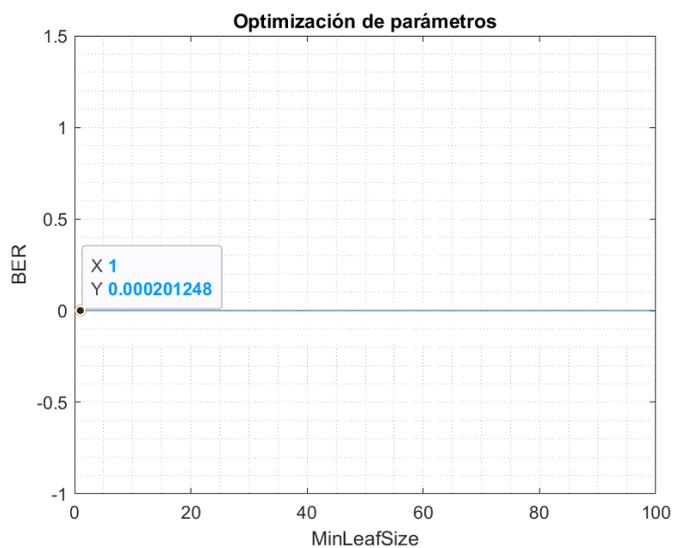


Figura 63

Curva de desempeño BER vs BoxConstraint – SVM 84 características

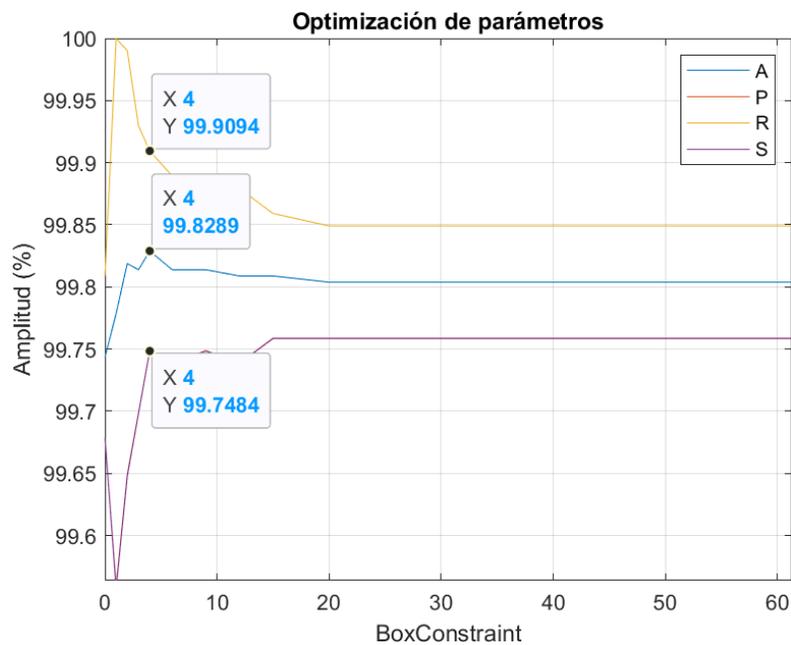
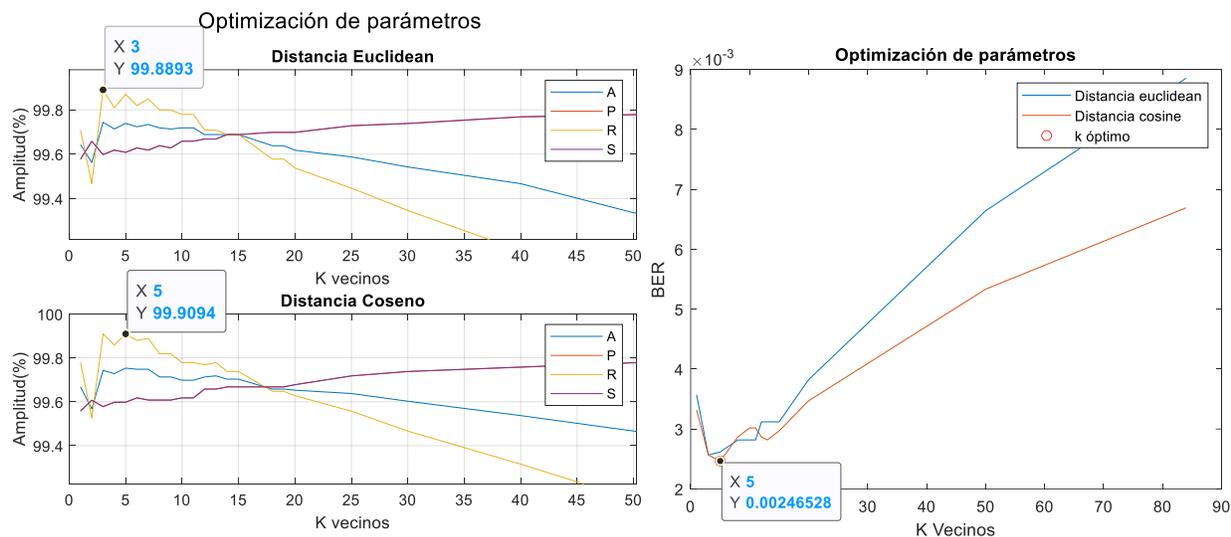


Figura 64

Curva de desempeño: (a) A, P, R, S vs BER (b) BER vs k Vecinos – k-NN 84 características

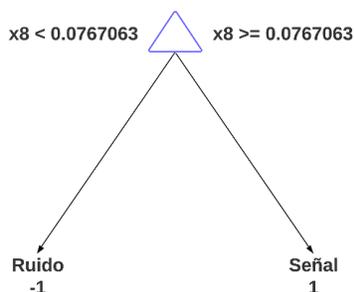


### Test a mejor modelo ML

El modelo DT en la partición 80 – 20 con la base de datos de las 84 características normalizadas con el método de mínimos y máximos obtuvo en validación en todas las métricas de rendimiento el 100% y un BER de 0. La característica que toma es la número 8 que corresponde al valor RMS en tiempo, el árbol generado se presenta en la Figura 65.

**Figura 65**

*Árbol de decisión del modelo DT con 84 características*



Se procede a probar el modelo con un conjunto de 158 registros de 20 minutos de duración cada uno. Estos registros abarcan un total de 806 microsismos aleatorios entre LP, VT, TC, TR. La elección de este número de microsismos se debe a que representa el 20% del total de microsismos de la base de datos.

En la fase de clasificación la Tabla 20 presenta las métricas de desempeño obtenidas en la clasificación de las etiquetas “+1” y “-1”.

**Tabla 20**

*Test de Clasificación del Modelo DT*

A %	P %	R %	S %	BER
97.41	92.73	93.97	98.23	0.038

En la fase de detección los resultados de las métricas de desempeño se muestran en la Tabla 21, donde a través del BER se indica que el modelo no es capaz de detectar 6 de cada 1000 microsismos.

**Tabla 21**

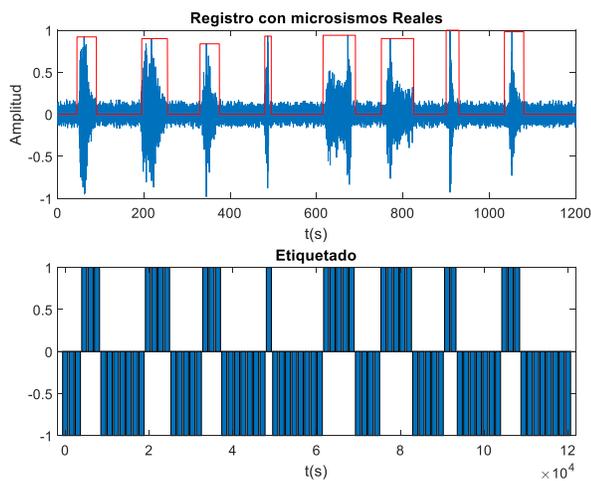
*Test en Detección del Modelo DT*

A %	P %	R %	S %	BER
99.66	99.26	99.26	99.37	0.006

La Figura 66 presenta un ejemplo de cómo el modelo realiza la detección, se puede verificar que detecta de forma correcta los 8 microsismos reales, que están dentro del registro de 20 minutos. El tiempo aproximado que se demora en detectar es de 5.17 segundos.

**Figura 66**

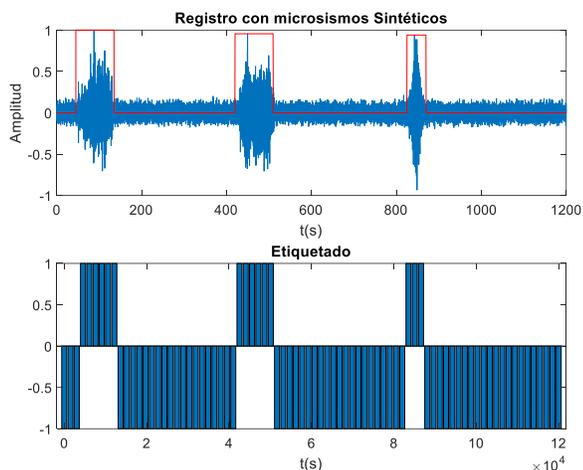
*Detección a registro con 8 microsismos reales*



La Figura 67, presenta la detección de un registro del volcán Llaima con señales sintéticas, se puede ver que este modelo logra detectar de forma correcta los 3 microsismos. El tiempo aproximado que se demora en detectar es de 4 segundos.

## Figura 67

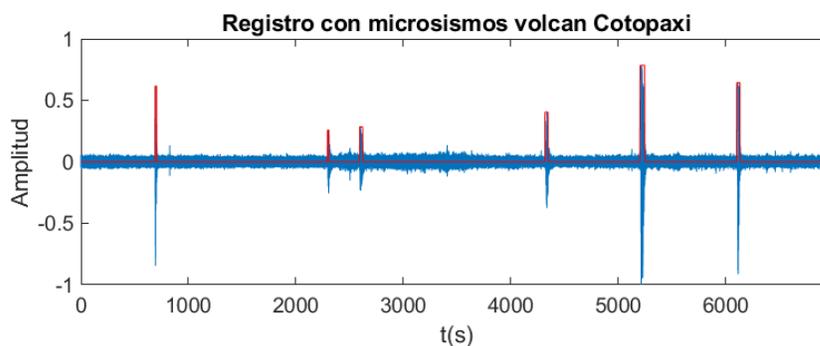
### *Detección a registro con 3 microsismos sintéticos*



Se prueba el modelo en registros distintos al volcán Llaima, la Figura 68 presenta el resultado de la detección de un registro del volcán Cotopaxi, se puede ver que este modelo logra detectar de forma correcta los microsismos. Sin embargo, al evaluar otro registro proveniente del mismo volcán el modelo presenta errores en la detección, esto se aprecia en la Figura 69. El tiempo aproximado que se demora en detectar el primer microsismo del volcán Cotopaxi es de 21 segundos debido a que es un registro mayor a 20 minutos y el tiempo que se demora en detectar el segundo microsismo es 4 segundos.

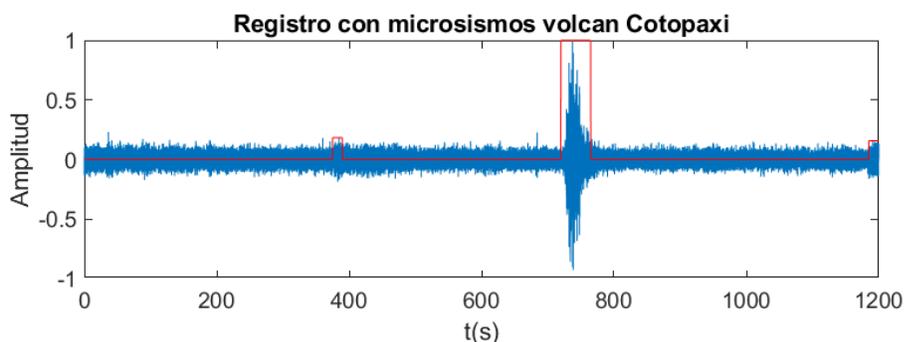
## Figura 68

### *Detección a registro del volcán Cotopaxi*



## Figura 69

*Detección a segundo registro del volcán Cotopaxi*



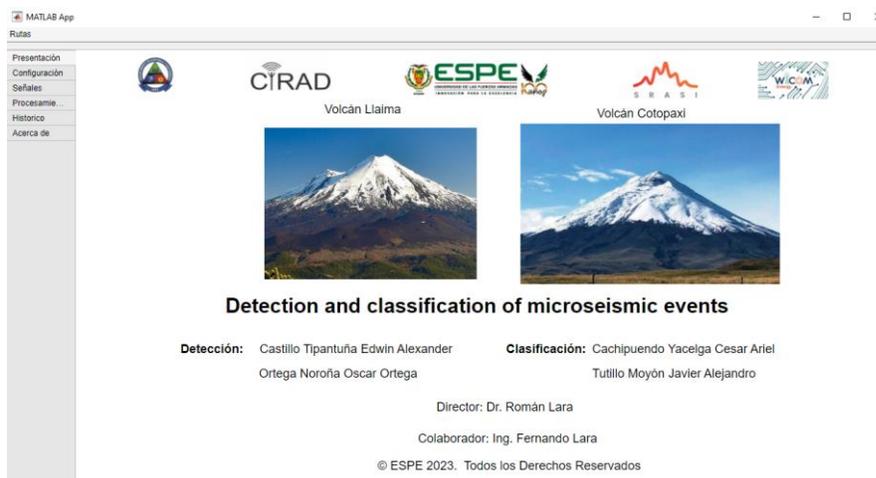
## Interfaz Gráfica

La interfaz gráfica es una modificación a una interfaz previa que dispone de un sistema de detección y clasificación del volcán Cotopaxi. A esta interfaz se agrega una nueva funcionalidad que es el detector basado en ML para el volcán Llaima, estas modificaciones se realizan mediante Design App que es propia de Matlab®.

La Figura 70 presenta la pestaña Presentación que es la pestaña que da la bienvenida al programa donde se muestra imágenes del volcán Llaima y Cotopaxi, nombre de los autores.

## Figura 70

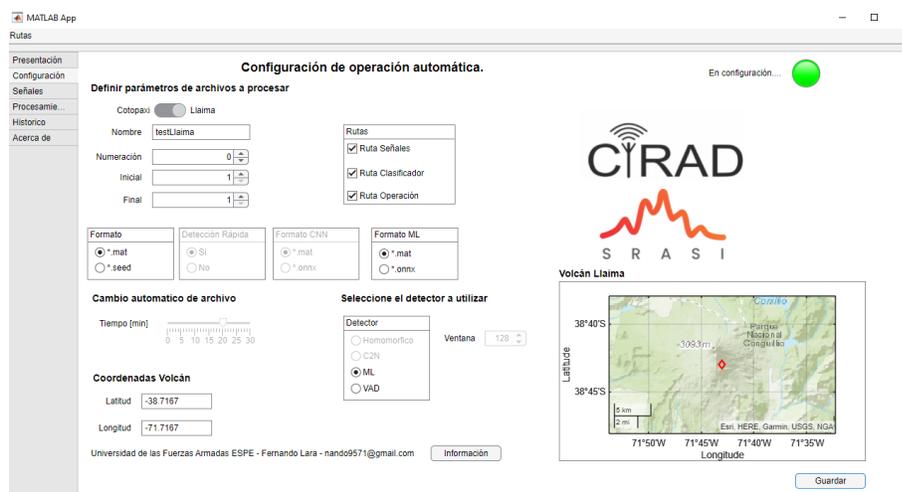
*Presentación del programa de detección y clasificación de microsismos*



La pestaña Configuración es donde seleccionaremos que microsismos deseamos detectar si del volcán Cotopaxi o Llaima, al elegir el volcán Llaima se muestra el mapa de ubicación del volcán y se activan las opciones de ML y VAD en este caso específico la opción de ML será la elegida. Posterior se configuran las rutas de donde provienen las señales, donde se encuentran los modelos de clasificación y en donde se guardarán las señales detectadas y clasificadas, esto mediante la pestaña Rutas ubicada en el lado izquierdo superior. Finalizada la configuración se presiona el botón guardar y se espera hasta que el led de estado se ponga de color verde, esto se presenta en la Figura 71.

**Figura 71**

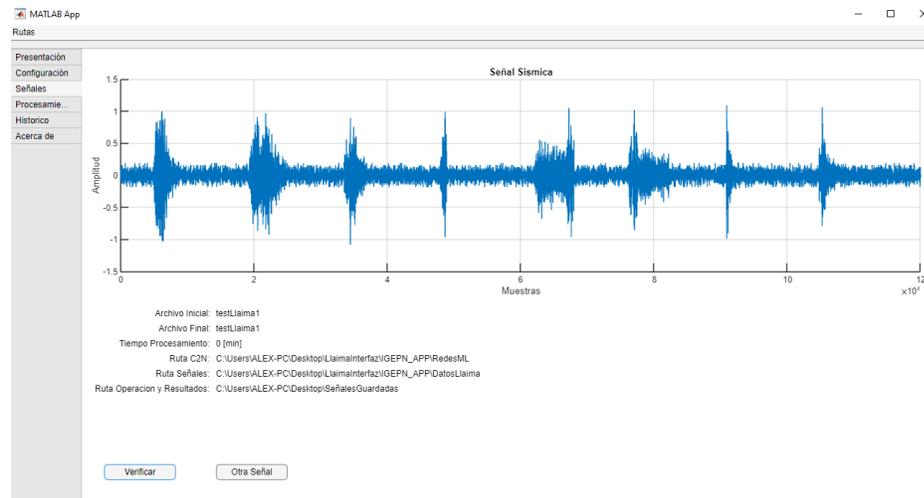
*Pestaña de configuración de parámetros*



La pestaña Señales grafica el registro que será utilizado para la detección y posterior clasificación esto se presenta en la Figura 72.

## Figura 72

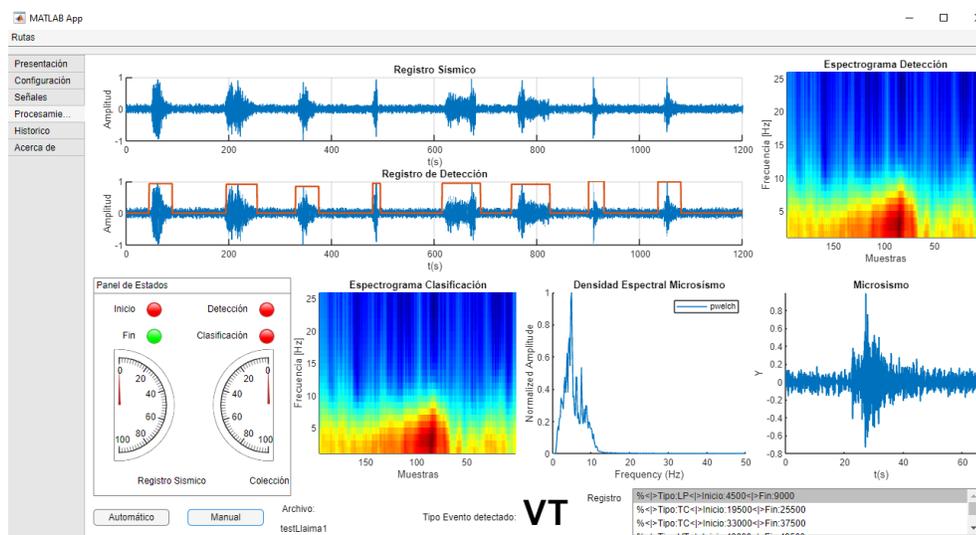
Gráfico del registro de 20 minutos



La Figura 73 presenta el contenido de la pestaña Procesamiento, aquí se genera el proceso de detección de los registros de 20 minutos del volcán Llaima, donde se visualiza el microsismo detectado, el espectrograma y la PSD. En la parte izquierda se puede ver el panel de estado.

## Figura 73

Proceso de detección de los registros de 20 minutos



Una vez finalizado el proceso se presiona la pestaña Histórico en ella se puede ver la tabla generada resultado del proceso de detección y clasificación el cual fue elaborado por otro colaborador. La tabla contiene los siguientes apartados: Network, Station, SampleRate, Component, Date, Type, Duration en segundos, StartPoint, EndPoint, además de un botón de almacenar que guardará esta base de datos en un archivo con extensión “.dat” o “.xlsx” Esto se presenta en la Figura 74.

Figura 74

Tabla con el resultado del proceso de detección y clasificación

MATLAB App

Rutas

Presentación

Configuración

Señales

Procesamie...

Historico

Acerca de

Datos de microsismos reconocidos

Network	Station	SampleRate	Component	Date	Type	Duration(s)	StartPoint	EndPoint
Chile	LAV	100	SHZ	13-Aug-2023	LP	65	4500	9000
Chile	LAV	100	SHZ	13-Aug-2023	TC	80	19500	25500
Chile	LAV	100	SHZ	13-Aug-2023	TC	65	33000	37500
Chile	LAV	100	SHZ	13-Aug-2023	VT	35	48000	49500
Chile	LAV	100	SHZ	13-Aug-2023	LP	95	61500	69000
Chile	LAV	100	SHZ	13-Aug-2023	TR	95	75000	82500
Chile	LAV	100	SHZ	13-Aug-2023	LP	50	90000	93000
Chile	LAV	100	SHZ	13-Aug-2023	VT	65	103500	108000

Estado  Almacenar

## Capítulo IV

### Conclusiones

Se ha logrado crear un algoritmo basado en la teoría de ML tradicional para detectar eventos sísmicos del volcán Llaima. Este modelo no solo se ha ajustado a las exigencias de rendimiento establecidas por instituciones encargadas de la vigilancia volcánica, sino que ha superado dichos requisitos al alcanzar un BER inferior a 0.01.

Los detectores tienen la capacidad de identificar registro con extensiones de 20 minutos que contienen tanto ruido como uno o varios microsismos. Dada la ausencia de tales registros en la base de datos disponible, se procede a generar una señal de ruido de 20 minutos. Esta señal de ruido se combina con las señales existentes, lo que permite abordar eficazmente esta limitación.

Los criterios utilizados por los diferentes autores citados en el presente documento fueron relevantes a la hora de tomar decisiones en las distintas etapas que compone al detector como en el filtrado, que permitió eliminar la frecuencia de 0.2 Hz; en la segmentación de señales, donde reafirmamos que la ventana de 15 segundos funciona de manera óptima en la detección de microsismos; en los métodos de procesamiento de señales, como la extracción de las 84 características que son las que generan los mejores resultados al entrenar los modelos de ML.

La normalización de la base de datos de las 84 características fue una etapa importante previo al entrenamiento de los modelos de ML, esto permite evitar que el modelo se concentre solo en características específicas debido a sus valores en amplitud significativamente altos o bajos, lo que impide que exista un sobre entrenamiento y los resultados obtenidos sean generalizables a nuevos datos.

En la fase de prueba el modelo DT en la partición 80 – 20 con la base de datos de las 84 características normalizadas con el método de mínimos y máximos obtuvo en A, P, R, S y BER un 97.41%, 92.73%, 93.97%, 98.23% y 0.038 respectivamente en la clasificación de etiquetas "+1" y "-1", sin embargo, en la etapa de detección el modelo alcanza en A, P, R, S y BER un 99.66%, 99.26%, 99.26%, 99.37% y 0.006 respectivamente, donde a través del BER se indica que el modelo no es capaz de detectar 6 de cada 1000 microsismos.

El modelo toma la característica 8 (Valor RMS en tiempo) como única para separar microsismo de ruido, esto se corrobora mediante el cálculo del promedio del valor RMS de las etiquetas +1 y -1, donde se obtiene para la etiqueta "+1" un promedio de 0.3459 y para la etiqueta "-1" un promedio de 0.0351, por lo tanto, esta diferencia genera el umbral de decisión que utiliza el modelo DT.

Se eligió 3dB como valor mínimo de SNR para etiquetar los registros segmentados, debido a que este valor permitía una distinción adecuada entre ventanas de ruido y microsismo. Para valores inferiores a 3dB ventanas de ruido eran etiquetadas como señal.

Es posible detectar hasta 8 microsismos como máximo en un registro de 20 minutos, al exceder esta cantidad, el algoritmo detecta 2 microsismos como si fuera 1, hasta el punto en que todo el registro se identifica como una única señal debido a la proximidad entre los microsismos.

Debido a que el modelo se ha entrenado únicamente con microsismos del volcán Llaima, esto lo convierte en un modelo exclusivo para este volcán, por este motivo se evidencian deficiencias al intentar detectar microsismos provenientes de otros volcanes, como el Cotopaxi. Aunque en ocasiones logra detectar correctamente algunos registros que se componen de 1 o varios microsismos, en otros casos confunde el ruido como si fuera una señal válida.

## Trabajos Futuros

El presente trabajo de titulación ha logrado detectar microsismos del volcán Llaima en registros de 20 minutos mediante el uso de técnicas de ML tradicional, sin embargo, nuestro grupo de investigación plantea investigar como la señal de ruido afecta a nuestro detector cuando la SNR es distinto a 3dB.

Se propone modificar parámetros adicionales a cada uno de los modelos propuestos como "Maxnumsplit" en DT, nuevas funciones del kernel en SVM o nuevas distancias en k-NN como minkowski, Manhattan entre otros, con el objetivo de investigar la posibilidad de lograr mejoras en los resultados obtenidos en la generalización del modelo a nuevos microsismos provenientes de otros volcanes.

El presente trabajo de titulación emplea las 84 características para entrenar los modelos de ML, por lo cual se propone realizar un estudio que incorpore técnicas de reducción de características, como la Eliminación Recursiva de Características (RFE, por sus siglas en inglés *Recursive Feature Elimination*), con el fin de mejorar el proceso de entrenamiento de los modelos de ML.

## Referencias

- Altamirano, B. (2021). *Sistema de reconocimiento de microterremotos en tiempo real del volcán Cotopaxi aplicando aprendizaje supervisado*.  
<http://repositorio.espe.edu.ec/jspui/handle/21000/23743>
- Amesquita, D. (2022, April 2). *Statologos*. <https://statologos.com/normalizar-datos-entre-0-y-1/>
- Awan, A. (2022, November). *A Complete Guide to Data Augmentation | DataCamp*.  
<https://www.datacamp.com/tutorial/complete-guide-data-augmentation>
- Bhatti, S. M., Khan, M. S., Wuth, J., Huenupan, F., Curilem, M., Franco, L., & Yoma, N. B. (2016). Automatic detection of volcano-seismic events by modeling state and event duration in hidden Markov models. *Journal of Volcanology and Geothermal Research*, 324, 134–143. <https://doi.org/10.1016/J.JVOLGEORES.2016.05.015>
- Bordino, J. (2022, June 30). *Vulcanología: qué es y objetivos - Resumen, historia, objetivos e instrumentos*. <https://www.ecologiaverde.com/vulcanologia-que-es-y-objetivos-3646.html>
- Canário, J. P., de Mello, R. F., Curilem, M., Huenupan, F., & Rios, R. A. (2020). Llaima volcano dataset: In-depth comparison of deep artificial neural network architectures on seismic events classification. *Data in Brief*, 30, 105627. <https://doi.org/10.1016/J.DIB.2020.105627>
- Curilem, M., de Mello, R. F., Huenupan, F., San Martín, C., Franco, L., Hernández, E., & Rios, R. A. (2018). Discriminating seismic events of the Llaima volcano (Chile) based on spectrogram cross-correlations. *Journal of Volcanology and Geothermal Research*, 367, 63–78. <https://doi.org/10.1016/J.JVOLGEORES.2018.10.023>
- DataFlair. (n.d.). *Support Vector Machines Tutorial - Learn to implement SVM in Python*. Retrieved August 12, 2023, from <https://data-flair.training/blogs/svm-support-vector-machine-tutorial/>

- Díaz, R. (n.d.). *Algoritmo KNN - cómo funciona y ejemplos en Python*. Retrieved August 12, 2023, from <https://www.themachinelearners.com/algoritmo-knn/>
- Editorial Etecé. (2023, February 7). *Cordillera de Los Andes*. <https://concepto.de/cordillera-de-los-andes/>
- Edmundo, G., & Cox, P. (2003, September 10). *Andeshandbook*. <https://www.andeshandbook.org/montanismo/cerro/146/Llaima>
- Equipo editorial Etecé. (2023, March 6). *Cinturón de fuego del Pacífico*. <https://concepto.de/cinturon-de-fuego-del-pacifico/>
- Hvilshoj, F. (2022, November 11). *Balanced and Imbalanced Datasets in Machine Learning*. <https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning/>
- Ibañez, J. M., & Carmona, E. (2000, August). *SISMICIDAD VOLCÁNICA*. [https://www.researchgate.net/publication/242584653\\_SISMICIDAD\\_VOLCANICA](https://www.researchgate.net/publication/242584653_SISMICIDAD_VOLCANICA)
- IBM. (n.d.). *¿Qué es un árbol de decisión? | IBM*. Retrieved August 12, 2023, from <https://www.ibm.com/es-es/topics/decision-trees>
- Kennett, B. L. N. (2001). *The Seismic Wavefield: Volume 1, Introduction and Theoretical Development* (Vol. 1). Cambridge University Press. [https://books.google.com.ec/books?hl=es&lr=&id=u02QAjN6EEEC&oi=fnd&pg=PR9&ots=4OYcA3veES&sig=zuzCYxk5uWb\\_8zKg1y64rdJ3ksw&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.ec/books?hl=es&lr=&id=u02QAjN6EEEC&oi=fnd&pg=PR9&ots=4OYcA3veES&sig=zuzCYxk5uWb_8zKg1y64rdJ3ksw&redir_esc=y#v=onepage&q&f=false)
- Lara, F., León, R., Lara-Cueva, R., Tinoco-S., A. F., & Ruiz, M. (2022). Detection of volcanic microearthquakes based on homomorphic deconvolution and STA/LTA. *Journal of Volcanology and Geothermal Research*, 421, 107439. <https://doi.org/10.1016/J.JVOLGEORES.2021.107439>

- Lara-Cueva, R. A., Benítez, D. S., Carrera, E. V., Ruiz, M., & Rojo-Álvarez, J. L. (2016). Feature selection of seismic waveforms for long period event detection at Cotopaxi Volcano. *Journal of Volcanology and Geothermal Research*, 316, 34–49.  
<https://doi.org/10.1016/J.JVOLGEORES.2016.02.022>
- Lara-Cueva, R., Larco, J. C., Benítez, D. S., Pérez, N., Grijalva, F., & Ruiz, M. (2020). On finding possible frequencies for recognizing microearthquakes at Cotopaxi volcano: A machine learning based approach. *Journal of Volcanology and Geothermal Research*, 407, 107092. <https://doi.org/10.1016/J.JVOLGEORES.2020.107092>
- Mendoza, J. (2020, November 23). *¿Qué es un modelo SVM?*  
<https://estadisticamente.com/que-es-un-modelo-svm/>
- Mohajon, J. (2020, May 28). *Confusion Matrix for Your Multi-Class Machine Learning Model*.  
<https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>
- Moreno, I. (2021, January 17). *Construyendo árboles de decisión*. StatDeveloper.  
<https://www.statdeveloper.com/construyendo-arboles-de-decision/>
- OPS. (n.d.). *Erupciones volcánicas*. Retrieved July 9, 2023, from  
<https://www.paho.org/es/temas/erupciones-volcanicas>
- Paez, C. (2022, August 27). *¿Por qué es importante normalizar los conjuntos de datos?* DEV.  
<https://dev.to/christianpaez/por-que-es-importante-normalizar-los-conjuntos-de-datos-1de6>
- Pérez, N., Benítez, D., Grijalva, F., Lara-Cueva, R., Ruiz, M., & Aguilar, J. (2020). ESeismic: Towards an Ecuadorian volcano seismic repository. *Journal of Volcanology and Geothermal Research*, 396, 106855. <https://doi.org/10.1016/J.JVOLGEORES.2020.106855>

Rosero Jácome, K. G. (2020). *Adaptación e implementación de algoritmos voice activity detection para la detección de eventos microsísmicos del volcán Cotopaxi.*

<http://repositorio.espe.edu.ec/jspui/handle/21000/21640>

SERNAGEOMIN. (n.d.). *Volcán Llaima - Red Nacional de Vigilancia Volcánica.* Retrieved July 7, 2023, from <https://rnvv.sernageomin.cl/volcan-llaima/>

SERNAGEOMIN. (2023, May 28). *Sernageomin asiste al cierre del FONDEF sobre clasificación de eventos sísmicos - SERNAGEOMIN.* <https://www.sernageomin.cl/sernageomin-asiste-al-cierre-del-fondef-sobre-clasificacion-de-eventos-sismicos/>

Sotaquirá, M. (2022, September 9). *La Matriz de Confusión | Codificando Bits.*

<https://www.codificandobits.com/blog/matriz-de-confusion/>