



Creación de bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning.

Estrella Caicedo, Myriam Cristina

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Telecomunicaciones

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

Ing. Lara Cueva, Román Alcides PhD.

18 de septiembre del 2023



Plagiarism report

tesis_CGAN_final_para_copy.pdf

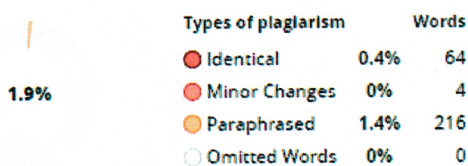
Scan details

Scan time:
August 25th, 2023 at 20:43 UTC

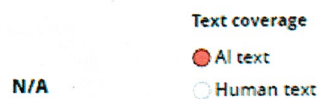
Total Pages:
61

Total Words:
15133

Plagiarism Detection



AI Content Detection



About this report
help.copyleaks.com



Firmado electrónicamente por:
ROMAN ALCIDES LARA
CUEVA

Ing. Lara Cueva, Román Alcides PhD.

C.C.:1713988218



Departamento de Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Telecomunicaciones

Certificación

Certifico que el trabajo de integración curricular: **“Creación de bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning”** fue realizado por la señorita **Estrella Caicedo, Myriam Cristina**, 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, 18 de septiembre de 2023



Ing. Lara Cueva, Román Alcides PhD.

C.C.:1713988218



Departamento de Departamento de Eléctrica, Electrónica y Telecomunicaciones
Carrera de Telecomunicaciones

Responsabilidad de Autoría

Yo, **Estrella Caicedo, Myriam Cristina**, con cédula de ciudadanía 1723522395, declaro que el contenido, ideas y criterios del trabajo de integración curricular **Título: “Creación de bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning”** 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, 18 de septiembre de 2023

Estrella Caicedo, Myriam Cristina

C.C.: 1723522395



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Telecomunicaciones

Autorización de Publicación

Yo **Estrella Caicedo, Myriam Cristina**, con cédula de ciudadanía 1723522395 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Título:** “**Creación de bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 18 de septiembre de 2023

Estrella Caicedo, Myriam Cristina

C.C.:1723522395

Dedicatoria

A Juan Estrella Prado. Gran parte de lo que soy, de lo que valoro y aprecio es gracias a ti. Esta alma de acetato que llevo es un pedazo de tu ser.

Myriam Cristina Estrella Caicedo

Agradecimientos

A todas las personas que me han aportado sus conocimientos a lo largo de mi vida. Desde que mi padre, Juan Enrique Estrella Prado, quien me enseñó a amarrarme los zapatos; cuando mi madre, Ayda del Carmen Caicedo Calderón, intentaba que escribiera mi nombre sin siquiera haber ingresado al jardín; aquellos días en que mi hermana, Marianita Leonor Estrella Caicedo en la escuela, estudiaba en voz alta y sin querer yo memorizaba sin siquiera entender del tema. A Santiago Gangotena que nos acogió en su hogar, sembró en mi cabeza un mundo de fantasía dentro de su casa de ensueño. Allí no solo mi locura se desarrolló, sino también los buenos modales y aprendí el cómo manejarse en una mesa al servir los alimentos y ver más allá del límite al cual estaba condenada. A ese lugar donde crecí y que recuerdo con tanta nostalgia, Cumbayá; a cada salida con mi familia donde veíamos las “viudas” todos los fines de año. Agradezco a todos los que formaron parte de una niñez entrañable, bajo el sol amable del amanecer sobre la Ciudadela México, donde mi abuelita Mariana nos esperaba siempre con una sonrisa y en su casa albergaba a mi hermana de otra madre, Angélica.

A ti Guambrita, hiciste que leer formara parte de lo que soy. Los pocos libros que han pasado por mis manos y hasta la actualidad, son mi tesoro. Alicia en el País de las Maravillas me marcó para siempre. Cuando encuentro en cualquier lectura, un párrafo escrito de manera elegante, recuerdo a quien me enseñó a volar por medio de las palabras, a valorar el correcto uso de los signos ortográficos, a releer lo que escribo hasta lograr coherencia, hasta que cada oración encaje de manera perfecta y permita hacer un viaje entre palabras.

Sé que sonará extraño agradecer a personas que ni siquiera conoceré, pero que gracias a mi padre las descubrí. Sí, la enseñanza no es solo académica, sino que hay conocimientos que quedan para toda la vida. El sonido melancólico, tenue y lleno de ruido que se puede escuchar en una radio AM “...van cantando por la Sierra...”, es algo invaluable. José Candelario “Tres Patines”, en la tremenda corte, se escuchaba en Radio Nacional Espejo, cada tarde que mi padre

sintonizaba a la hora del almuerzo. Que decir de Don Ramón, las rizas más estridentes que he tenido fueron al ver “El chavo del 8”. La incursión por la radio me llevó a oír programas que formaron parte de mis tardes al hacer deberes o viajar en bus, Animales Suelos y en especial los jueves que eran retro.

Cuando ingresé a la universidad, me sorprendió lo capaces que son las personas, y que una buena enseñanza hace la diferencia. Agradezco a la Escuela Politécnica Nacional, por haberme hecho descubrir el amor que le tengo al idioma Inglés, con sus excelentes profesores del CEC. Más allá del instituto de idiomas, se encontraban de los mejores ingenieros que he conocido, Ingeniero Gonzalo Calvache e Ingeniero Carlos León Calvache, que, con una simple materia, Geometría, hicieron que me dé cuenta de que yo si podía entender y, por lo tanto, seguir una ingeniería.

En la Universidad de las Fuerzas Armadas ESPE, me he formado para ser profesional, durante estos 6 años, con altos y bajos. Mi más sincero agradecimiento al Ingeniero Vinicio Carrera, quien me mostró la única materia que ha despertado mi pasión, Introducción al Aprendizaje Automático, su forma de enseñar es de las mejores, no tiene precio alguien que comparte sus conocimientos con tanta pasión, que uno como estudiante ama la materia. Finalmente, al Ingeniero Román Lara, PhD, por la oportunidad de desarrollar un tema que tiene que ver con el aprendizaje automático.

Viejito, los días sin usted son bastante difíciles, su apoyo incondicional, su compañía, su sonrisa y sus caras, su locura compartida, la manito y la manota, su ser tan puro, hicieron más llevadero todos estos años. Usted siempre será mi compañerito, el viejito pocho que espero ver antes de partir de este mundo, al último que desearía escuchar, los últimos deditos que quisiera tener entre mis manos.

Escribir refiriéndome solo a personas suena a suprimir todo lo que es intangible. Tantas bandas musicales que me han acompañado en esta odisea llamada vida y no solo en mi paso por la universidad. Por el 2013 la magia de oír Extremoduro o Marea me permitía darle un sentido a la vida. Por mis oídos y mi corazón han pasado un sinnúmero de bandas. Invasada por las sombras se oía el sonido ahogado mientras sentía por mis dedos la neblina del *Black Metal*. En medio de mi soledad y el caos, la confusión y la desesperación me rodeaban los sonidos de *Cigarettes After Sex* y entre cada una de sus hebras se escondía *Tango with Lions*. Pero hay algo que no ha cambiado y que no cambiará, la palabra sublime no es suficiente para una banda que susurra en el alma y responde con un grito en el corazón, me ha acompañado desde el colegio y mientras escribo este párrafo retruenan sus notas musicales. ¡*In Flames!*

“(Don't Fear) The Reaper”, para todos esos seres que se han ido, amigos de cuatro patas que han sabido hacer la diferencia con su paso apresurado por este mundo, *Navy* y *Pretty*, incluso los que no han estado en mi casa y que fueron los mejores amigos de mi Viejito, Coquito.

Be Good.

Myriam Cristina Estrella Caicedo

Índice de Contenidos

Resumen	17
Abstract.....	18
Capítulo I Introducción	19
Antecedentes	20
Justificación e Importancia	24
Objetivo	25
Actividades	25
Organización del Trabajo.....	26
Capítulo II: Materiales y Métodos.....	27
Materiales.....	27
Métodos	28
Base de Datos.....	29
Preprocesamiento	32
CGAN.....	41
Generador.....	43
Discriminador.....	46
Entrenamiento.....	49

	11
Procesamiento Final.....	51
Matriz de confusión	53
Aplicación.....	55
Capítulo III: Análisis de Resultados.....	57
Filtrado de Fases.....	57
Parámetros para el modelo CGAN ideal.....	63
Resultados del modelo CGAN final	70
Señales LP, TC, TR, VT en el tiempo.....	75
Evaluación de señales sintéticas microsísmicas por medio de herramientas.....	83
Clasificación de microsismos por medio del uso del primer autoencoder.....	83
Clasificación de microsismos por medio del uso del segundo autoencoder	85
Software para clasificación de microsismos por medio del uso de DT, k-NN, SVM y Votación.....	87
Capítulo IV Conclusiones y Trabajos Futuros.....	93
Conclusiones.....	93
Trabajos Futuros	95
Referencias.....	97

Índice de Tablas

Tabla 1 Características de las computadoras.....	27
Tabla 2 Arquitectura del generador.....	43
Tabla 3 Tamaño del kernel para el generador.....	46
Tabla 4 Arquitectura del discriminador.....	46
Tabla 5 Tamaño del kernel para el discriminador.....	48
Tabla 6 Matriz de confusión para cuatro clases.....	53
Tabla 7 Parámetros variantes para pruebas en el modelo CGAN.....	65
Tabla 8 Métricas de evaluación del primer autoencoder.....	84
Tabla 9 Métricas de evaluación del segundo autoencoder	86
Tabla 10 Métricas de evaluación del clasificador por medio del uso de DT	88
Tabla 11 Métricas de evaluación del clasificador por medio del uso de k-NN	89
Tabla 12 Métricas de evaluación del clasificador por medio del uso de SVM	90
Tabla 13 Métricas de evaluación del clasificador por medio del uso de Votación	91
Tabla 14 Resultados de las herramientas empleadas	92

Índice de Figuras

Figura 1 Diagrama de bloques del proyecto	28
Figura 2 Espectros de frecuencia para microsismos VT, LP, TR, TC	30
Figura 3 Volcán Llaima y sus estaciones sísmicas	31
Figura 4 Diagrama de bloques del preprocesamiento	33
Figura 5 Señales originales en el dominio del tiempo.....	34
Figura 6 Diferencia entre la señal filtrada y la no filtrada en el dominio de la frecuencia	35
Figura 7 Señales normalizadas en el dominio del tiempo.....	37
Figura 8 FFT de la Señal Temporal Real para cada uno de los eventos LP, VT, TR y TC.....	40
Figura 9 Red CGAN para señales sintéticas	41
Figura 10 Diagrama de bloques de la CGAN	42
Figura 11 Red generadora de la CGAN.....	44
Figura 12 Red discriminadora de la CGAN	47
Figura 13 Diagrama de bloques del posprocesamiento.....	51
Figura 14 Pantalla principal de la interfaz gráfica	55
Figura 15 Generador de señales con el uso de CGAN.....	56
Figura 16 Fases totales de los eventos LP y TC	58
Figura 17 Fases totales de los eventos TR y VT	58

Figura 18 Ejemplo de criterio en señales LP para selección de fases	59
Figura 19 Ejemplo de criterio en señales TC para selección de fases.....	59
Figura 20 Fases filtradas.....	61
Figura 21 Señal origina y señal desplazada.....	62
Figura 22 Modo colapso con exceso de filtros.....	63
Figura 23 Respuesta en frecuencia de los cuatro eventos generados con un modelo colapsado	64
Figura 24 Falla de convergencia con un epoch grande	65
Figura 25 Respuesta en frecuencia de los cuatro eventos generados con demasiadas épocas	66
Figura 26 Caso 1: 90 Epoch, 128 minibatch y 2520 iteraciones	67
Figura 27 Respuesta en frecuencia generadas con el Caso 1	67
Figura 28 Caso 2: 90 Epoch, 15 minibatch y 21510 iteraciones	68
Figura 29 Respuesta en frecuencia generadas con el Caso 2	68
Figura 30 Entrenamiento con 13 épocas y 5 de minibatch size.....	70
Figura 31 Magnitudes frecuenciales de eventos LP reales.....	71
Figura 32 Magnitudes frecuenciales de eventos LP sintéticos	71
Figura 33 Magnitudes frecuenciales de eventos TC reales	72
Figura 34 Magnitudes frecuenciales de eventos TC sintéticos	72

Figura 35	Magnitudes frecuenciales de eventos TR reales	73
Figura 36	Magnitudes frecuenciales de eventos TR sintéticos	73
Figura 37	Magnitudes frecuenciales de eventos VT reales	74
Figura 38	Magnitudes frecuenciales de eventos VT sintéticos	74
Figura 39	Magnitud LP sintético, fase LP natural y señal sintética microsísmica LP	76
Figura 40	Magnitud VT sintético, fase VT natural y señal sintética microsísmica de tipo VT	76
Figura 41	Magnitud TC sintético, fase TC natural y señal sintética microsísmica de tipo TC.....	77
Figura 42	Magnitud TR sintético, fase TR natural y señal sintética microsísmica de tipo TR.....	77
Figura 43	Señal microsísmica artificial LP y su PSD	79
Figura 44	Señal microsísmica real LP y su PSD	79
Figura 45	Señal microsísmica artificial TC y su PSD.....	80
Figura 46	Señal microsísmica real TC y su PSD.....	80
Figura 47	Señal microsísmica artificial TR y su PSD.....	81
Figura 48	Señal microsísmica real TR y su PSD.....	81
Figura 49	Señal microsísmica artificial VT y su PSD.....	82
Figura 50	Señal microsísmica real VT y su PSD.....	82
Figura 51	Matriz de confusión de primer autoencoder.....	84
Figura 52	Matriz de confusión del segundo autoencoder	86

Figura 53 Matriz de confusión del sistema de clasificación por medio del uso de DT88

Figura 54 Matriz de confusión del sistema de clasificación por medio del uso de k-NN89

Figura 55 Matriz de confusión del sistema de clasificación por medio del uso de SVM.....90

Figura 56 Matriz de confusión del sistema de clasificación por medio del uso de Votación91

Resumen

La prevención de catástrofes se ha vuelto trascendental en los últimos años debido al crecimiento demográfico, expansión territorial, falta de programas de control y evaluación de riesgos. En este sentido, establecer sistemas de alerta temprana para eventos naturales como erupciones volcánicas, terremotos y tsunamis son cruciales para proteger vidas humanas. Por esta razón, la aplicación de sistemas de reconocimiento automático de microsismos puede ser efectivos para los sistemas de alerta temprana en posibles erupciones. Sin embargo, la creación de estos sistemas se ve obstaculizada por la falta de información disponible. Por lo tanto, existe la necesidad de generar bases de datos sintéticas mediante el empleo de técnicas de *Machine Learning* y *Deep Learning*. De esta manera, se puede aumentar el tamaño de las bases de datos disponibles. El presente trabajo se enfoca en el uso del modelo de red neuronal adversario generativo condicional (CGAN, del inglés *Conditional Generative Adversarial Network*) que se fundamenta en la combinación de redes generativo y discriminatorio el cual se utiliza para generar señales que se puedan presentar en un volcán activo. Este trabajo se constituye de varias etapas y comienza con el preprocesamiento en donde, las señales de los distintos eventos deben ser acondicionadas para posteriormente proseguir con la etapa de diseño de la CGAN. Para este diseño se busca encontrar los parámetros que mejoren el entrenamiento y la generación de señales sintéticas. En el posprocesamiento se busca reconstruir los eventos en el dominio del tiempo con el uso de magnitudes frecuenciales sintéticas y fases reales. Finalmente se tiene como resultado las señales sintéticas de microsismos. Además, las señales sintéticas se han evaluado con diferentes métodos de clasificación que no se encuentran dentro del alcance del proyecto actual, pero que se desarrollan en proyectos paralelos. La exactitud de la clasificación es aceptable al usar características en frecuencia y tiempo donde se obtiene, en el mejor de los casos, 81%.

Palabras clave: microsismos, redes neuronales adversarias generativas condicionales, señales sintéticas, Llaima, Volcán.

Abstract

Catastrophe prevention has become transcendental in recent years due to population growth, territorial expansion, lack of control programs and risk assessment. In this sense, establishing early warning systems for natural events such as volcanic eruptions, earthquakes and tsunamis are crucial to protect human lives. For this reason, the application of automatic recognition systems of microseisms can be effective for early warning systems in possible eruptions. However, the creation of these systems is hampered by the lack of available information. Therefore, there is a need to generate synthetic databases through the use of Machine Learning and Deep Learning techniques. In this way, the size of the available databases can be increased. This paper focuses on the use of the Conditional Generative Adversarial Network (CGAN) model, which is based on the combination of generative and discriminatory networks, which is used to generate signals that can be presented in an active volcano. This work consists of several stages starting with preprocessing. In the preprocessing, the signals of the different events must be conditioned to later continue with the stage where the CGAN is designed. For this design, we seek to find the parameters that improve the training and the generation of synthetic signals. In the post processing, we seek to reconstruct the events in the time domain using synthetic frequency magnitudes and real phases. Finally, the synthetic signals of microseisms are obtained. In addition, the synthetic signals have been evaluated using different classification methods that are not within the scope of the current project but are being developed in parallel projects. The accuracy of the classification is acceptable when using characteristics in frequency and time where it is obtained, in the best of cases, 81%.

Keywords: microseisms, conditional generative adversarial neural networks, synthetic signals, Llaima, volcano.

Capítulo I

Introducción

La sismología es una técnica antigua que permite determinar el estado de los volcanes. Al analizar las diferentes señales microsísmicas asociadas a la actividad de un volcán, se obtiene información sobre posibles intrusiones de magma, alteraciones de cámaras magmáticas, el ascenso de magma antes de una erupción inminente, y más. Por lo general, las erupciones volcánicas están precedidas por microsismos que pueden ocurrir antes del evento (IGN, 2023). En esta región geográfica se localizan varios países en los que las erupciones volcánicas representan una amenaza, que incluyen Estados Unidos, México, Guatemala, Costa Rica, Colombia, Ecuador y Chile.

El volcán Llaima se encuentra en la Región de Araucanía (S 38° 41' – W 71° 44') en Chile, a unos 72 kilómetros al noreste de Temuco, en el límite occidental de la cordillera de los Andes. Tiene una altura de 3.125 metros y una caldera de ocho kilómetros de diámetro. Debido a su historial de erupciones continuas desde 1640, se considera uno de los cuatro volcanes más activos de América del Sur y presenta alto riesgo. Por su ubicación, el volcán Llaima es un atractivo turístico rodeado de poblados con actividad productiva principalmente agrícola y ganadera.

Para garantizar la seguridad de los habitantes, existe una agencia estatal denominada Observatorio Vulcanológico de los Andes del Sur (OVDAS), que supervisa el Llaima y otros 42 volcanes en Chile. Específicamente, para la monitorización del volcán Llaima, la agencia, cuenta con 9 estaciones que recogen de manera constante la actividad sísmica (Canário, Fernandes, & Curilem, Data In Brief, 2021) . Los datos de las estaciones se almacenan y son interpretados con el fin de proveer información a las autoridades para la correcta gestión de emergencias volcánicas, luego estos datos son guardados para tener una base de datos.

Cada señal debe ser analizada y esto además de requerir tiempo, se necesita personal capacitado para que la información importante no sea eliminada. En el caso del volcán Llaima, que se considera uno de los más activos, se generan una gran cantidad de datos que deben ser etiquetados de manera adecuada, lo que puede ser una tarea difícil y laboriosa. Por lo tanto, resulta esencial que la información recopilada sea evaluada y categorizada automáticamente para reducir el tiempo de etiquetado en las actividades sísmico-volcánicas y evitar la pérdida de información importante.

Las señales identificadas en el volcán Llaima incluyen microsismos de tipo Volcano Tectónico (VT, del inglés *Volcano Tectonic*). Estas se refieren a la rotura de rocas dentro del volcán y es causada por fuerzas tectónicas asociadas con el movimiento de fluidos en grietas. Estas señales tienen un espectro de banda ancha máximo de 10 Hz. Los eventos de periodo largo (LP, del inglés *Long Period*) que son causados por el movimiento de magma y gases hacia la superficie del volcán. Su espectro de frecuencia se encuentra en el rango de 0.5 a 5 Hz. Aunque estos eventos suelen ocurrir antes de una erupción volcánica, también son comunes en volcanes sin erupciones. Los Tremores (TR) son señales continuas y de alta amplitud producidas por resonancia de larga duración y la presencia de eventos VT y LP continuos. Estas señales tienen un espectro de banda ancha que oscila entre 0.5 a 3.0 Hz, y disminuyen gradualmente al final del evento. Finalmente, los eventos Tectónicos (TC) no están relacionados con la actividad volcánica y son causados por fallas geológicas (Canário P. , Mello, Curilem, Huenupan, & Rios, 2020).

Antecedentes

La actividad volcánica implica procesos físicos que producen oscilaciones terrestres discretas denominados microsismos, los cuales se generan por el transporte de masa y energía. La monitorización de estos eventos produce un gran volumen de señales, lo que genera desafíos en almacenamiento y procesamiento de dicha información. Para hacer frente a

esto, se requiere el diseño de herramientas eficientes capaces de extraer información relevante y detectar patrones implícitos en los datos. Dado que las erupciones suelen estar precedidas por cambios en la sismicidad, es fundamental modelar los patrones sísmicos para revelar información sobre la probabilidad, el momento y la gravedad de dicha actividad. Desde esta perspectiva, es necesario analizar las características espectrales de múltiples señales para interpretar los eventos del volcán (Canário P. , Mello, Curilem, Huenupan, & Rios, 2020).

Los microsismos que ocurren en un volcán son los LP, VT, TR, Muy Largo Periodo (VLP, del inglés *Very Long Period*), Tornillo (TO), Híbrido (HB) y Acústico (MF, del inglés *Medium Frequency*) (sernageomin, 2023). En general, los LP y VT son los microsismos más comunes en los volcanes, mientras que los demás tipos de señales ocurren con menor frecuencia y no son tan característicos. A pesar de esto, la mayoría de los observatorios todavía procesan visualmente los datos de las señales, lo que puede ser una tarea difícil debido a que la interpretación y análisis de las señales puede cambiar con el tiempo.

Para la caracterización de microsismos se han utilizado métodos como son estimadores de entropía, análisis de densidad espectral de potencia (Jaramillo, León, Lara, Benítez, & Ruiz, 2014), análisis por medio de las transformadas de Fourier y Wavelet (Lara, Bernal, Saltos, & Benítez, 2015) , además de técnicas como son las de detección de actividad de voz (VAD, del inglés *Voice Activity Detection*) (Lara, Rodriguez, & Larco, 2020). Se ha utilizado además clasificación mediante técnicas de aprendizaje automático con algoritmos como son k vecinos más cercanos (k -NN, del inglés *k-Nearest Neighbors*) (Lara R. , Benítez, Carrera, Ruiz, & Álvarez, 2016), árboles de decisión (DT, del inglés *Decision Tree*) (Lara, y otros, 2020), máquina de vectores de soporte (SVM, del inglés *Support Vector Machine*) (Lara R. , Benítez, Carrera, Ruiz, & Álvarez, 2019) y bosque aleatorio. Además de técnicas de aprendizaje profundo con el uso de redes neuronales convolucionales (CNN, del inglés *Convolutional Neural Network*) (Lara, Lara, Larco, Carrera, & León, 2021). No obstante, hay una escasez de

bases de datos de microsismos etiquetadas disponibles al público para alimentar los modelos de clasificación debido a la falta de procesamiento y categorización de dichas señales. La escasez de datos se la pueden solventar mediante técnicas de generación de señales sintéticas.

La generación de señales sintéticas puede lograrse a través de diversas técnicas. Una opción consiste en emplear modelos matemáticos. Además, es posible incorporar ruido gaussiano o uniforme en señales existentes con el propósito de simular condiciones específicas. Otra alternativa es utilizar generación basada en distribuciones estadísticas para crear valores sintéticos. Por otro lado, los métodos de interpolación y extrapolación permiten estimar valores, mientras que las técnicas de muestreo aleatorio generan muestras sintéticas que conservan propiedades estadísticas. Además, la técnica de *Bootstrap* se puede aplicar para simular múltiples muestras a partir de datos y estimar la variabilidad de propiedades de señales sintéticas como en los trabajos de (Pérez, 2022) y (Zapata, 2022). Finalmente, los modelos de aprendizaje automático, incluidas las GAN, son efectivos para generar señales realistas basadas en patrones aprendidos de datos existentes.

Una serie de trabajos, que tiene que ver con la técnica de *data augmentation*, se han realizado con el uso de la técnica de GAN. En uno de estos se realiza la creación de datos sintéticos para mejorar la clasificación de lesiones hepáticas (Frid-Adar, Klang, Amitai, Goldberger, & Greenspan, 2018), donde poseen imágenes de tomografía computarizada solo con un número de 182 lesiones hepáticas divididas en quistes, metástasis y hemangiomas. Al clasificarlas sin usar el aumento de datos, no da buenos resultados de sensibilidad, pero al utilizar datos aumentados la sensibilidad casi alcanza el 100%. Otro trabajo que se puede mencionar es el usado para el reconocimiento de emociones audiovisuales de modo multimodal (Ma, Li, Ni, Huang, & Zhang, 2022). La propuesta implica la identificación de estados emocionales humanos que fusionen audio e imágenes, con el propósito de emplearlos

en interacciones con robots. El problema principal es la adquisición de datos y esto se vuelve más complicado al agregar el factor de que los datos deben ser multimodales. Para solucionar el problema se usó generadores y discriminadores de audio y visual que se basan en la correlación máxima de Hirschfeld- Gebelein- Rényi, y finalmente se obtuvo alta efectividad.

En cuanto a las CGAN se ha usado para trabajos como es la detección de COVID-19 con el uso de radiografías de tórax (Loey, Manogaran, & Nour, 2020). El especial interés es la detección temprana, para esto se usa un modelo de aprendizaje de transferencia profunda sobre imágenes de tomografías del tórax. El problema es la limitada cantidad de imágenes y con el uso de CGAN se mejora el rendimiento de la clasificación. El proyecto ProSe se enfoca en la interacción robot-humano, por tanto, es necesario el reconocimiento de emociones para que la comunicación sea fluida y natural. Como solución se usó una red adversarial generativa condicional para el control independiente de las expresiones. Además, se consideran factores como la iluminación y la posición del cráneo.

Con respecto a la generación de señales volcánicas artificiales, se tienen varios trabajos. El primero que se puede mencionar es el que mediante un modelo CGAN se originan señales VT y LP. Este trabajo de la Escuela Politécnica Nacional (Yépez, 2022) se enfocaron en la generación de estas señales con ayuda de Python y con una base de datos del Instituto Geofísico de la Escuela Politécnica Nacional (IGEPN) del volcán Cotopaxi. Lograron tener señales aceptables y parecidas a las originales tanto en frecuencia como en tiempo. También se puede mencionar el trabajo de titulación hecho en la Universidad de las Fuerzas Armadas ESPE en el que se generaron señales sintéticas de microsismos del volcán Cotopaxi. A través del uso de un modelo de red neuronal adversario generativo condicional (Galarza & Vega, 2022) se generaron señales LP y VT. En este caso se usó MATLAB® y la base de datos fue proporcionada por el IGEPN.

Justificación e Importancia

En una erupción volcánica se presentan diversos fenómenos físicos que, además de alterar el entorno natural, conllevan diversos riesgos para la sociedad. En este contexto, la generación y análisis de señales microsísmicas, junto con su clasificación, detección, reconocimiento y monitorización, desempeñan un propósito central: salvaguardar vidas humanas en zonas que están en peligro inminente debido a la actividad volcánica. Con la monitorización de distintas zonas donde se produce actividad volcánica se identifican patrones de señales que preceden erupciones. Por medio de esto se posibilita emitir alertas tempranas y permite evacuaciones oportunas. La mejora de sistemas automatizados aumenta la precisión, para minimizar el riesgo de alertas erróneas. Además, el análisis profundo de señales proporciona una comprensión para interpretar eventos volcánicos y tomar decisiones informadas. Gracias a esto se busca reducir la pérdida de vidas y la protección efectiva de las comunidades en peligro.

Por este motivo, se requiere generar señales artificiales que sean tan parecidas a las reales que no se puedan distinguir entre ellas. De esta manera, se puede aumentar el tamaño de las bases de datos disponibles y mejorar los sistemas de detección para señales microsísmicas, con el cual se dará una alerta temprana oportuna y confiable. Para esto el presente trabajo se enfoca en el uso del modelo CGAN, que es una técnica de *Deep Learning* que se basa en el uso de redes neuronales artificiales para la generación de datos sintéticos condicionados a una entrada y se fundamenta en la combinación de redes generativo y discriminatorio. El submodelo generativo se encarga de generar muestras de datos sintéticos, mientras que el discriminador distingue entre las muestras generadas por la red generativa y las reales. La idea es que el modelo generativo aprenda a crear muestras que sean difíciles de distinguir de las muestras reales, y que el discriminador se actualice para identificar las diferencias entre muestras sintéticas y reales.

La creación de bases de datos sintéticas de microsismos mediante el uso de técnicas de *Deep Learning*, con el uso de la CGAN, es importante por varias razones. Una de las principales es la escasez de datos ya que, en algunos casos, la cantidad de datos disponibles para entrenar modelos de aprendizaje automático es limitada. La creación de bases de datos sintéticas puede ayudar a aumentar el tamaño de la base de datos. La utilización de CGAN permite crear señales sintéticas que sean más cercanas a las señales reales. Esto se debe a que los generadores de CGAN utilizan información adicional para crear señales que se ajusten a las características de cada evento. Se debe considerar la dificultad para la recolección de datos en microsismos en tiempo real y el trabajo de etiquetado que puede ser costoso y requiere gran cantidad de tiempo y recurso humano. También se puede mencionar que permite la creación de señales sintéticas variables. Esto permite que los modelos de aprendizaje automático se entrenen en una gama de situaciones y así mejoren su rendimiento y capacidad de generalización. Finalmente, se puede decir que la creación de bases de datos sintéticas permite la simulación de microsismos futuros y el desarrollo de modelos predictivos para el manejo de desastres.

Objetivo

Crear bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning.

Actividades

- Identificación de las propiedades y variables del problema a tener en cuenta.
- Reunión de la base de datos representativa, con datos provistos por el Observatorio Vulcanológico de los Andes del Sur.
- Creación de una estructura de datos (temporales, espaciales, otros.) con un soporte común.
- Caracterización de microsismos.

- Generación de bases sintéticas con Deep Learning.
- Pruebas y evaluación de desempeño.

Organización del Trabajo

El trabajo está organizado en cuatro capítulos, en donde se explica de manera teórica y práctica cada parte que ha tenido el desarrollo de este proyecto. Para el primer capítulo se tiene la introducción donde se incluyen los antecedentes para realizar este trabajo, la justificación y la importancia que trata sobre el porqué es necesario la realización de este proyecto. Además, se incluye el objetivo general como las actividades alineadas a lo que se busca del proyecto. Por último, se tratan los trabajos previos en el estado del arte.

El segundo capítulo se centra en el tratamiento de las señales obtenidas de la base de datos perteneciente al volcán Llaima. Primero se hace el preprocesamiento, posteriormente explicar la metodología que se siguió para la creación del modelo CGAN. Se explica además el posprocesamiento para luego enfocarse en el entrenamiento y la determinación de los mejores parámetros que brinde buenos resultados y costo computacional. Además de evitar problemas de convergencia con el uso de la herramienta que permite la monitorización.

El tercer capítulo trata del análisis de los resultados obtenidos al variar distintos parámetros. Además de la monitorización de cada uno de estos entrenamientos y las señales generadas tanto de los modelos no óptimos como del modelo seleccionado. Adicionalmente se hace la comparación con las señales reales del volcán Llaima. Finalmente se evalúan los resultados de la red final con el uso de herramientas creadas con aprendizaje automático.

En el último capítulo, es decir, el cuarto, se contemplan las conclusiones del trabajo y también se incluye un análisis para futuros trabajos.

Capítulo II: Materiales y Métodos


El presente capítulo trata de los materiales y software empleados para la programación del modelo CGAN. Incluye el preprocesamiento, el entrenamiento y el posprocesamiento, además del diseño de la aplicación. Luego de esto, la metodología aplicada para el desarrollo de las distintas partes del modelo que son el generador y el discriminador, su respectivo entrenamiento, además de la explicación detallada del pre y posprocesamiento.

Materiales

En el trabajo se utiliza el *software* MATLAB® R2022a y la utilización de la herramienta *Deep Learning Toolbox™* que sirve para diseñar e implementar redes neuronales profundas con algoritmos y modelos previamente entrenados. El entrenamiento es fuerte y requiere características para las computadoras, como se puede ver en la Tabla 1. Esto permite que el tiempo requerido en las distintas fases no sea exagerado.

Tabla 1

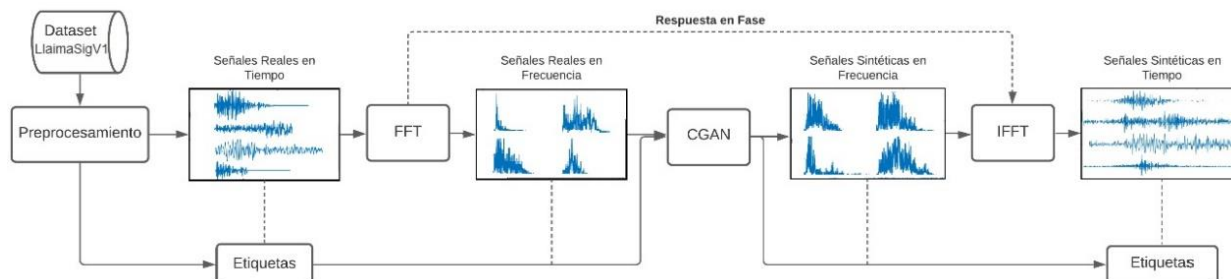
Características de las computadoras

DELL® Precision 5820 Tower	DELL Inspiron 15 i5577 5328BLK
	
<ul style="list-style-type: none"> - Intel® Xeon® W-2223 (4 cores, 8 threads, 3.60 GHz to 3.90 GHz) - 16 GB RAM - 512 GB SSD 	<ul style="list-style-type: none"> - Intel core I7 7th Gen (4 core, 8 threads, 2.8 GHz) - 8 GB RAM - 128 GB SSD y 1 TB - Nvidia GeForce GTX 1050 4GB

Métodos

Figura 1

Diagrama de bloques del proyecto



Nota. Generación de señales sintéticas de microsismos por medio del uso de un modelo de red neuronal adversario generativo condicional. Adaptado de (Galarza & Vega, 2022)

En la Figura 1 se presenta el diagrama de bloques que explica de manera resumida los pasos que se van a seguir en el proyecto. Como primera instancia, la red debe ser alimentada por una base de datos de microsismos pertenecientes al volcán Llaima que se la puede obtener en (Canário J. P., Mello, Rios, Huenupan, & Curilem, 2020). Los datos proporcionados requieren un preprocesamiento debido a dos razones fundamentales. En primer lugar, los archivos de las señales no están en un formato compatible con el software y están separados, es decir, carecen de una estructura matricial. Para abordar esto, se ha generado el archivo "LlaimaSigV1" y posteriormente convertido al formato *.mat* para mayor conveniencia. En segundo lugar, con el uso de este archivo, se extraen las señales temporales, sin embargo, antes de utilizarlas, se requiere un cambio al dominio de la frecuencia mediante la Transformada Rápida de Fourier (FFT, del inglés *Fast Fourier Transform*). Una vez obtenidas las señales en el dominio de la frecuencia, junto con las etiquetas de los respectivos eventos, se alimenta el modelo CGAN.

Las señales generadas a la salida de la red CGAN están en el dominio de la frecuencia y deben ser transformadas al dominio del tiempo con el uso de la transformada rápida de

Fourier inversa (IFFT, del inglés Inverse Fast Fourier Transform). Finalmente se realiza un posprocesamiento para obtener las señales etiquetadas sintéticas. Para explicar estos procesos de manera resumida, se tiene un diagrama de bloques donde incluye el preprocesamiento, entrenamiento de la CGAN y el posprocesamiento.

Base de Datos

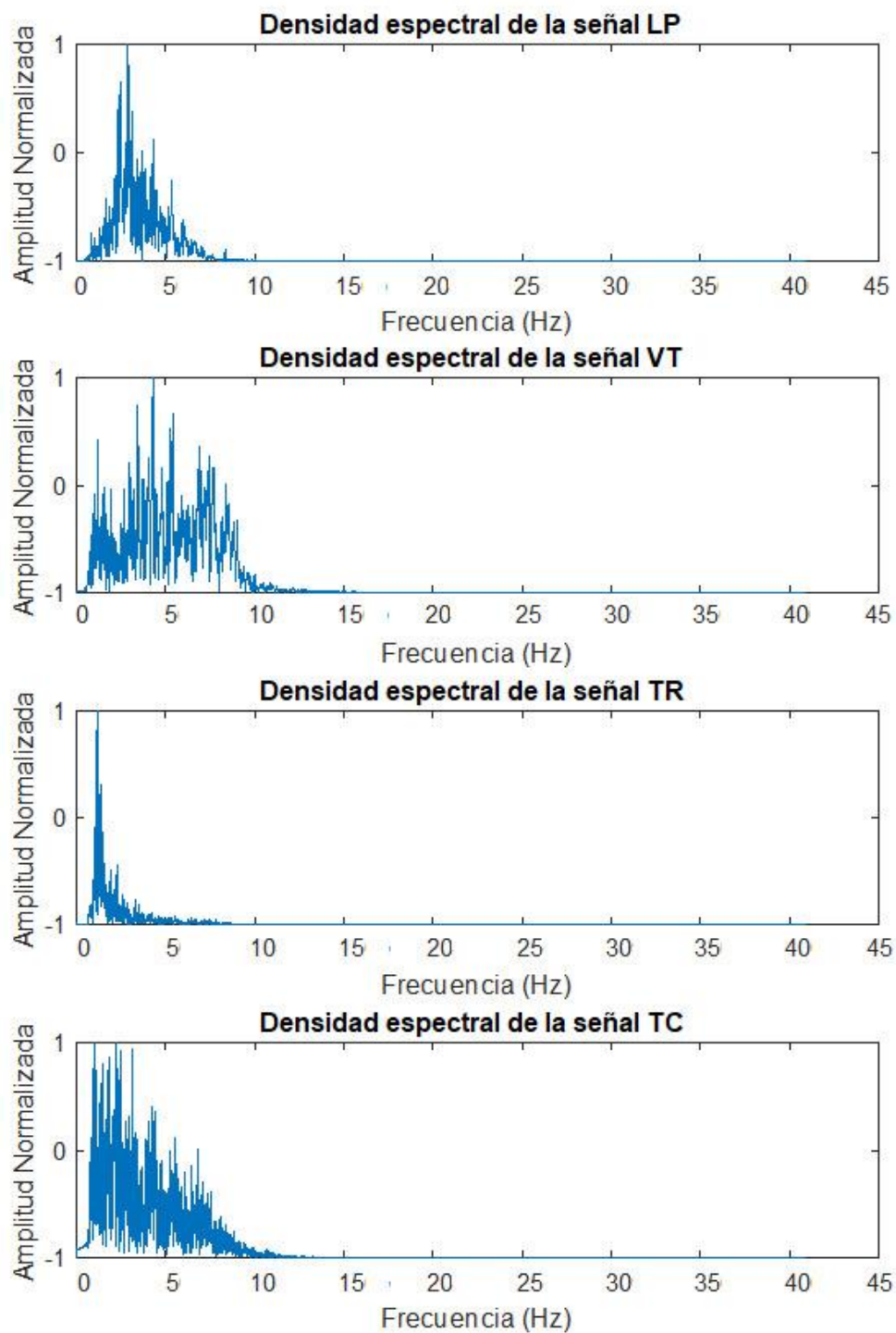
Los microsismos presentes en el volcán Llaima incluyen VT, que involucra la rotura de rocas al interior de un edificio volcánico. Presentan frecuencias bajas debido a la resonancia causada por la presión inducida por el desplazamiento de fluidos magmáticos y el aumento en el volumen de fluidos geotérmicos de alta presión (Servicio Nacional de Geología y Minería, 2019). En otras palabras, la causa de estos eventos está conectado con el movimiento de fluidos en grietas. Estas señales tienen el espectro de banda ancha máximo de 10 Hz como se comprueba en la Figura 2(b) (Canário P. , Mello, Curilem, Huenupan, & Rios, 2020).

Los eventos LP son causados por el movimiento magmático o hidrotermal y gases a través de grietas y conductos (Servicio Nacional de Geología y Minería, 2019). Su espectro de frecuencia se encuentra en el rango de 0.5 a 5 Hz, esto se puede comprobar en la Figura 2(a). Aunque estos eventos tienden a manifestarse previo a una erupción volcánica, también se observan con frecuencia en volcanes que no están en proceso de erupción (Canário P. , Mello, Curilem, Huenupan, & Rios, 2020).

Los TR son señales continuas o sucesión de pequeños sismos de alta amplitud generados debido a resonancia prolongada y la presencia de eventos VT y LP continuos. Estas señales tienen un espectro de banda ancha que oscila entre 0.5 a 3.0 Hz, como se observa en la Figura 2(c). Su intensidad disminuye gradualmente al final del evento. Por último, los eventos TC no guardan relación con la actividad volcánica, ya que son causados por fallas geológicas (Canário P. , Mello, Curilem, Huenupan, & Rios, 2020), el espectro se observa en la Figura 2(d).

Figura 2

Espectros de frecuencia para microsismos VT, LP, TR, TC



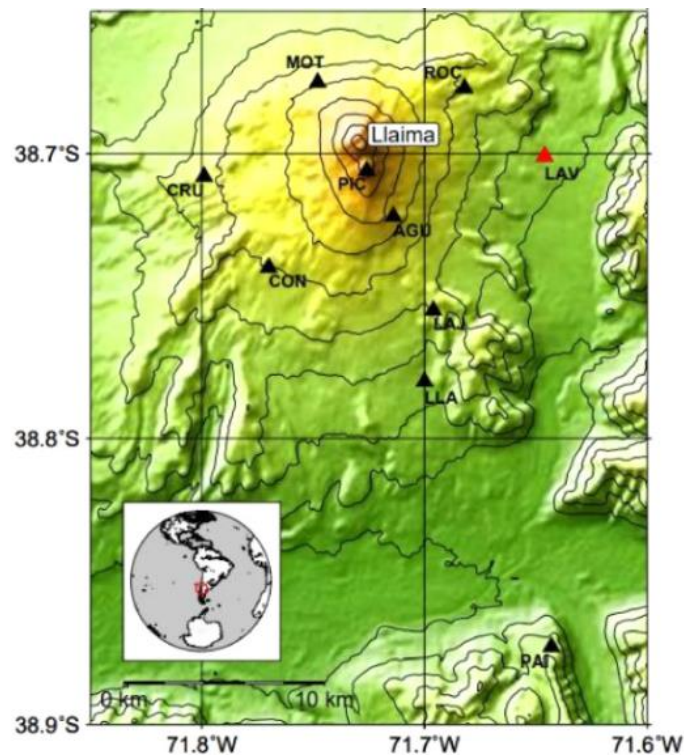
Nota. La Figura representa ejemplos de microsismos: (a) LP, (b) VT, (c) TR y (d) TC.

Para el entrenamiento se utiliza el conjunto de datos (Canário J. P., Mello, Rios, Huenupan, & Curilem, 2020) que incluyen 3592 señales de tiempo-frecuencia 1D provenientes de una de las siete estaciones que monitorizan el volcán Llaima, el LAV, como se puede observar en la Figura 3. Estas señales fueron recolectadas en un periodo de tiempo que va desde 2010 hasta 2016 y etiquetadas por especialistas del OVDAS en cuatro clases de la siguiente manera:

- 304 VT
- 1310 LP
- 490 TR
- 1488 TC

Figura 3

Volcán Llaima y sus estaciones sísmicas



Nota. Estaciones de monitorización ubicadas en el volcán Llaima. Adaptado de: (Canário, Fernandes, & Curilem, Data In Brief, 2021)

Las señales de la base de datos originales se muestrearon a 100 Hz se filtraron por medio de un filtro de paso de banda Butterworth de décimo orden en el rango [1,10] Hz para preservar el ancho de banda que contiene el rango de interés. Estos datos fueron almacenados en formato hdf5 y tienen que ser cambiados a *.mat* para poder usarlos posteriormente en el *software* de MATLAB®.

Preprocesamiento

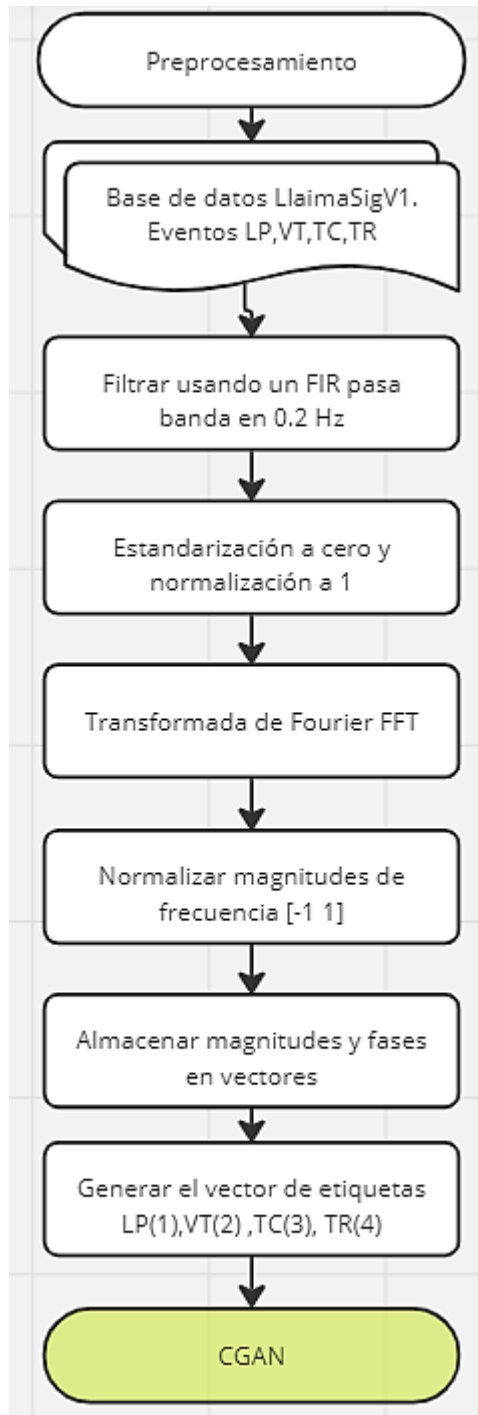
La base de datos tiene cuatro tipos de microsismos y cada evento posee características como son el país, la estación en donde se obtuvo y el rango de años en el que se hicieron las mediciones. También se incluye la frecuencia de muestreo, el tipo de evento, la duración, el punto de inicio y fin. Finalmente, se dispone de una columna de matrices de tipo double que abarcan todos los eventos, cada uno con dimensiones de 1x6000. En este contexto, el valor 6000 representa el intervalo de tiempo, en donde cada evento tiene una duración de 60 segundos. Todos los eventos son utilizados con ocurrencia de 304 VT, 1310 LP ,490 TR y 1488 TC con un total de $M = 3592$ microsismos. El total de señales, $s = \text{señal de 6000 muestras}$, se representan en una matriz S a continuación como:

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

Para una visión general del preprocesamiento se tiene su diagrama de bloques en la Figura 4.

Figura 4

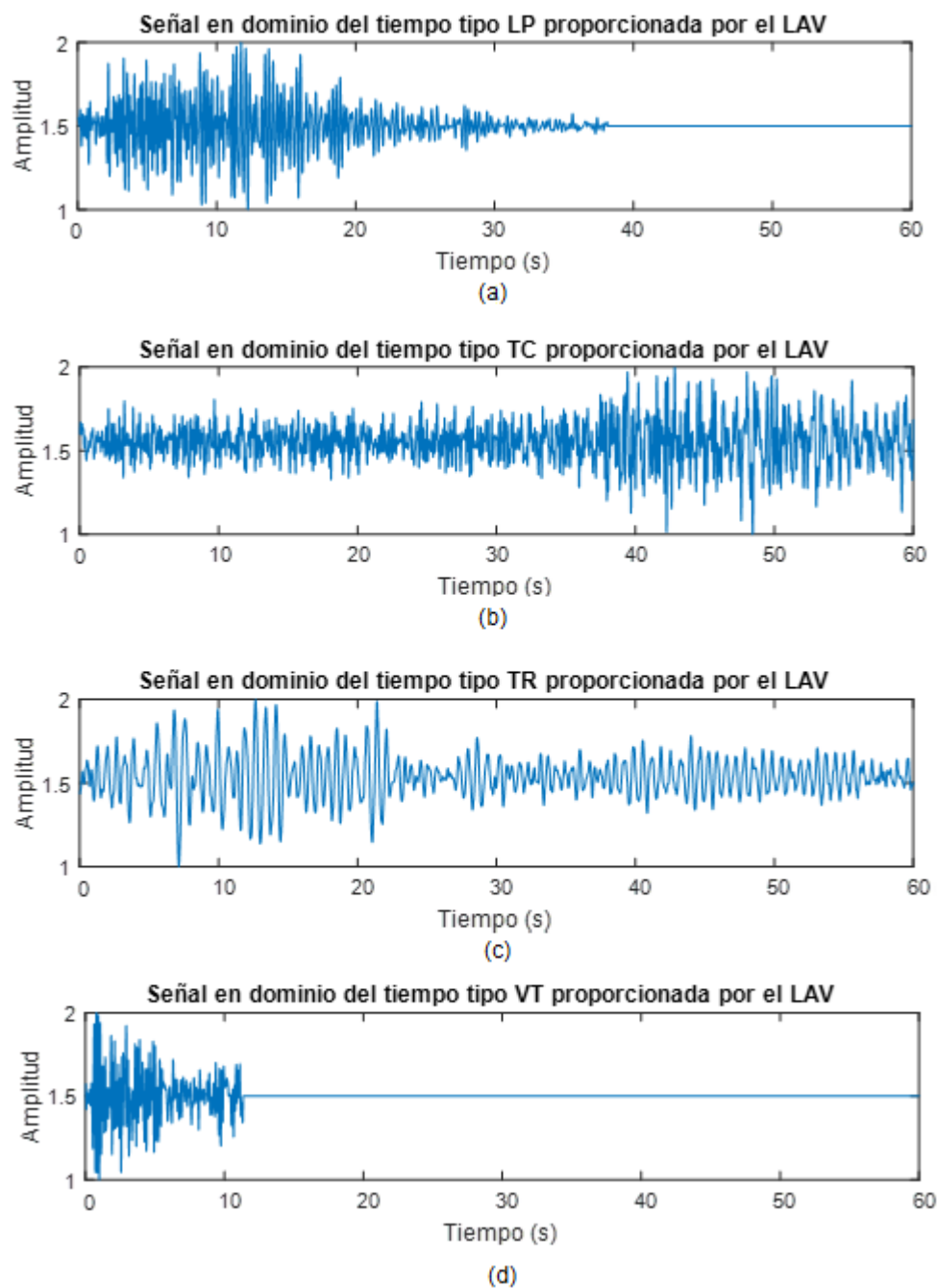
Diagrama de bloques del preprocesamiento



Las señales originales se pueden observar en la Figura 5, representadas en el dominio del tiempo y con su amplitud original.

Figura 5

Señales originales en el dominio del tiempo



Nota. La Figura representa señales originales de microsismos en el dominio del tiempo: (a) LP, (b) TC, (c) TR y (d) VT.

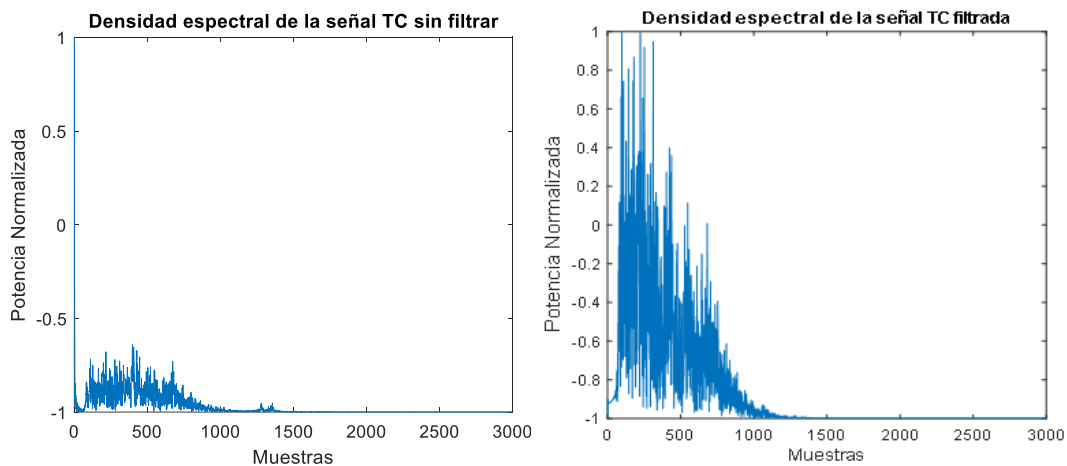
De la base de datos LlaimaSigV1 se necesitan las señales y los tipos de evento que corresponden a las etiquetas. Para la banda de 0.2 Hz, que corresponde al choque del mar contra la superficie terrestre, es filtrada por medio de un FIR pasa banda de orden 256 en el rango de frecuencias de 0.7 Hz a 49.5 Hz. Para representar esta operación se tiene $h_M\{S_M\}$ en donde h_M es el filtro que se aplica a las señales S_M . El resultado se guarda en una matriz como se muestra a continuación:

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

El efecto de este proceso se ve en el dominio de la frecuencia y no se aprecia en el tiempo. Este paso es muy importante debido a que, si no se hace un filtraje correcto, no se pueden apreciar las características en el dominio de la frecuencia. La diferencia entre una señal filtrada de la que no se puede observar en la Figura 6.

Figura 6

Diferencia entre la señal filtrada y la no filtrada en el dominio de la frecuencia



La técnica z-score o puntuación estándar, es un método que describe un punto de datos en relación con la desviación media y estándar de un grupo de puntos. En esta técnica se asignan los datos en una distribución cuya media sea 0 y desviación estándar de 1. De esta manera, se puede determinar la posición relativa de cada punto de datos en relación con los

demás y con la distribución general (ORACLE, s.f.). La técnica de z-score se usa para la estandarización (señal centrada en “0”), y se la representa con el operador matemático $zS_M\{h_M\}$ que se aplica al conjunto de microsismos filtrados H . Cada señal filtrada h_M pasa por el proceso de estandarización y el resultado de esta operación se guarda en la matriz ZS cómo se observa a continuación:

$$ZS = \{zS_1, zS_2, zS_3, \dots, zS_M\} \quad (3)$$

Con las señales estandarizadas se procede a la normalización. La normalización del tipo máximo ayuda a tener las amplitudes balanceadas a través del uso de la siguiente ecuación:

$$X' = \frac{X}{|X_{max}|} \quad (3)$$

donde X' es el vector normalizado, X el vector por normalizar y X_{max} el valor máximo del vector (Galarza & Vega, 2022). Al aplicar este tipo de normalización a las señales estandarizadas se tiene el operador $nm_M\{zS_M\}$, es decir, cada señal que ha sido estandarizada zS_M pasa por la normalización de tipo máximo. Al adaptar la ecuación (3) con el uso de microsismos se tiene la ecuación:

$$nm = \frac{zS}{|zS_{max}|} \quad (4)$$

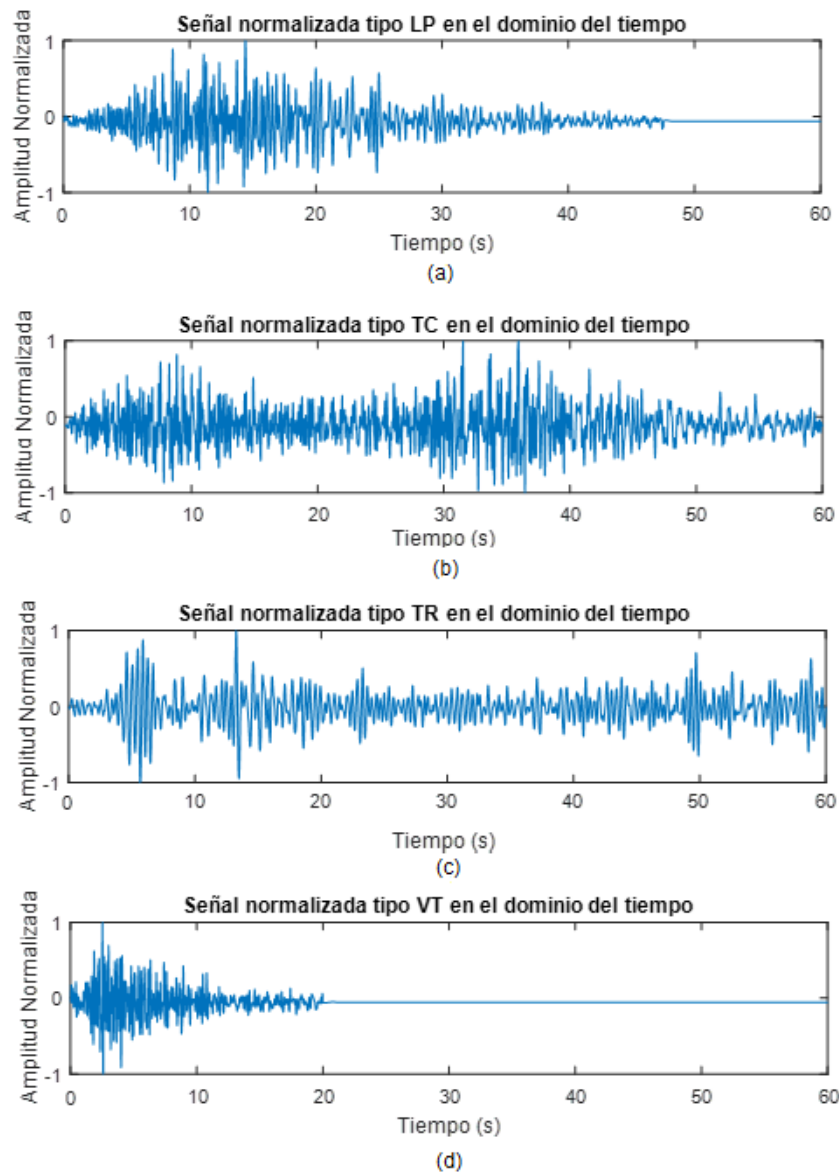
El resultado de esta operación es guardado en la matriz de la manera indicada abajo:

$$NM = \{nm_1, nm_2, nm_3, \dots, nm_M\} \quad (5)$$

Las señales originales sin este preprocesamiento se pueden observar en la Figura 5 y las señales con la amplitud normalizada en el dominio del tiempo se observan en la Figura 7.

Figura 7

Señales normalizadas en el dominio del tiempo



Nota. La Figura representa señales normalizadas originales de microsismos en el dominio del tiempo: (a) LP, (b) TC, (c) TR y (d) VT.

Con las señales temporales que han sido filtradas, estandarizadas y normalizadas se obtienen las magnitudes m_s y las fases p_s de las señales por medio de la FFT, para el cual se aplica el operador matemático $f_M\{nm_M\}$. La matriz resultante de este operador se lo aprecia a continuación:

$$F = \{f_1, f_2, f_3, \dots, f_M\} \quad (6)$$

Al aplicar la FFT a cada señal se tienen las magnitudes frecuenciales ms y sus respectivas fases ps . Las matrices que representan el conjunto de cada una de estas se representan de acuerdo con la ecuación (7) para las magnitudes y con la ecuación (8) para las fases.

$$MS = \{ms_1, ms_2, ms_3, \dots, ms_M\} \quad (7)$$

$$PS = \{ps_1, ps_2, ps_3, \dots, ps_M\} \quad (8)$$

Las razones por las que se trabaja en el dominio de la frecuencia y no del tiempo es que la clasificación de las señales es en función de los dos dominios, pero se debe considerar que en el tiempo las características son mayores que las en frecuencia. Además, la clasificación visual que realizan las instituciones encargadas de etiquetar eventos microsísmicos, es por medio de la revisión de sus características en frecuencia. El dominio frecuencial permite analizar y comprender mejor las contribuciones de frecuencia y observar características específicas de una señal. Adicionalmente sirve para aplicar filtros, como se hizo en este trabajo, ya que permite eliminar o atenuar componentes no deseados de una señal en el dominio de la frecuencia.

El número de puntos de la FFT debe ser en potencia de 2, por tanto, para el número de muestras de las señales que es 6000 el valor más próximo que cumple con las características antes mencionadas es de $N=8192$ que es alta resolución. Cuando se genera el vector de frecuencia mediante la transformada de Fourier, se produce una simetría hermitiana que hace que el resultado de la frecuencia aparezca dos veces en el vector resultante. Por esta razón, el vector de frecuencia real es de tamaño $N/2=4096$, ya que los valores restantes son el reflejo.

Para finalizar el preprocesamiento, las magnitudes frecuenciales deben pasar por una normalización de amplitud del tipo *Min-Max*. En este contexto se realiza una transformación lineal en los datos originales donde se obtiene los datos escalados en el rango (a, b). Para esto se utiliza la siguiente ecuación:

$$X' = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}} \quad (9)$$

donde X' es el vector normalizado, a es el punto mínimo, b el punto máximo, X_{min} el valor mínimo del vector y X_{max} el valor máximo del vector. Para el caso actual se utiliza el operador matemático $z1_M\{ms_M\}$ con el cual, la normalización de tipo mínimo máximo se aplica a cada una de las magnitudes frecuenciales ms_M , así que al adaptar la ecuación (9) para el caso actual se obtiene la ecuación 10. Además, la matriz que se obtiene al aplicar el operador matemático se observa en la ecuación 11.

$$z1 = a + \frac{(ms - ms_{min})(b - a)}{ms_{max} - ms_{min}} \quad (10)$$

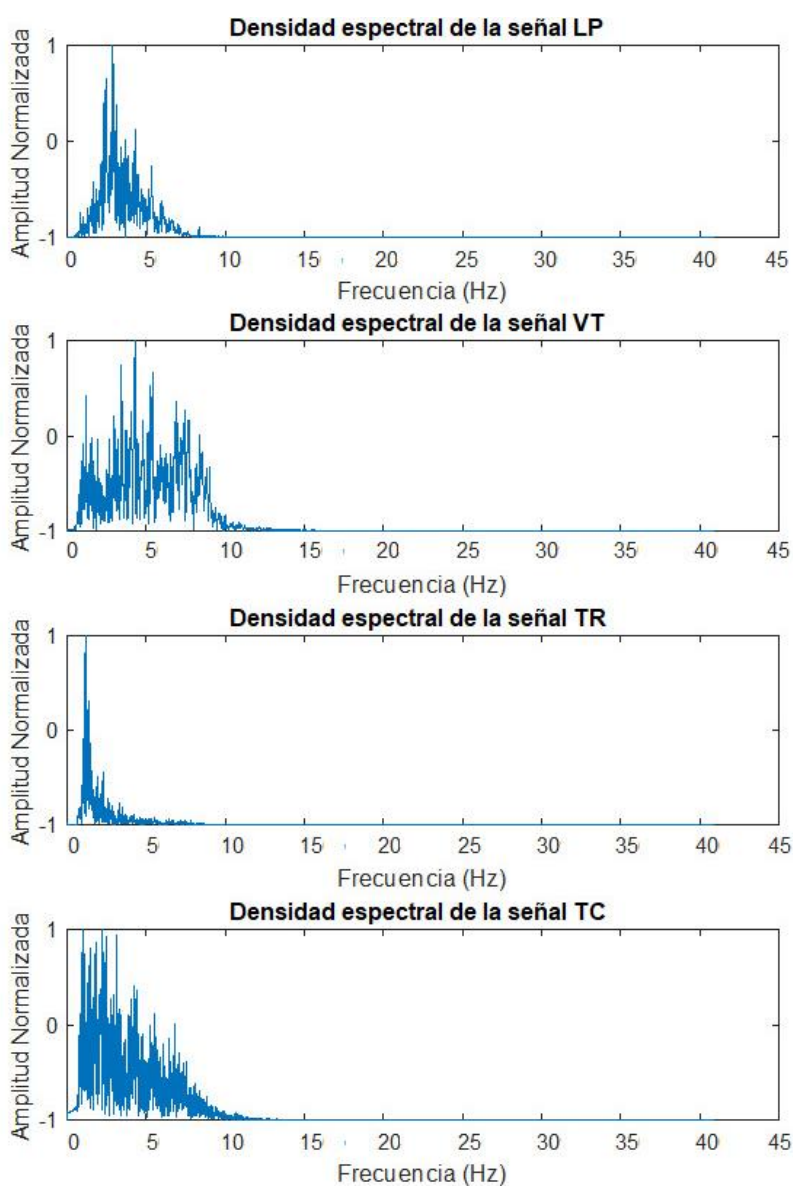
$$Z1 = \{z1_1, z1_2, z1_3, \dots, z1_M\} \quad (11)$$

Los valores de a y b deben ser -1 y 1 respectivamente ya que, dentro de la red, específicamente en el generador, hay una función de activación de tipo tangente hiperbólico. Una función de activación es una herramienta matemática, expresada en la forma de una función $f(x)$, que se agrega a una red neuronal artificial para ayudar en el proceso de aprendizaje. Existen varios activadores entre los cuales se pueden nombrar a sigmoide, tangente hiperbólica, *Rectified Linear Unit* (ReLU), *Leaky ReLu* y *softmax*. La función tangente hiperbólica es una transformación matemática que escala los valores de entrada a un rango entre -1 y 1. A medida que los valores de aumentan, se acercan asintóticamente a 1, mientras que los valores de entrada muy bajos se acercan a -1.

Finalmente, se pueden observar las señales en las que se ha aplicado la FFT como se puede ver en la Figura 8. Con este paso terminado, ya se dispone con las entradas de la red neuronal tanto las señales en magnitud y las etiquetas. Las fases de la señal original se usan en el posprocesamiento para la reconstrucción.

Figura 8

FFT de la Señal Temporal Real para cada uno de los eventos LP, VT, TR y TC



Nota. La Figura representa ejemplos de microsismos: (a) LP, (b) VT, (c) TR y (d) TC.

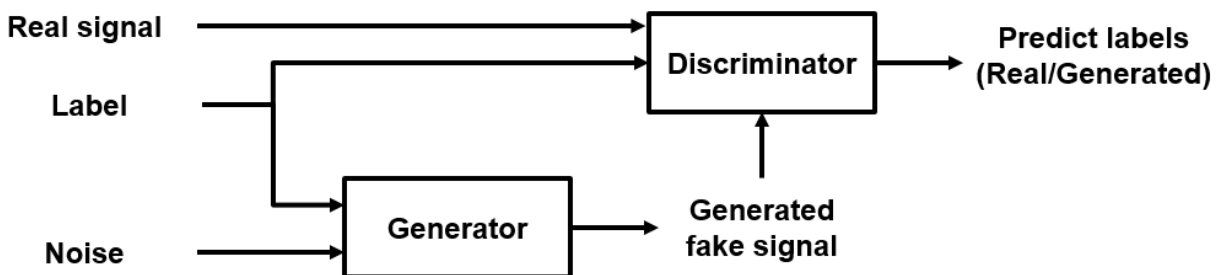
CGAN

Las redes adversarias generativas (GAN, del inglés *Generative Adversarial Network*) son una técnica que permiten crear datos sintéticos que se asemejan a los datos reales. Esta herramienta es muy útil cuando las simulaciones son costosas computacionalmente. Por su parte, las CGAN utilizan etiquetas de datos durante el entrenamiento para generar datos pertenecientes a categorías específicas. El proceso de entrenamiento del CGAN usa redes convolucionales 1-D y un bucle para entrenar, junto con una matriz de aprendizaje profundo (The MathWorks, 2023).

Las CGAN constan de dos redes que se entrenan juntas, el generador y el discriminador. El generador crea datos etiquetados por medio del uso de las etiquetas y una matriz de ruido. Esta situación permite que las muestras generadas sigan la misma estructura de los datos de entrenamiento. Esto se realiza con el fin de que el discriminador las clasifique como reales. La red discriminadora es alimentada con las señales reales, las etiquetas y las señales falsas que fueron generadas previamente por la red generadora. Esta red clasifica las señales reales de las falsas. La red CGAN se la puede observar en la Figura 9.

Figura 9

Red CGAN para señales sintéticas

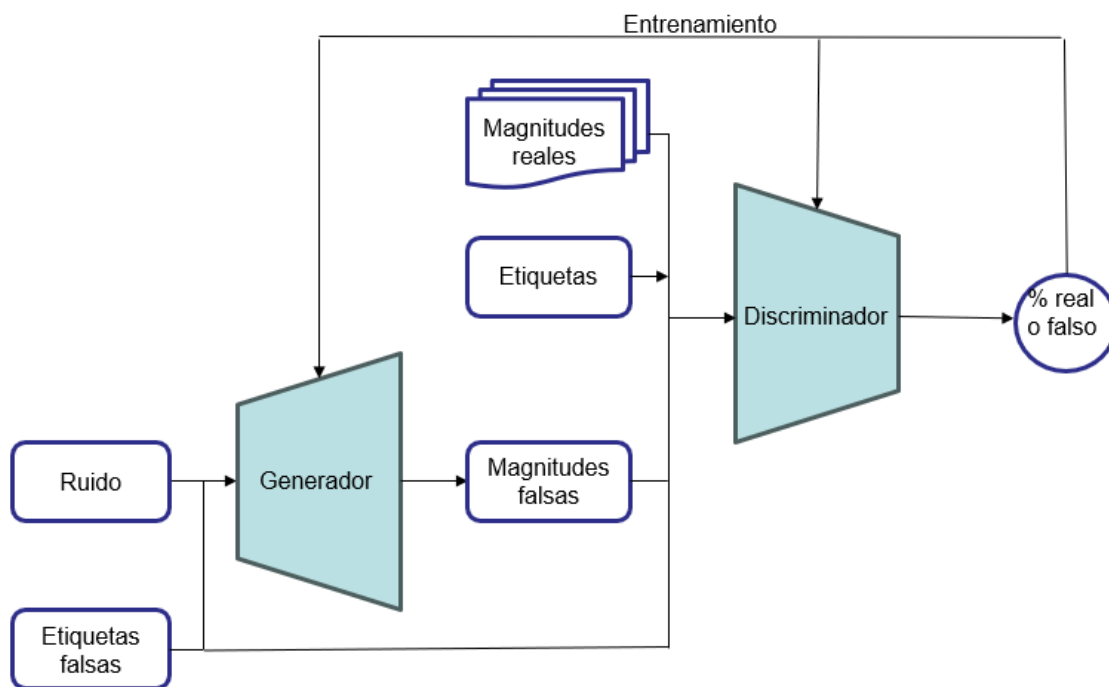


Nota. Generate Synthetic Signals Using Conditional GAN. Adaptado de (The MathWorks, 2023).

El funcionamiento de la red CGAN se explica en la Figura 10, en donde están como bloques principales el generador y el discriminador. En el generador, a su entrada, se tiene un vector de ruido y el ingreso de falsas etiquetas. Como salida se obtiene la magnitud falsa la cual puede ser de cualquiera de los cuatro tipos de microsismos. Estas magnitudes, las magnitudes reales y las etiquetas de los eventos ingresan al discriminador. El discriminador es el encargado de discernir entre una señal falsa de una real al generar una función de pérdida. La función de pérdida determina que tan real es el resultado del generador.

Figura 10

Diagrama de bloques de la CGAN



Este entrenamiento se debe realimentar, con el uso del cálculo de gradientes de las funciones de pérdida. Esto se realiza hasta que los pesos del *kernel* y las etiquetas sean modificados de tal forma que las señales falsas no sean reconocidas como tales por el discriminador.

Generador

La estructura del generador se puede observar en la Tabla 2, en donde el tamaño del filtro d es de 64. Las capas de entrada corresponden al ruido aleatorio de tamaño 100. Este vector debe ser convertido en matrices de 4 por 4 por 1024 debido a que se debe proyectar y remodelar con la función *projectAndReshape* para redimensionar la entrada y tener la salida en tamaño adecuado.

Tabla 2

Arquitectura del generador

Operación	Kernel Size	Salida
<i>Entrada z</i>	100	
<i>projectandReshape</i>	100	(4x1x1024)
<i>Etiquetas</i>	(1x1x1)	
<i>Concat (proj + etiquetas)</i>	(4x1x1024) + (1x1x1)	(4x1x1025)
<i>Trans Conv2D (Stride=1)</i>	(8xd, kernel_size=13)	(16x1x512)
<i>batchNormalization</i>		
<i>ReLU</i>		
<i>Trans Conv2D (Stride=4)</i>	(4xd, kernel_size=6)	(64x1x256)
<i>batchNormalization</i>		
<i>ReLU</i>		
<i>Trans Conv2D (Stride=4)</i>	(2xd, kernel_size=6)	(256x1x128)
<i>batchNormalization</i>		
<i>ReLU</i>		
<i>Trans Conv2D (Stride=4)</i>	(1xd, kernel_size=6)	(1024x1x64)
<i>batchNormalization</i>		
<i>ReLU</i>		
<i>Trans Conv2D (Stride=4)</i>	(1, kernel_size=6)	(4096x1x1)
<i>batchNormalization</i>		
<i>Tanh</i>		

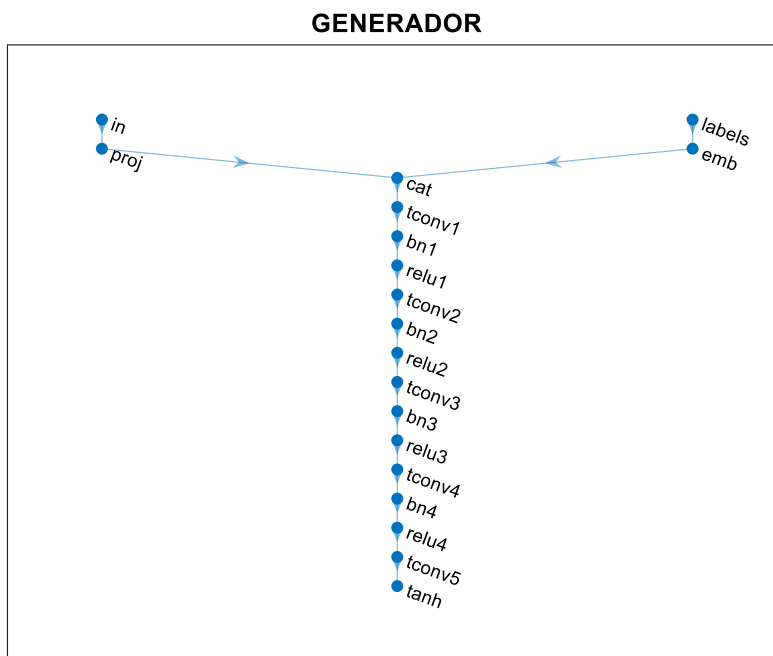
Nota. Las señales en tiempo del volcán Llaima llegan a tener 6000 muestras y en el trabajo de (Galarza & Vega, 2022) trabajan con 8192 muestras, que es también utilizado en el presente trabajo, por tanto, la arquitectura del generador es la misma.

Por otro lado, se tiene que, en la entrada las etiquetas junto con el vector ruido remodelado, ingresan al generador. Antes de esto se debe considerar que la red CGAN trabaja con imágenes, por lo tanto, se deben convertir en imágenes con la función *imageInputLayer*. Al

generador solo ingresa una capa, es decir, las entradas deben ser concatenadas como se ve en la Figura 11.

Figura 11

Red generadora de la CGAN



Una serie de capas, que están agrupadas en cinco bloques, se usan en la red generadora. Cada uno de estos contiene una convolución transpuesta que se utiliza para aumentar el tamaño de la imagen lo que ayuda a mejorar la calidad de las imágenes generadas. En la convolución transpuesta se consideran dos conceptos importantes, el *padding* y el *stride*. El *padding* es adicionar valores de relleno en los bordes de las imágenes y sirve para controlar el tamaño de la salida y no afectar la información de los bordes por el tamaño del *kernel*. El *stride* determina cuántos píxeles se mueve el *kernel* en cada iteración y puede afectar el tamaño de la salida reduciéndolo.

Otra parte que constituye estos cinco bloques es la capa de *Batch Normalization* que sirve para normalizar las activaciones y para realizarlo se lo hace a través de la resta de la

media y división de la desviación estándar. Además, ayuda a acelerar el entrenamiento ya que puede permitir tasas de aprendizaje altas, mejora la estabilidad de la red para una convergencia rápida y previene el sobreajuste.

Por último, para los cinco bloques se utiliza la función de activación ReLU con excepción del último bloque en el cual se utiliza la función de activación *tanh*. Las funciones de activación se utilizan para evitar el desvanecimiento del gradiente. Este es un problema debido a que los gradientes se vuelven pequeños a medida que se propagan hacia capas anteriores de la red durante la retropropagación. Si los gradientes se vuelven pequeños tiene como consecuencia que la red no aprenda de manera eficiente. La función de activación ReLU devuelve el valor de entrada si es positivo y cero en caso contrario. La función *tanh* asegura que los valores generados estén dentro de un rango específico, por ejemplo, al generar imágenes, los valores de píxeles están en el rango de 0 a 255 o -1 a 1.

En la función de *transposedConv2dLayer* se necesitan valores como el tamaño del filtro que multiplica a su tamaño d de 64. Para el primer bloque de los cinco mencionados anteriormente, se tiene un valor máximo de 512. Al avanzar entre capas, este valor disminuye en potencia de dos ya que al final se obtiene la respuesta en frecuencia y no es necesario la identificación grande de características.

Para el valor del *kernel*, debido a la coincidencia en los puntos FFT tomados para el volcán Cotopaxi en (Galarza & Vega, 2022) que son 8192 y que se toma la mitad del este valor 4096, son los mismos para el caso del volcán Llaima. Estos dependen del ancho final W_o , que es el ancho que se obtiene en cada capa de las cinco, W_i que es el ancho inicial, s es el *stride* y p el *padding* como se muestra en la Tabla 3.

Tabla 3*Tamaño del kernel para el generador*

Bloque	W_i	W_o	s	Tamaño del Kernel
1	4096	1024	4	6
2	1024	256	4	6
3	256	64	4	6
4	64	16	4	6
5	16	4	4	6
6	4	1	1	4

Nota. La tabla es tomada de (Galarza & Vega, 2022) debido a que el ancho inicial correspondiente a $W_i = 4096$ coincide entre las señales de los volcanes Llaima y Cotopaxi.

Discriminador

Como anteriormente se mencionó, la tabla del generador es tomada de un trabajo hecho para el volcán Cotopaxi. De igual manera se realiza para el discriminador que se puede ver en la Tabla 4 en donde d es el tamaño del filtro igual a 64.

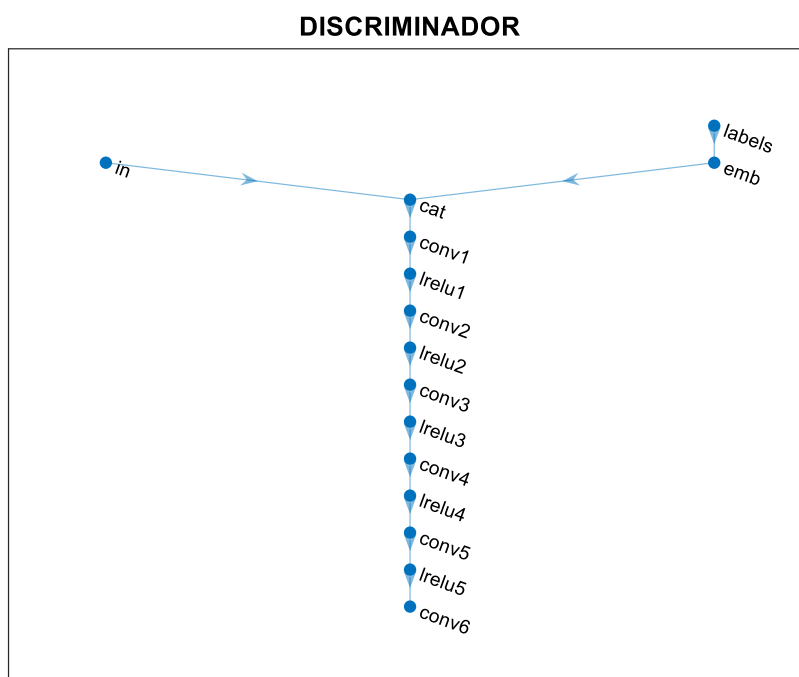
Tabla 4*Arquitectura del discriminador*

Operación	Kernel Size	Salida
<i>Entrada</i> \times $G(z)$	(4096x1x1)	(4096x1x2)
<i>Etiquetas</i>	(1x1x1)	
<i>Conv1D</i> (Stride=4)	(d , kernel_size=6)	(1024x1x64)
<i>LReLU</i> ($\alpha=0.2$)		
<i>Conv1D</i> (Stride=4)	(2xd, kernel_size=6)	(256x1x128)
<i>LReLU</i> ($\alpha=0.2$)		
<i>Conv1D</i> (Stride=4)	(4xd, kernel_size=6)	(64x1x256)
<i>LReLU</i> ($\alpha=0.2$)		
<i>Conv1D</i> (Stride=4)	(8xd, kernel_size=6)	(16x1x512)
<i>LReLU</i> ($\alpha=0.2$)		
<i>Conv1D</i> (Stride=4)	(16xd, kernel_size=6)	(4x1x1024)
<i>LReLU</i> ($\alpha=0.2$)		
<i>Conv1D</i> (Stride=1)	(1, kernel_size=4)	(1x1x1)

A las entradas se ingresa el vector de magnitudes en frecuencia y las etiquetas que fueron separadas por eventos en el preprocesamiento. Para ingresar a la CGAN, se necesita que las magnitudes se conviertan en imágenes y esto se logra con la función *imageInputLayer*. La concatenación de estas imágenes y de las etiquetas se observa en la Figura 12.

Figura 12

Red discriminadora de la CGAN



A diferencia del generador, el discriminador posee seis bloques en los que cada uno consta de una capa de convolución, seguido de la función de activación Leaky ReLU (LReLU). El uso de esta función de activación específica se debe a la recomendación de (Radford, 2016). Dadas las características de LReLU, en la cual permite que una pequeña fracción negativa de la información fluya a través de la función, es decir, que permite un pequeño gradiente distinto de cero para valores menores a cero (Maas, Hannun, & Ng, 2013). Con esto se soluciona el problema de “dying ReLU problem” en el cual muchas neuronas tienen salida cero, se las

considera como muertas ya que no contribuyen al aprendizaje y se corre el riesgo de que el discriminador no distinga entre muestras reales y generadas.

La primera capa del discriminador tiene una entrada de tamaño de la magnitud frecuencial de las señales que corresponde 4096. Los pesos en el discriminador deben ser inicializados. Por tanto, en cada capa convolucional, se los establece con pesos aleatorios que siguen una distribución gaussiana. Esta práctica previene el estancamiento durante el entrenamiento y aborda un problema posterior en la generación de señales sintéticas, evita que todas sean similares entre sí. Además, en cada una de las capas convolucionales el tamaño del filtro se incrementa en potencias de dos hasta llegar a la última capa en donde el tamaño del filtro corresponde a uno. El valor inicial del filtro es de 64. Para el tamaño del *kernel* de cada una de las capas convolucionales se utiliza (Galarza & Vega, 2022) y los valores se pueden resumir en la Tabla 5. Por último, se inicializa la función de activación LReLU con 0.2 (Radford, 2016).

Tabla 5

Tamaño del kernel para el discriminador

Bloque	W_i	W_o	s	Tamaño del kernel
1	4096	1024	4	6
2	1024	256	4	6
3	256	64	4	6
4	64	16	4	6
5	16	4	4	6
6	4	1	1	4

Nota. La tabla es tomada de (Galarza & Vega, 2022) debido a que el ancho inicial correspondiente a $W_i = 4096$ coincide entre las señales de la base de datos de los volcanes Llaima y Cotopaxi.

Entrenamiento

El entrenamiento del modelo CGAN se hace a través de ciclos, en donde se recorren los datos de entrenamiento y los parámetros de la red se actualizan en cada iteración. El número de iteraciones dependen del número de épocas (*epoch*) y del tamaño de los mini lotes (*minibatch size*) El número de épocas corresponden a un ciclo completo en el conjunto de datos de entrenamiento y se actualizan los pesos en función de los errores. El *minibatch size* corresponde a la cantidad de ejemplos de entrenamiento que se procesan simultáneamente antes de realizar una actualización de los pesos.

Si el discriminador es capaz de distinguir rápidamente entre las respuestas en frecuencia reales y las generadas, puede llevar a problemas en el entrenamiento del generador. Para lograr un mejor equilibrio en el aprendizaje del discriminador y el generador se invierte aleatoriamente las etiquetas de algunas de las magnitudes en frecuencia reales, es decir que, para cada época, se mezclan los datos de entrenamiento y se repite el *minibatch size*. Por cada *minibatch*, se usa la función adicional *trainGAN* obtenida de (MathWorks, 2023), la cual genera un objeto *dIarray* que contiene una matriz de valores aleatorios para la red generadora. Luego se evalúan los gradientes del modelo con *dIfeval* y la función auxiliar *modelGradients*. Los parámetros de la red se actualizan por medio de la función *adamupdate*.

La función auxiliar *modelGradients* recibe como entrada las redes del generador y el discriminador, un *minibatch* de datos de entrada y una matriz de valores aleatorios. Su objetivo es calcular los gradientes de pérdida, tanto del generador como del discriminador, en relación a los parámetros que se pueden aprender, así como las puntuaciones de ambas redes. La función de pérdida utilizada se define en la función auxiliar *ganLoss*. Por medio de las pérdidas, se calculan los gradientes.

Los parámetros, tanto del discriminador como del generador, se actualizan en cada iteración al usar el optimizador de *Adam*. El optimizador de *Adam* es un algoritmo basado en

gradientes de primer orden de funciones estocásticas. Este es sencillo de implementar, computacionalmente eficiente, tiene pocos requisitos de memoria, es invariable al cambio de escala diagonal de los gradientes y es muy adecuado para grandes datos. Se utilizan los parámetros recomendados por los creadores, que incluyen la tasa de aprendizaje, así como los coeficientes beta uno y dos. Estos coeficientes se establecen en valores que se han determinado como los más efectivos para el optimizador de Adam. Se utiliza la tasa de aprendizaje $\alpha = 0.001$, el término $\beta_1 = 0.5$ y $\beta_2 = 0.999$ (Kingma & Ba, 2015).

Para supervisar el progreso de entrenamiento, se utiliza parte del *Toolbox Deep Learning* de MATLAB®, donde se ve la competencia entre el generador y el discriminador. Es posible examinar las imágenes visualmente o utilizar el gráfico de puntuación para identificar posibles problemas. En ciertas situaciones, el gráfico de puntuación puede indicar que no es recomendable continuar con el entrenamiento y es necesario detenerlo, ya que se ha alcanzado un estado de fallo del cual el proceso de entrenamiento no puede recuperarse.

El gráfico de puntuación utiliza una escala de cero a uno. Durante el entrenamiento, las dos redes deben converger a 0.5, pero este es un caso ideal y para que la red sea exitosa no necesariamente debe converger en ese valor, si no que puede rondar cerca de este. Las imágenes y la variación del vector de ruido son visualizados hasta que, al converger, el ruido se ajuste a la forma que tienen las magnitudes frecuenciales de los eventos.

La puntuación del discriminador depende de la probabilidad que las imágenes de entrada sean clasificadas en la clase correcta. En otras palabras, depende de las medias de la probabilidad de que una imagen sea real y las medias de la probabilidad de que las imágenes de entrada pertenezcan a la clase correcta. Esta puntuación se representa con la resta de 1 menos las probabilidades de que una imagen pertenezca a la clase generada. La puntuación del generador es la media de las probabilidades correspondientes a la salida del discriminador para las imágenes generadas.

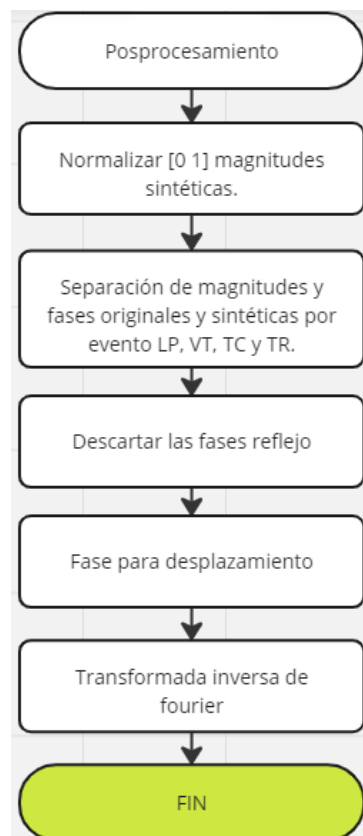
Mediante la técnica de prueba y error, se variaron los parámetros de *epoch* y *minibatch* size. Se mantuvo fijo tanto el número de capas de las redes generadora y discriminadora, el tamaño de la entrada al generador. Además, se mantuvo invariable el número de filtros. Sin embargo, es importante considerar que, al usar un número muy alto de filtros, esto aumenta la carga computacional, ya que más filtros en el generador permiten extraer más características, pero también puede resultar en un tiempo de ejecución más prolongado. Por lo tanto, se debe elegir un valor que logre un equilibrio adecuado.

Procesamiento Final

Al tener las señales sintéticas generadas de los cuatro eventos y sus respectivas etiquetas, se debe proceder a una etapa de posprocesamiento que se observa en la Figura 13.

Figura 13

Diagrama de bloques del posprocesamiento



Las magnitudes sintéticas m_{SS_M} se normalizan con el método *min-max* de 0 a 1 ya que la salida del modelo CGAN, por la naturaleza del mismo, está desde -1 a 1. Para la aplicación de este proceso se tiene el operador matemático $z_M\{m_{SS_M}\}$ y la matriz resultante de aplicar la normalización a cada magnitud sintética se observa a continuación:

$$Z = \{z_1, z_2, z_3, \dots, z_M\} \quad (12)$$

Además, para la reconstrucción de la señal, se debe tener tanto las magnitudes como las fases, por lo tanto, al tener un solo vector de magnitudes generado de manera sintética sms , este se debe separar en los distintos eventos microsísmicos. A continuación, las fases, con las cuales el resultado de la reconstrucción temporal mejora, se seleccionan con la ayuda de un criterio que se trata en el siguiente capítulo. Además, se construye una fase sintética que se puede encontrar en la ecuación (24) que permite el desplazamiento de la señal, esto corresponde a aplicar $P_{desplazamiento_{sint}}\{XB_M\}$. Finalmente, se aplica la IFFT para que las magnitudes frecuenciales sintéticas y las fases originales pasen al dominio del tiempo. La IFFT if se puede representar como el operador $if_M\{sms_M, ps_M\}$ en donde las magnitudes sintéticas corresponden a sms y las fases originales a ps . La matriz resultante de esta operación es la señal en el tiempo sintética xb , la cantidad de señales a generar B la elige el usuario. La matriz para señales sintéticas en el dominio del tiempo se la puede observar en la ecuación (12)

$$IF = \{if_1, if_2, if_3, \dots, if_B\} \quad (12)$$

$$P_{desplazamiento_{sint}} = \{xb_1, xb_2, xb_3, \dots, xb_B\} \quad (13)$$

Matriz de confusión

Para la validación de las señales sintéticas generadas, se ha optado por usar clasificadores de trabajos que se realizan a la par de este, y para ello se utiliza una herramienta conocida como matriz de confusión. Esta matriz muestra la cantidad de aciertos y errores cometidos por un modelo de clasificación en función de las clases que trata de predecir. Un ejemplo de matriz de confusión se la puede observar en la Tabla 6. Gracias a esta herramienta se pueden obtener métricas como son la Exactitud (A , del inglés *Accuracy*), la Precisión (P), la Especificidad (S , del inglés *Specificity*), la Sensibilidad (R , del inglés *Recall*) y la Tasa de Error Balanceada (BER del inglés, *Balance Error Rate*).

Tabla 6

Matriz de confusión para cuatro clases

		Clase Predicha			
		1	2	3	4
Clase Real	1	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)
	2	Falso Negativo (FN)	Verdadero Positivo (VP)	Falso Negativo (FN)	Falso Negativo (FN)
	3	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)
	4	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)

Nota. La tabla muestra un ejemplo para la segunda clase de cómo se cambian los valores alrededor del verdadero positivo, este proceso se lo debe hacer para cada una de las clases.

La matriz de confusión se organiza en una tabla con filas y columnas que representan las clases reales y las clases predichas. Una matriz de confusión puede contener diferentes valores, como son los Verdaderos Positivos (VP), Falsos Positivos (FP), Verdaderos Negativos (VN) y Falsos Negativos (FN). Los VP es el número de resultados que realmente pertenecen a

esa clase y se predicen correctamente como tales. Los FN es el número de resultados que pertenecen a una clase específica pero que se predicen incorrectamente. Los FP es el número de resultados que no pertenecen a una clase pero que se predicen incorrectamente como esa clase. Los VN corresponden a los resultados que no pertenecen a una clase específica y se predicen correctamente como no pertenecientes a esa clase.

Con estos valores de la matriz de confusión se calculan las métricas de A, P, R, S y el BER. La A se relaciona con qué tan cercano está el resultado de una medición al valor verdadero. Si este valor aumenta, también lo hace la certeza de que la clasificación fue exacta. La P , por su parte, cuantifica la dispersión de los valores en un conjunto, lo cual se verifica por la frecuencia con la que el modelo acierta. La R incluye la proporción de casos positivos que se identificaron correctamente, mientras que la S se refiere a la capacidad del algoritmo para clasificar correctamente los casos negativos. Por último, el BER evalúa el número total de predicciones incorrectas en la prueba (Galarza & Vega, 2022). Cada una de estas métricas de rendimiento se calcula mediante las siguientes ecuaciones:

$$A(\%) = \frac{VP + VN}{VN + VP + FN + FP} \times 100 \quad (14)$$

$$P(\%) = \frac{VP}{VP + FP} \times 100 \quad (15)$$

$$R(\%) = \frac{VP}{VP + FN} \times 100 \quad (16)$$

$$S(\%) = \frac{VN}{VN + FP} \times 100 \quad (17)$$

$$BER = 1 - \frac{R + S}{200} \quad (18)$$

Aplicación

Con el uso de *App Designer* de MATLAB®, se ha desarrollado una aplicación, que se puede observar en la Figura 14, la cual permite generar señales sintéticas en el dominio de tiempo en donde se realiza una división en la que el total de señales a generar se reparte en cuatro partes iguales, esto asegura que cada evento genere la misma cantidad de señales.

Figura 14

Pantalla principal de la interfaz gráfica



El generador de señales tiene varias funciones, como se puede comprobar en la Figura 15. Las funciones incluyen ver la información de cuantas señales se van a generar, se puede escribir el número de señales totales a generar, el nombre de cómo se llama la tabla que se genera y en que directorio se guarda. Una vez generadas las señales, la aplicación permite ver un ejemplo de cada uno de los eventos donde se muestra la señal en el dominio del tiempo en la cual la amplitud es igual a las señales originales del Llaima, pero en tiempo está con 8.192

segundos. La densidad espectral de potencia se obtiene como una estimación por medio del método de *Welch* con 512 muestras para resolución media. La PSD puede revelar características que permiten la detección de patrones específicos que diferencian cada evento.

Figura 15

Generador de señales con el uso de CGAN

The image shows a web application interface for generating synthetic signals. At the top, there is a navigation bar with 'Inicio', 'CGAN', and a '+' icon. The main title is 'SISTEMA GENERADOR DE SEÑALES SINTÉTICAS'. Below the title is a '+ Información' button. The interface contains several input fields and buttons: 'Cantidad de señales sintéticas a generar' with the value '12', 'Nombre del archivo a generar' with the value 'Tabla_CGAN', and a 'Directorio donde se guardarán las señales' field with a 'Seleccionar' button. There are also 'Generar' and 'Graficar Señal + PSD' buttons. At the bottom, there is a 'Status' field.

Inicio CGAN +

SISTEMA GENERADOR DE SEÑALES SINTÉTICAS

+ Información

Cantidad de señales sintéticas a generar

Nombre del archivo a generar

Directorio donde se guardarán las señales

Status

Capítulo III: Análisis de Resultados

En este capítulo se presentan los resultados del proceso utilizado para seleccionar las fases adecuadas de cada uno de los eventos. Además de análisis de los distintos experimentos realizados para llegar a la red definitiva, las fallas que puede presentar la CGAN, la variación de distintos parámetros para encontrar finalmente los que proporcionen los mejores resultados. Con esto se muestran los gráficos obtenidos en la frecuencia, en el dominio del tiempo y las PSD de los distintos eventos. Como paso final, se va a mostrar la evaluación de software que incluye un clasificador de microsismos desarrollados por el Grupo de Investigación en Sistemas Inteligentes (WiCom-Energy) y el Centro de Investigación de Redes Ad Hoc (CIRAD) implementado con la ayuda de *autoencoders*.

Filtrado de Fases

Al reconstruir la señal, se debe considerar tanto las fases originales como las magnitudes sintéticas. Sin embargo, al analizar las fases de cada señal, se observa que cada evento presenta fases que son más lineales y otras que se desvían en diferentes direcciones, como se puede apreciar en la Figura 16 y la Figura 17. Por lo tanto, se aplica el criterio presentado en la Figura 18 y Figura 19. En estas figuras se muestra cómo al utilizar una fase lineal, la reconstrucción en el tiempo resulta con menos ruido en comparación con la señal reconstruida al usar una fase reflejo.

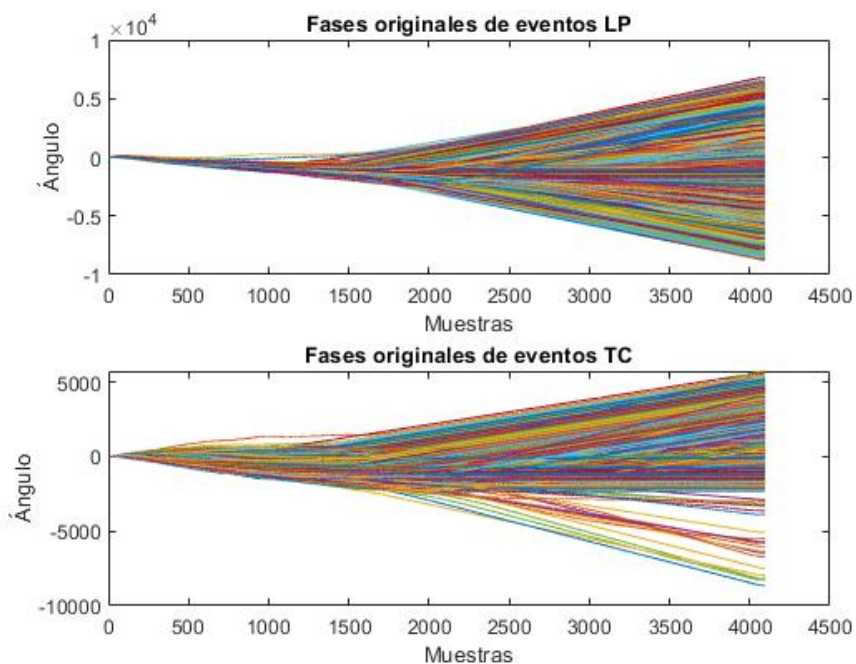
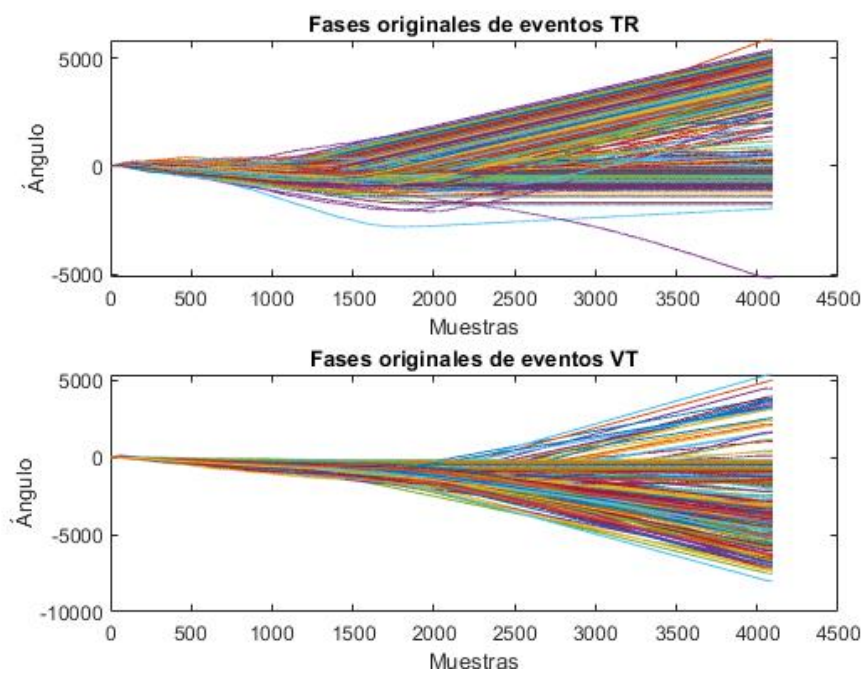
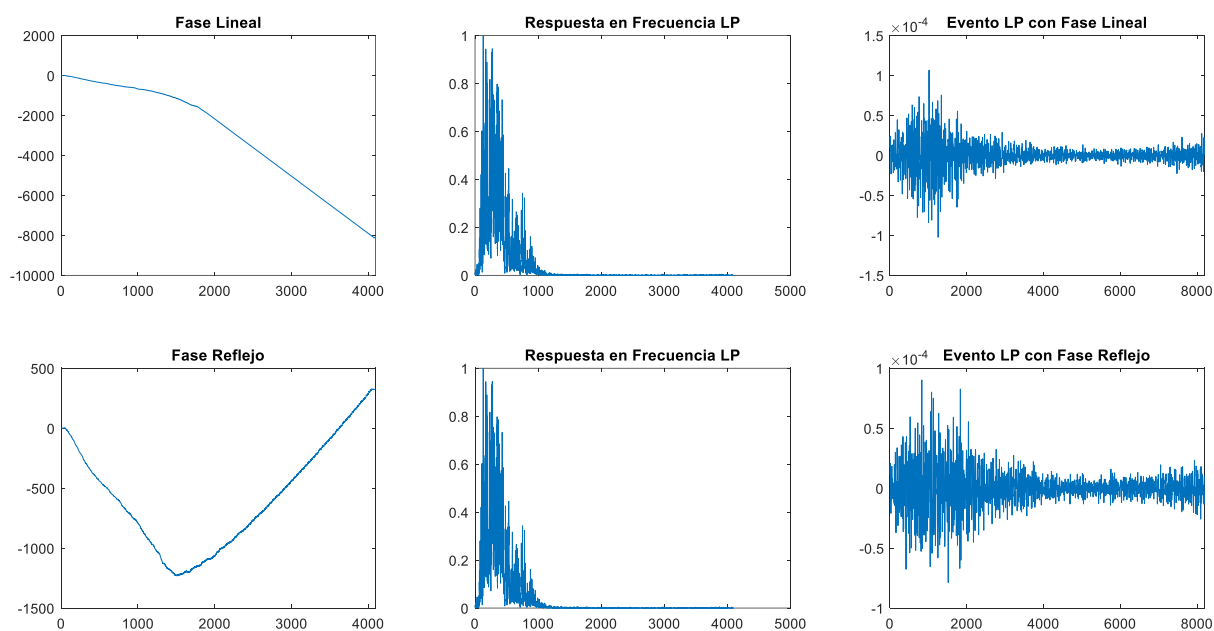
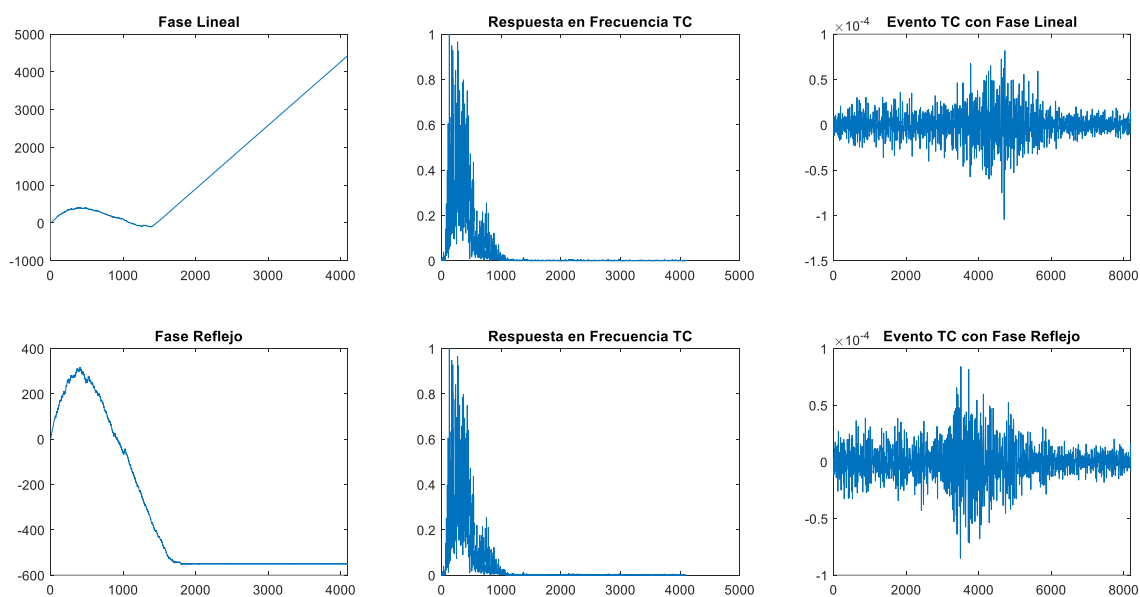
Figura 16*Fases totales de los eventos LP y TC***Figura 17***Fases totales de los eventos TR y VT*

Figura 18

Ejemplo de criterio en señales LP para selección de fases

**Figura 19**

Ejemplo de criterio en señales TC para selección de fases



Para que las fases reflejo sean eliminadas, se realiza un análisis estadístico. En el vector de fases, con el uso de los últimos valores de las muestras, se calcula la media y, al considerar el evento, lo que se busca es que no haya valores superiores o inferiores a la media obtenida. Finalmente se eliminan esos valores y se consiguen las fases más lineales. Las medias calculadas de los cuatro eventos se ven a continuación:

$$\overline{\overline{x}}_{LP} = \frac{\sum_{i=1}^{N_{LP}} X_{i_{LP}}}{N_{LP}} = \frac{\sum_{i=1}^{N_{LP}} X_{i_{LP}}}{1310} = -2723 \quad (19)$$

$$\overline{\overline{x}}_{TC} = \frac{\sum_{i=1}^{N_{TC}} X_{i_{TC}}}{N_{TC}} = \frac{\sum_{i=1}^{N_{TC}} X_{i_{TC}}}{1488} = -1962 \quad (20)$$

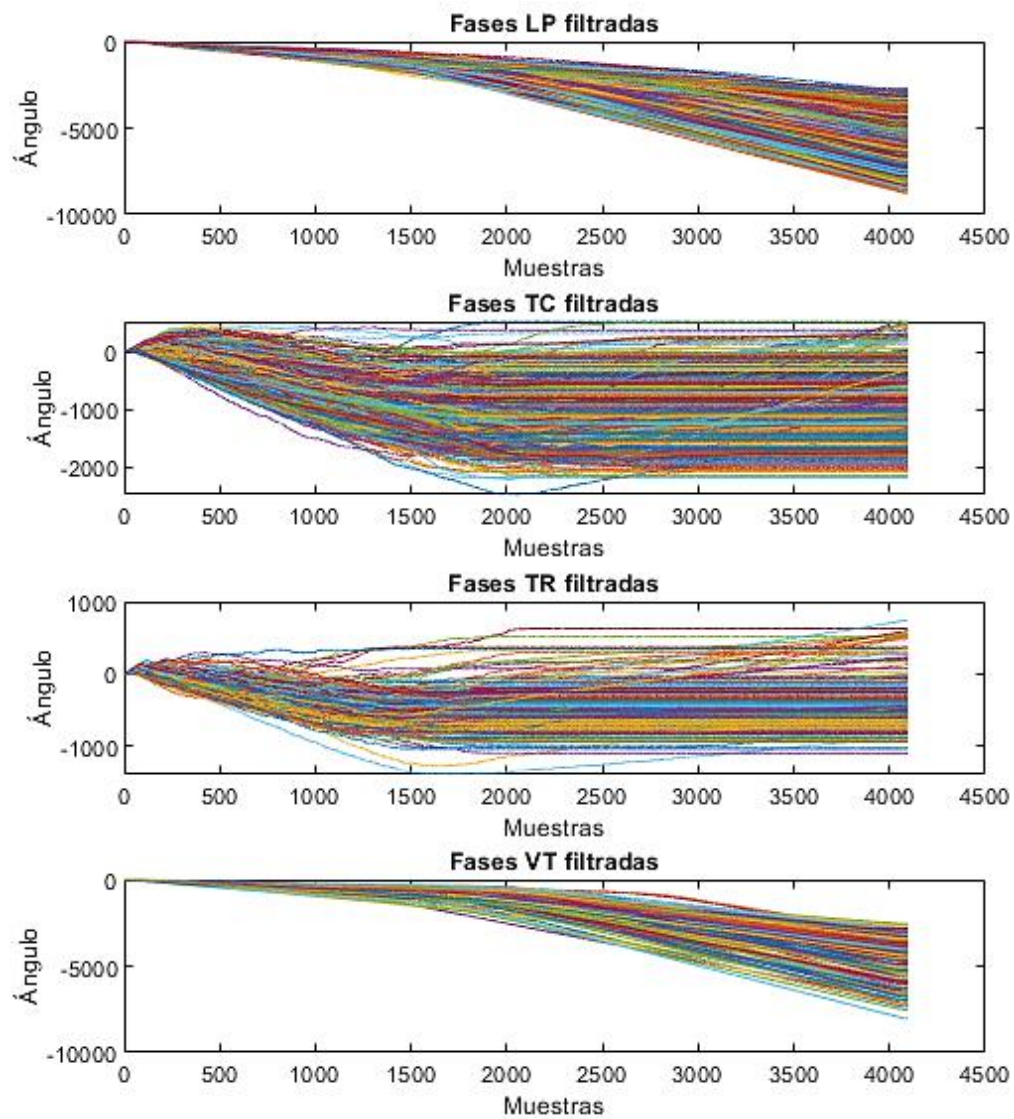
$$\overline{\overline{x}}_{TR} = \frac{\sum_{i=1}^{N_{TR}} X_{i_{TR}}}{N_{TR}} = \frac{\sum_{i=1}^{N_{TR}} X_{i_{TR}}}{490} = -1787 \quad (21)$$

$$\overline{\overline{x}}_{VT} = \frac{\sum_{i=1}^{N_{VT}} X_{i_{VT}}}{N_{VT}} = \frac{\sum_{i=1}^{N_{VT}} X_{i_{VT}}}{304} = VT = -2455 \quad (22)$$

Con los valores calculados se considera que para las señales LP y VT, las fases seleccionadas son aquellas que se ubican por debajo de su valor medio. En cambio, para los eventos TC y TR, las fases utilizadas son las que están por encima de su valor medio. El resultado de las fases filtradas se visualiza en la Figura 20.

Figura 20

Fases filtradas



Un problema que se observa es que, al instante de reconstruir la señal temporal al usar las magnitudes sintéticas y las fases reales, existe un desplazamiento de la señal, como se observa en la Figura 21(a), en donde, parte de la señal del evento está al final. Este fenómeno se conoce como “desplazamiento de fase” y ocurre cuando se aplica FFT en la señal original $x(t)$ esto trae una modificación en $X(\omega)$. La fase $\phi(\omega)$ en $X(\omega)$ está relacionada con el desplazamiento en el tiempo τ de la forma $\phi(\omega) = -\omega\tau$. Por lo tanto, después de la FFT, las

componentes de frecuencia en el dominio de la frecuencia se expresarían como

$X'(\omega) = |X(\omega)| \cdot e^{-j(\omega\tau)}$. Cuando se aplica la IFFT y pasa al dominio del tiempo la señal queda desplazada en el tiempo $x'(t) = x(t - \tau)$.

Para corregir este fenómeno, se ha optado por crear una fase sintética que ayude a las señales de todos los eventos, desplazarse lo suficiente para que la señal no esté cortada, se tiene como resultado lo que se muestra en la Figura 21(b). La fase sintética ha sido la creada a continuación:

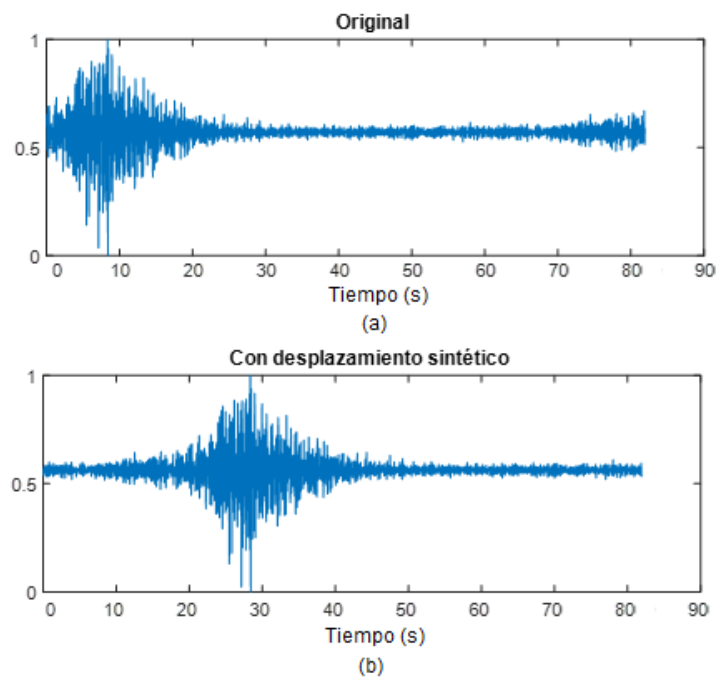
$$y1 = \text{linspace}(0, 2 \times \pi, 4096) \quad (23)$$

$$P_{\text{desplazamiento}_{\text{sint}}} = -y1 \times 1000 \quad (24)$$

Con el desplazamiento, las fases originales y las señales sintéticas, se realiza la IFFT.

Figura 21

Señal origina y señal desplazada



Nota. La Figura representa el proceso de desplazamiento: (a) Señal reconstruida en el dominio del tiempo sin ningún proceso. (b) Señal reconstruida desplazada en el dominio del tiempo.

Parámetros para el modelo CGAN ideal

Para probar la falla que puede conllevar el uso de demasiados filtros, se tiene el entrenamiento con 128 filtros tanto para el generador como para el discriminador, debido a que, si se usa mayor cantidad de filtros como, por ejemplo, para el valor de 256, el entrenamiento no empieza y el equipo deja de responder, por lo tanto, también se consideró hacer el entrenamiento únicamente con 50 épocas y 25 de valor para *minibatch*, con un total de 7150 iteraciones. Los resultados del entrenamiento se tienen en la Figura 22, en donde se aprecia que la presencia de demasiados filtros causa que la red tenga un error de convergencia, ya que, el avance del entrenamiento de la red generadora y discriminadora se alejan del valor de convergencia, además el recurso computacional es demasiado alto y hace que los equipos sufran de sobreesfuerzo. Las magnitudes frecuenciales de los distintos eventos del modelo colapsado con exceso de filtros se pueden ver en la Figura 23.

Figura 22

Modo colapso con exceso de filtros

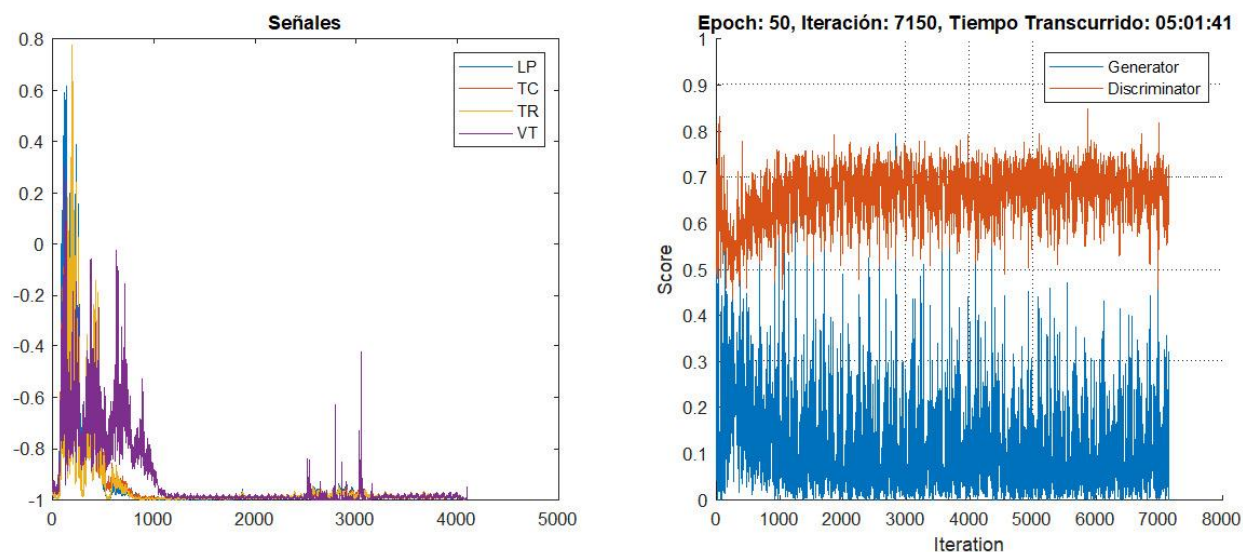
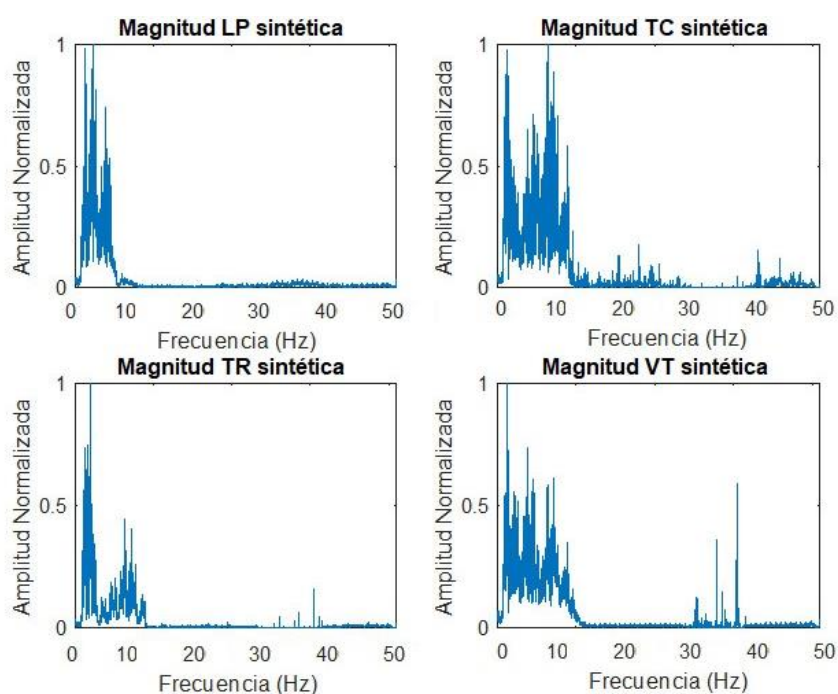


Figura 23

Respuesta en frecuencia de los cuatro eventos generados con un modelo colapsado



Por lo tanto, se fija el número de filtros y el número de capas convolucionales, pero se varía el número de *epoch* y el tamaño del *minibatch*. Para el entrenamiento se usaron todas las señales microsísmicas. Para identificar el número de épocas se toma en cuenta (Karras, y otros, 2020) en donde el número de épocas se sugiere que esté cerca del total de las señales. Al probar con 3000 épocas se decide solo realizar los entrenamientos de 1000, sin embargo, incluso este valor de épocas es demasiado alto y la red tiende a entrenarse de manera errada como se observa en la Figura 24. Al tener en cuenta el entrenamiento hecho con 1000 épocas, se observó que el entrenamiento de la red es correcto hasta antes de 100 épocas. Por lo tanto, los entrenamientos posteriores se realizan en base a 90 épocas ya que no tiene sentido continuar el entrenamiento si se logra el equilibrio entre el generador y el discriminador, es decir, ya se tiene la convergencia. Como punto inicial se toma 1000 épocas y luego este valor reduce considerablemente a 90, los resultados del entrenamiento se observan en la Tabla 7.

Tabla 7

Parámetros variantes para pruebas en el modelo CGAN

Epoch	Iteraciones	Minibatch	Puntaje Generador	Puntaje de discriminador
1000	28000	128	0.16	0.6
90	2520	128	0.25	0.5
90	21510	15	0.15	0.65

Los resultados presentados en la tabla anterior indican que, al emplear un número excesivo de épocas, además de generar una carga computacional significativa, que se evidencia en la Figura 24 con un tiempo de entrenamiento de aproximadamente 20 horas, la red entrenada tiende a experimentar un error de convergencia. Aunque el proceso de entrenamiento progresa de manera óptima hasta cierto punto, alrededor de la iteración 4000, se observa una divergencia y los valores del generador y el discriminador se desvían de la convergencia óptima; esto se conoce como sobre entrenamiento. Los resultados correspondientes a las magnitudes de frecuencia generadas de forma sintética se encuentran ilustrados en la Figura 25.

Figura 24

Falla de convergencia con un epoch grande

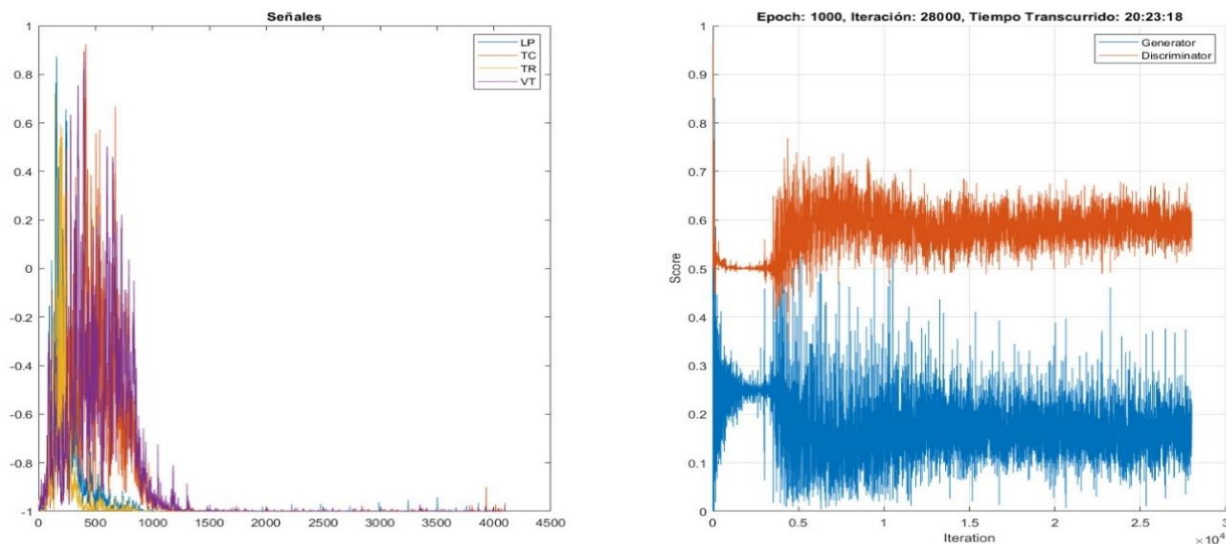
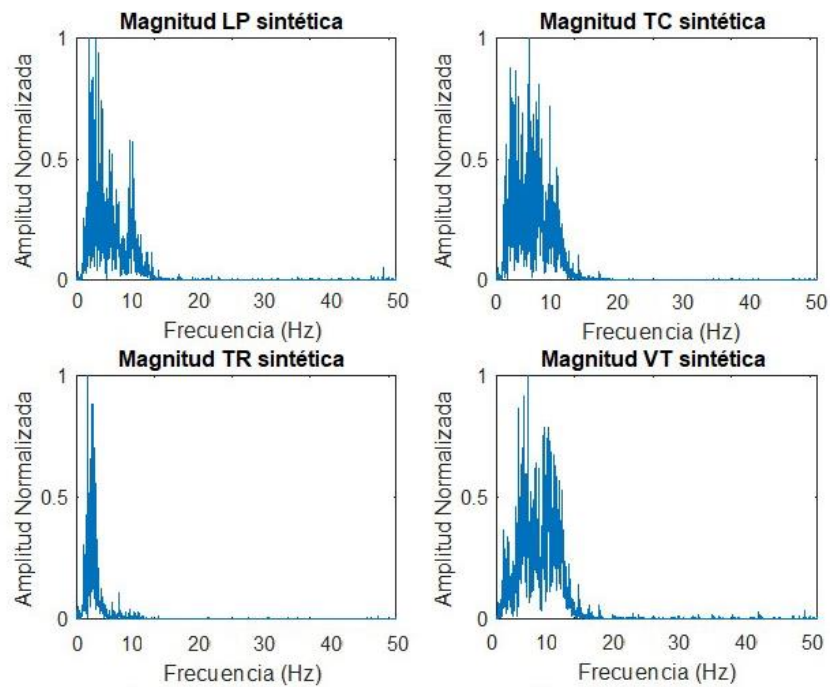


Figura 25

Respuesta en frecuencia de los cuatro eventos generados con demasiadas épocas



Se debe considerar que el uso de demasiadas épocas produce que el entrenamiento se aleje de la convergencia, por lo tanto, se toma como valor de *epoch* de 90, con este valor se prueba el *minibatch* mayor que el de las épocas con 128. Los resultados del entrenamiento se observan en la Figura 26 y las señales de magnitud frecuenciales se observan en la Figura 27.

Figura 26

Caso 1: 90 Epoch, 128 minibatch y 2520 iteraciones

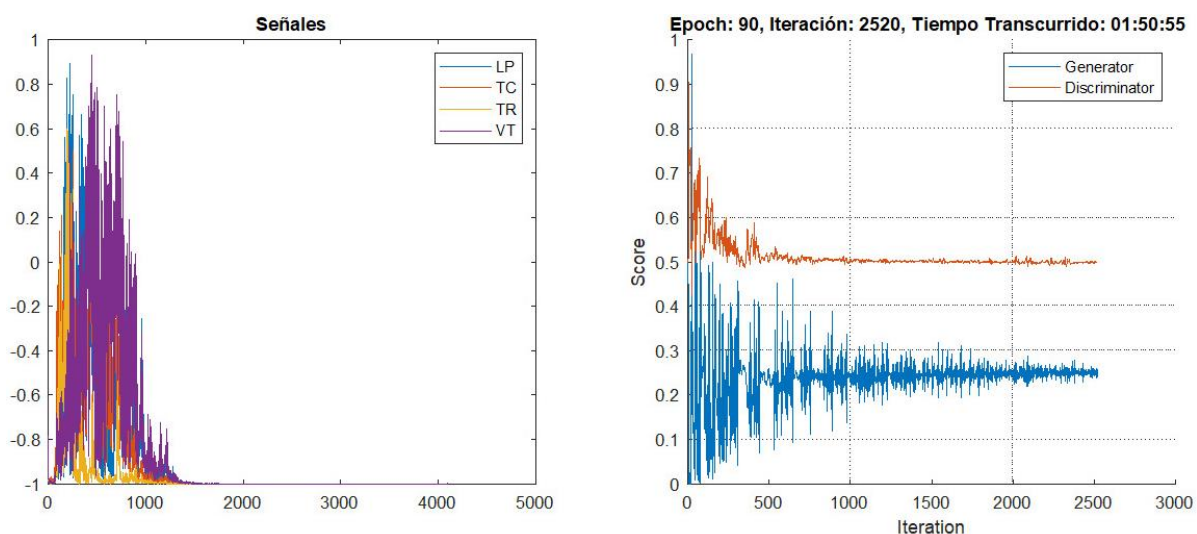
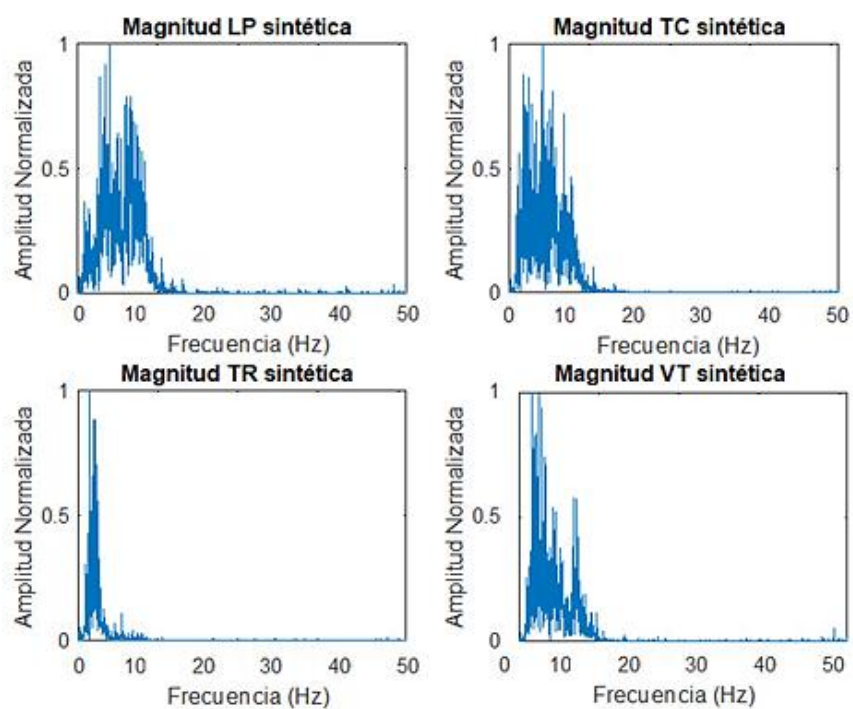


Figura 27

Respuesta en frecuencia generadas con el Caso 1



Al usar el valor de *epoch* de 90, se prueba el *minibatch* menor que el de las épocas con 15, es decir, el *minibatch* es muy pequeño. Los resultados del entrenamiento se observan en la Figura 28 y las señales de magnitud frecuenciales se observan en la Figura 29.

Figura 28

Caso 2: 90 Epoch, 15 minibatch y 21510 iteraciones

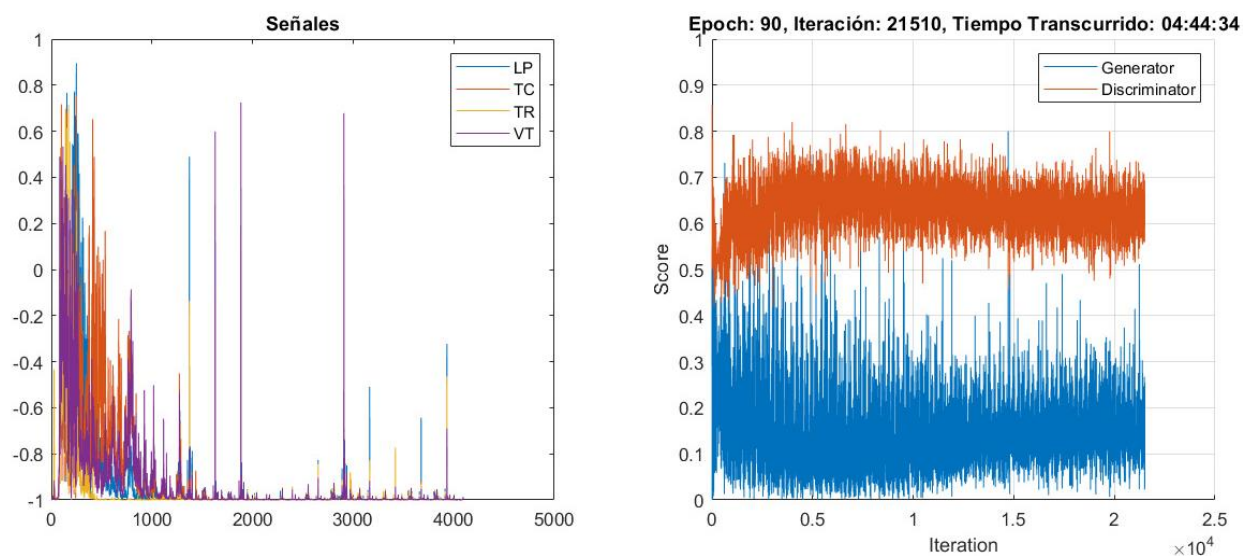
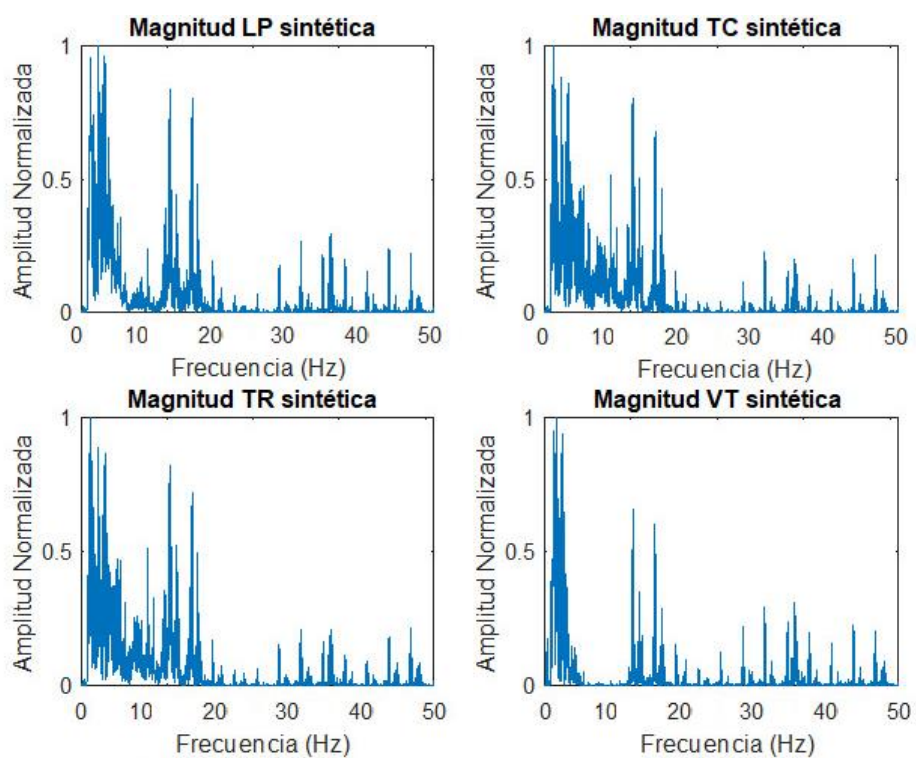


Figura 29

Respuesta en frecuencia generadas con el Caso 2

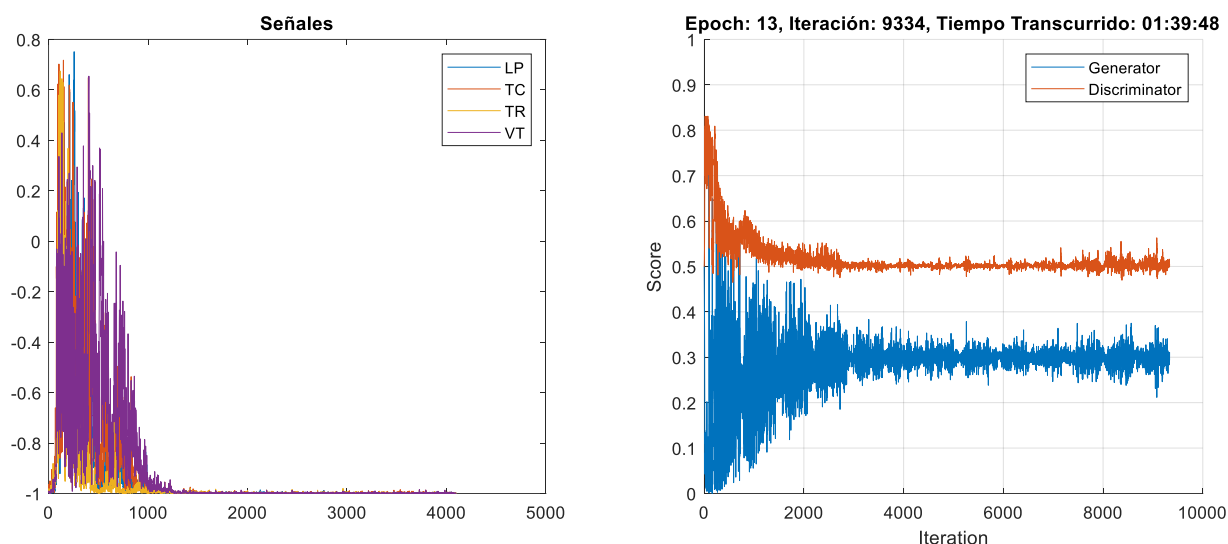


Dos casos se analizaron en donde se mantiene el número de épocas en 90 y se varía el valor de *minibatch size*. Se constató que un valor de tamaño de *minibatch* mayor al de las épocas resulta en un entrenamiento exitoso, pero resulta perjudicial cuando se utiliza un valor excesivamente bajo. Para el modelo que utiliza 128 de *minibatch* los resultados adecuados y la red tiene estabilidad; el puntaje alcanzado por el generador fue de 0.25, y el del discriminador de 0.5, además las señales generadas son mejores. En el caso 2 se observa que a pesar de todas las iteraciones que se realizan en el entrenamiento, no se llega a la convergencia, incluso la forma que toma la curva que permite monitorizar el entrenamiento del generador y del discriminador, es muy ruidosa.

Un entrenamiento adicional es realizado en donde se combinan los valores recomendados del optimizador de *Adam*, dado por sus creadores, y el parámetro que se usó en un ejemplo desarrollado por MATLAB® en (The MathWorks, 2023), donde el valor del parámetro $\alpha = 0.001$ es remplazado por el valor de $\alpha = 0.0002$. Este cambio tuvo como consecuencia que la red no necesita altos valores de épocas ni un tamaño grande de *minibatch*, por tanto, el recurso computacional se redujo de manera significativa. Se utilizaron los valores de 13 *epoch*, 5 en el *minibatch*, con un resultado de 9334 iteraciones, un puntaje de 0.5 en el discriminador y uno de 0.3 en el generador, con una diferencia entre ambas redes de 0.2. El tiempo de entrenamiento fue de 1 hora y 39 minutos y se lo puede observar en la Figura 30.

Figura 30

Entrenamiento con 13 épocas y 5 de minibatch size



Resultados del modelo CGAN final

Como mejores opciones, en donde se ha alcanzado la convergencia, se tiene el Caso 1 y el entrenamiento adicional. Ambos tienen características que favorecen al momento de considerarlos, la diferencia notable es que, en el entrenamiento adicional, el número de *epoch* es menor, solo se necesitan 13 comparadas con las 90 que son del caso 1. Otra ventaja es que la puntuación del generador es de 0.30, es decir, el generador engaña mejor al discriminador, comparado con el Caso 1 que llega a 0.25. Aunque ambos convergen, la curva de las redes es más suave para el Caso 1. El costo computacional para ambas es parecido, para el Caso 1 se utiliza 1 hora y 50 minutos y, para el caso del entrenamiento adicional se tiene 1 hora y 39 minutos. Al considerar todas las diferencias entre los dos modos de entrenamiento, el modelo final se lo puede observar con 13 épocas en la Figura 30, ya que, en el entrenamiento, las redes generadora y discriminadora se estabilizan antes, aproximadamente desde la iteración 6000, por tanto, el entrenamiento final para este modelo se realizó solo con 13 épocas.

Con el modelo CGAN originado, se obtienen las distintas magnitudes frecuenciales de los cuatro eventos. Para poder compararlos se pueden observar diferentes ejemplos de magnitudes reales. En la Figura 31 para eventos LP y se pueden comparar con las magnitudes sintéticas generadas que se observan en la Figura 32. En la Figura 33 las magnitudes frecuenciales reales para señales TC se puede comprar con la sintética en la Figura 34. En la Figura 35 para eventos reales TR se compara con las generadas en la Figura 36 y para eventos VT se observa en la Figura 37 las reales comparadas con sintéticas de la Figura 38.

Figura 31

Magnitudes frecuenciales de eventos LP reales

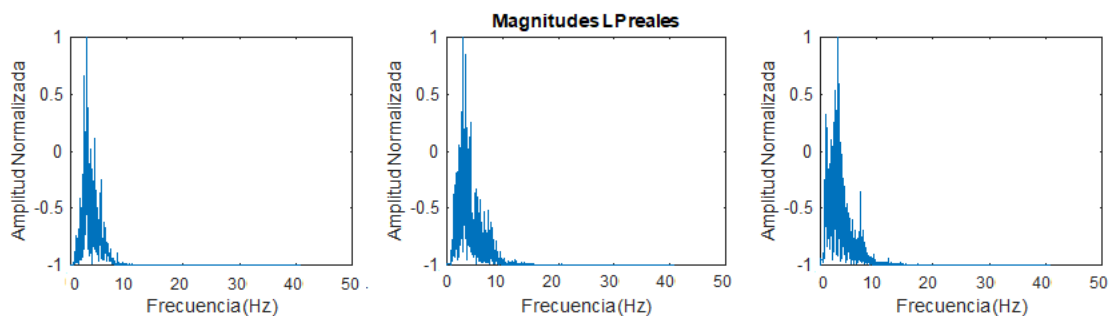


Figura 32

Magnitudes frecuenciales de eventos LP sintéticos

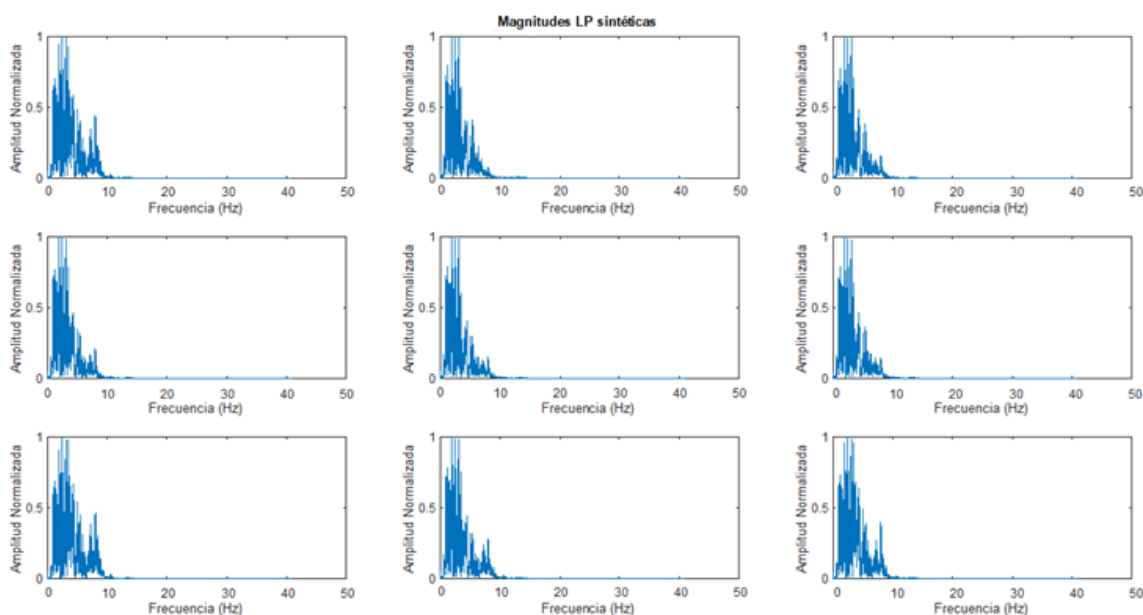
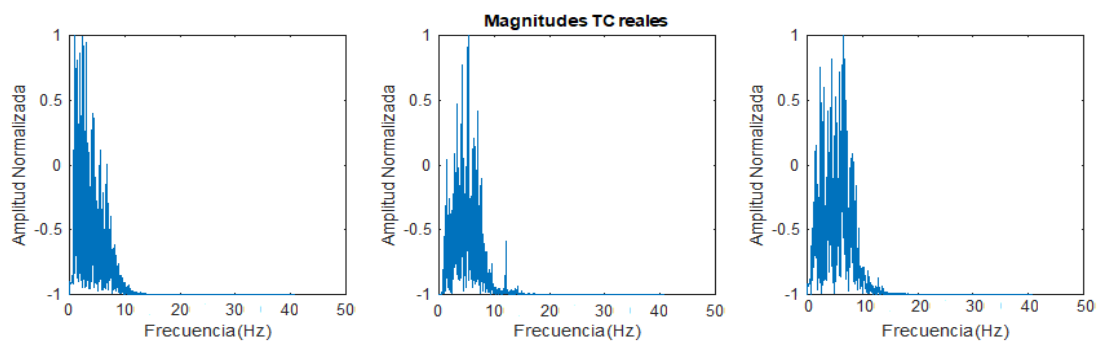


Figura 33

Magnitudes frecuenciales de eventos TC reales

**Figura 34**

Magnitudes frecuenciales de eventos TC sintéticos

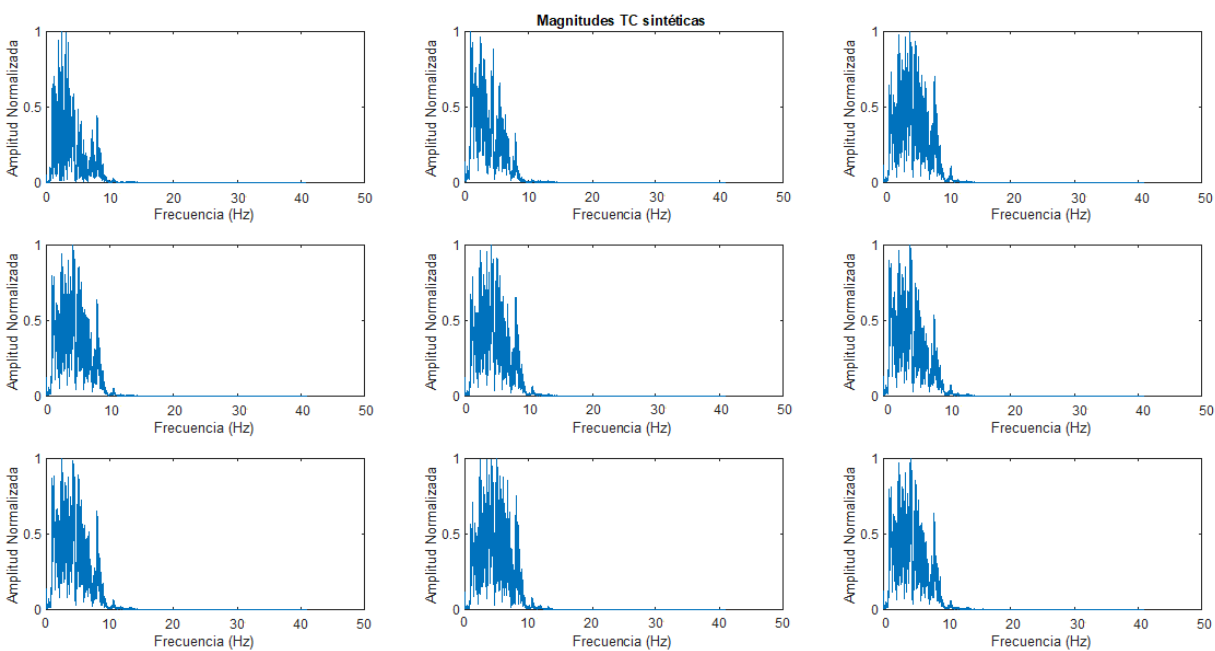


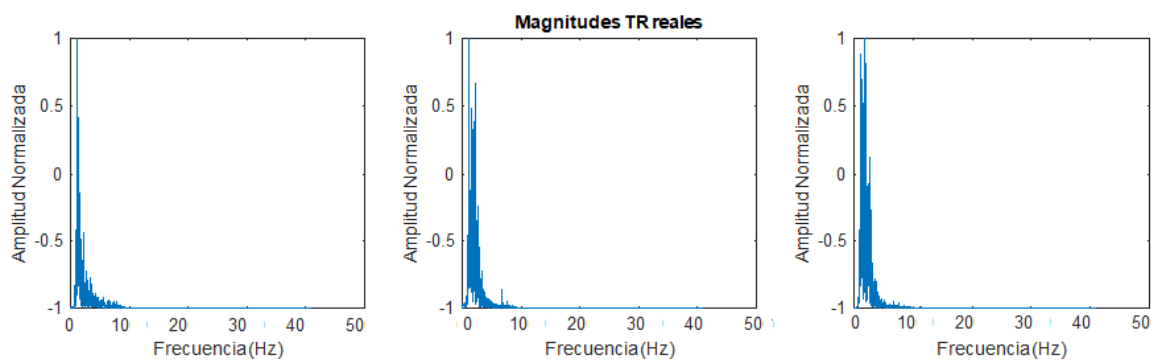
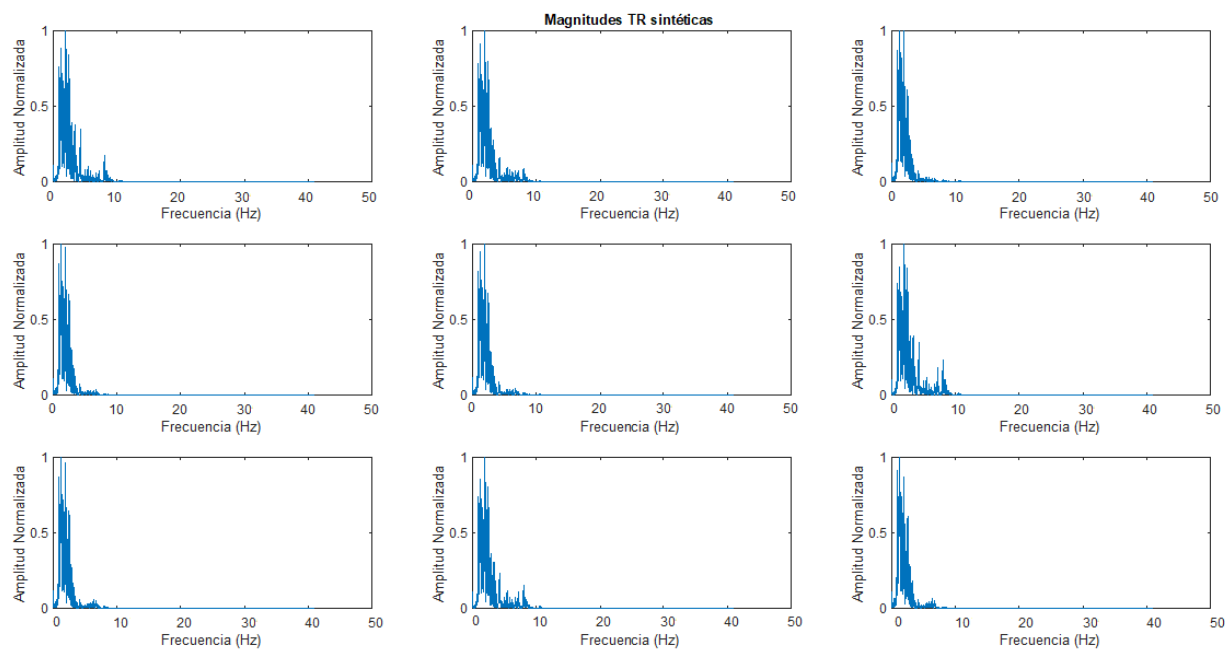
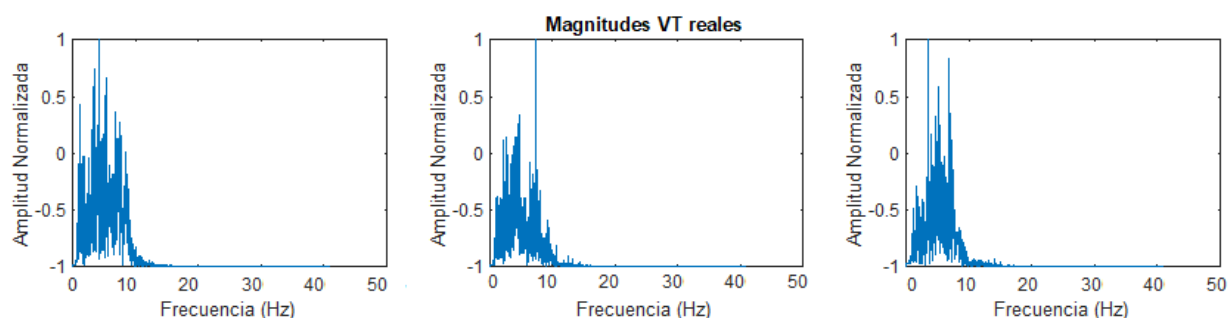
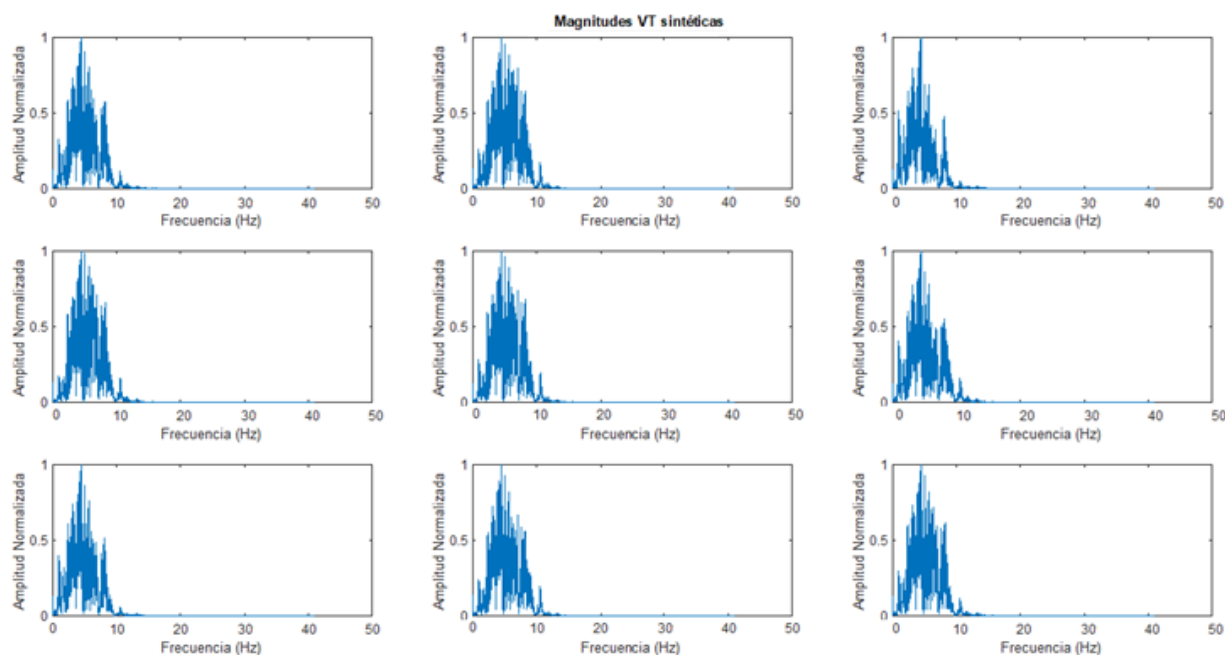
Figura 35*Magnitudes frecuenciales de eventos TR reales***Figura 36***Magnitudes frecuenciales de eventos TR sintéticos*

Figura 37*Magnitudes frecuenciales de eventos VT reales***Figura 38***Magnitudes frecuenciales de eventos VT sintéticos*

Como se observa, las magnitudes frecuenciales generadas con el modelo CGAN son similares unas de otras, aunque entre eventos si se tiene diferencia. Este fenómeno en el que no se genera gran variedad se conoce como modo colapso. El modo colapso es cuando la GAN produce una pequeña variedad de imágenes con muchos duplicados (modos). Esto sucede cuando el generador no logra aprender una representación rica, ya que tiende a asociar

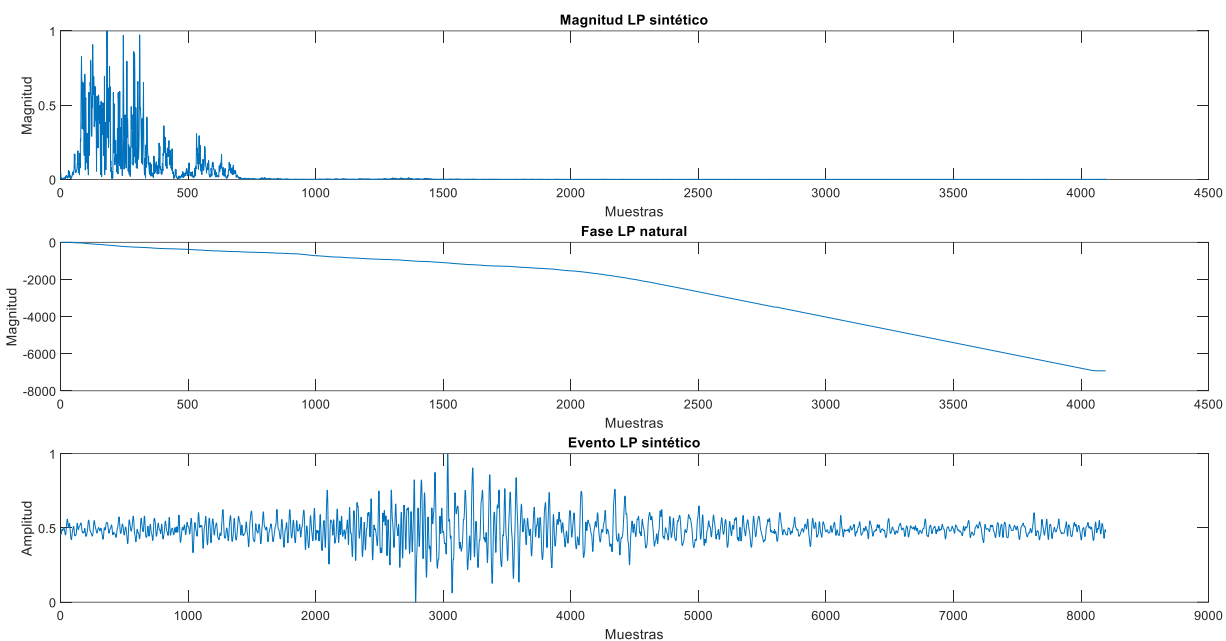
resultados similares con entradas diferentes. Para comprobar la presencia de un modo colapso, se debe examinar las imágenes generadas. Si se detecta poca diversidad en las salidas y algunas de ellas son prácticamente idénticas, es probable que haya un caso de modo colapso. Las soluciones planteadas en (MathWorks, 2023) incluyen el aumento de las dimensiones de los datos de entrada al generador. En el caso del presente trabajo se usa el valor de *numLatentInputs* de 100, este se cambió por 200 sin resultado positivo. Otra solución planteada es aumentar la cantidad de filtros del generador para permitirle detectar una variedad más amplia de características. En el trabajo se planteó el uso de 64 filtros tanto para el generador como para el discriminador, este valor fue cambiado por 128 solo en el generador, pero de igual manera que la anterior solución, no se obtuvo un resultado favorable. Y la última solución consistía en asignar aleatoriamente etiquetas falsas a imágenes reales, pero este proceso ya está implementado. Por tanto, al ver las características de la base de datos, en trabajos que están fuera del alcance de este proyecto, pero que se hicieron de manera simultánea, se observa que los intervalos de confianza de todos los eventos tienen poca variabilidad. Además, si es que se hace un diagrama de dispersión, las clases quedan demasiado compactas para cada evento. Al analizar esto se puede decir que el resultado de como la CGAN trabaja es por la naturaleza de la base de datos, en la cual las señales son “cuasi perfectas”.

Señales LP, TC, TR, VT en el tiempo

En la Figura 39, Figura 40, Figura 41 y Figura 42 se tiene la magnitud, la fase y la señal en tiempo generada de los eventos LP, VT, TC y TR. Las fases que se utilizan son las filtradas y son las que tienen la forma más lineal. Se muestran las magnitudes frecuenciales generadas con el modelo CGAN.

Figura 39

Magnitud LP sintético, fase LP natural y señal sintética microsísmica LP

**Figura 40**

Magnitud VT sintético, fase VT natural y señal sintética microsísmica de tipo VT

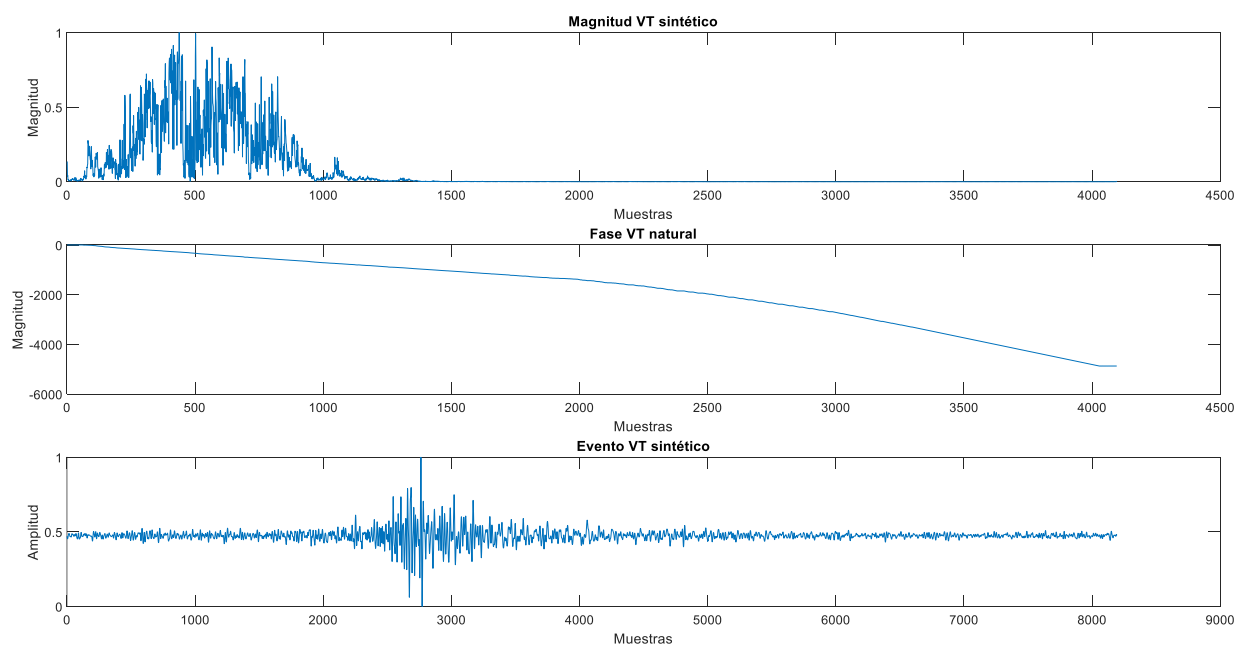
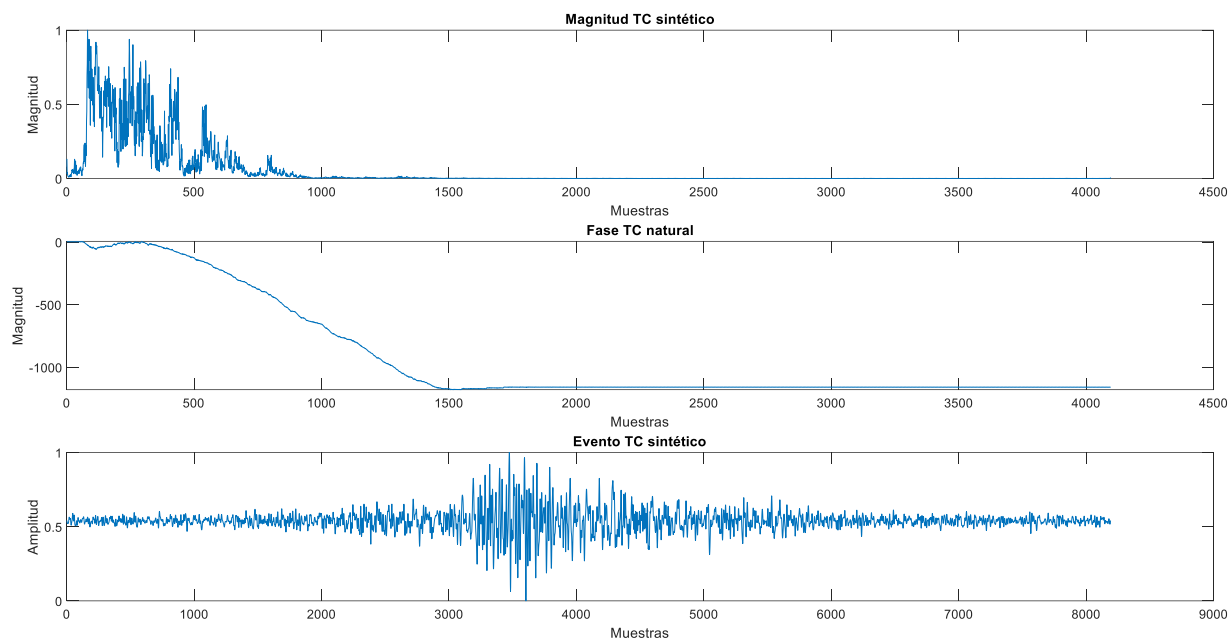
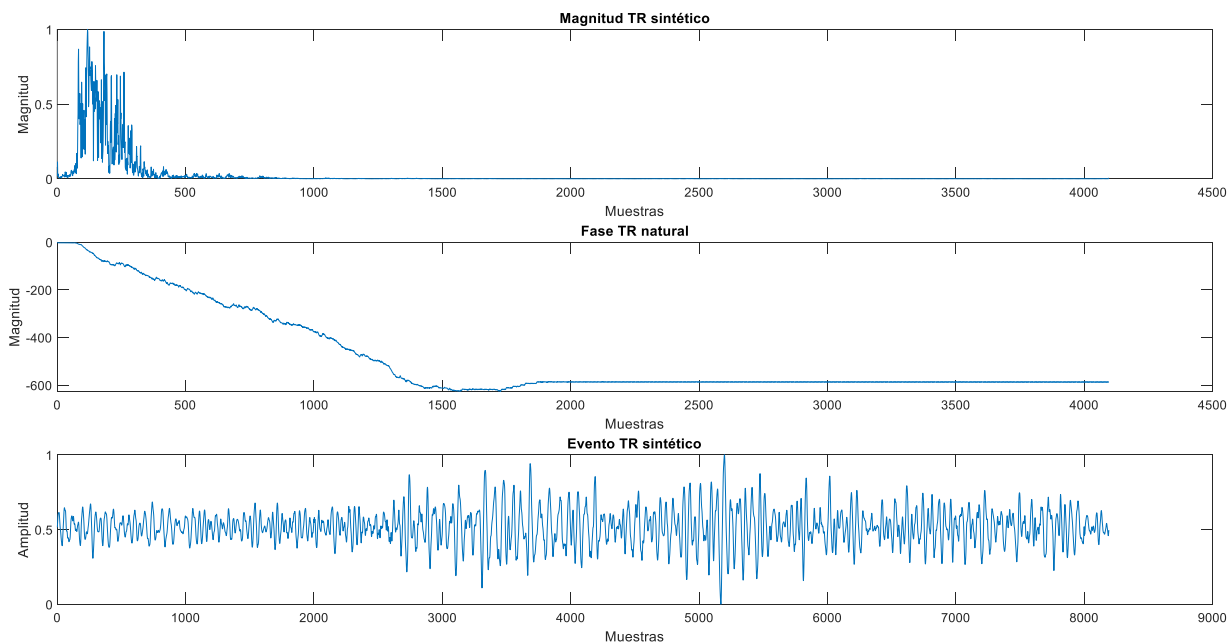


Figura 41

Magnitud TC sintético, fase TC natural y señal sintética microsísmica de tipo TC

**Figura 42**

Magnitud TR sintético, fase TR natural y señal sintética microsísmica de tipo TR



La representación de potencia se hace por medio del uso de la densidad espectral de potencia de Welch, en el presente trabajo se lo realiza con el uso de 512 puntos para una resolución media. Para esto se ha obtenido de cada uno de los eventos sintéticos, la representación en tiempo y su respectiva PSD. En la Figura 43 se tiene un ejemplo para microsismos sintéticos de LP que se puede contrastar con una señal microsísmica real en la Figura 44. Al hacer una inspección visual, las señales sintéticas se asemejan a las características de la real tanto en tiempo como en su PSD. La Figura 45 corresponde a un ejemplo de microsismos sintéticos de TC que se puede comparar con el evento real de la Figura 46. Como en el caso de la señal anterior, al observar el evento real y el sintético, tienen semejanzas tanto en tiempo como en su PSD. La Figura 47 corresponden a un microsismo TR sintético y en la Figura 48 corresponde a uno real. En este se puede ver en la PSD, que la forma tiene más ruido, pero que, al observarlo en el tiempo, se conservan las características del evento. Finalmente, para eventos VT se tiene un ejemplo de señales microsísmicas artificiales en la Figura 49 para una señal sintética y la Figura 50 para una señal real. Se debe tomar en cuenta que estos son ejemplos de señales reales, pero que hay una gran variedad en cuanto a su PSD, por el contrario, las PSD de las señales sintéticas son bastante similares unas de otras.

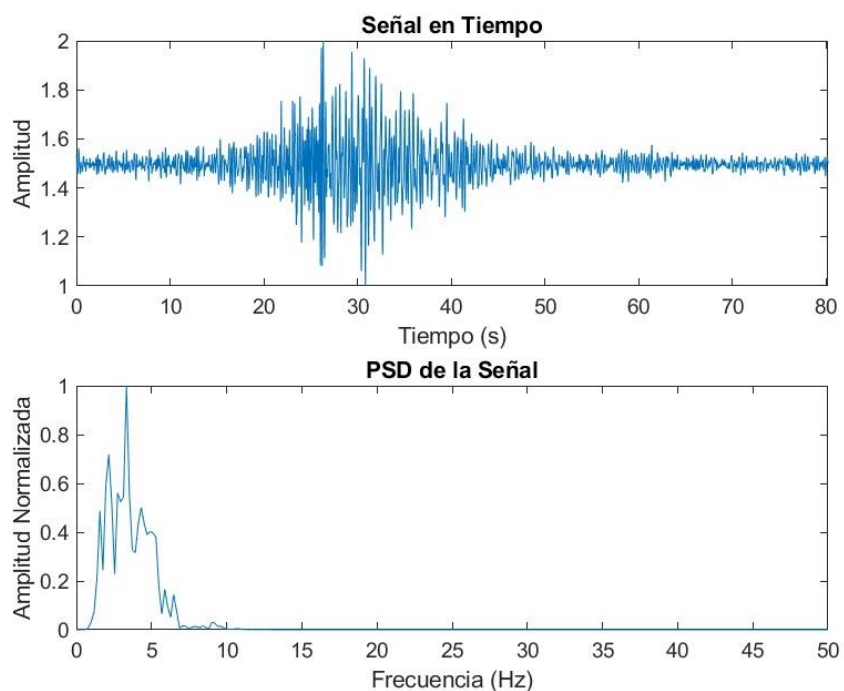
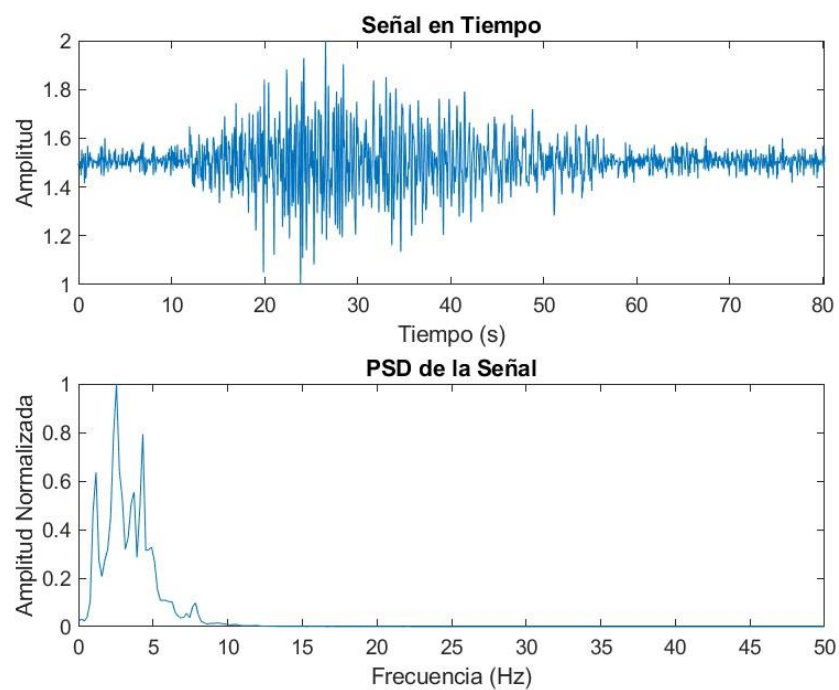
Figura 43*Señal microsísmica artificial LP y su PSD***Figura 44***Señal microsísmica real LP y su PSD*

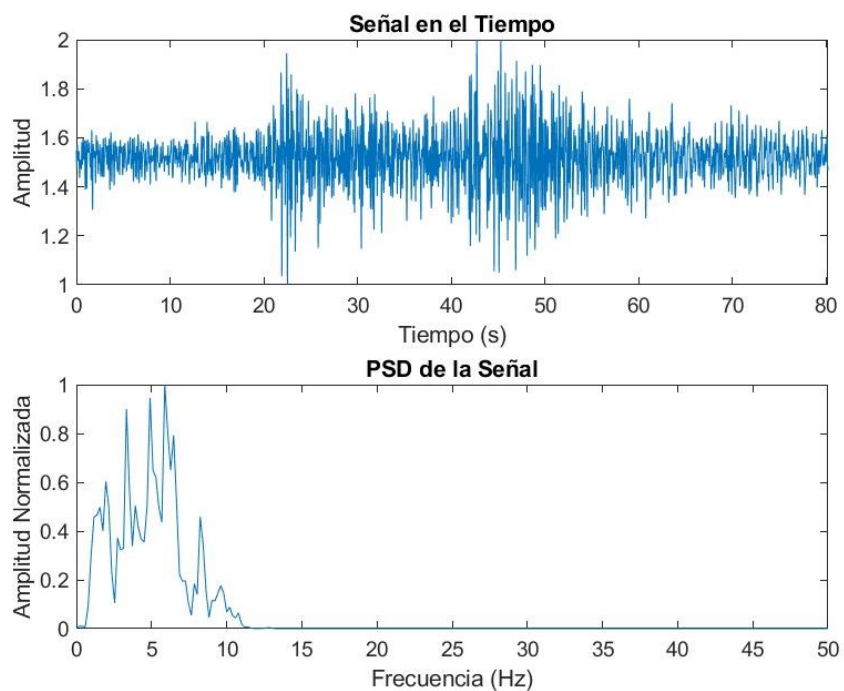
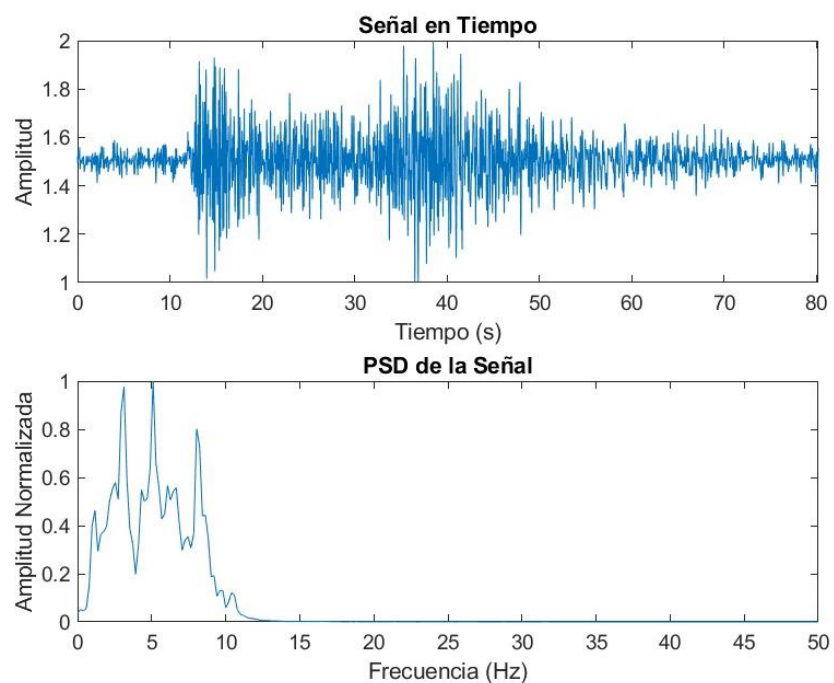
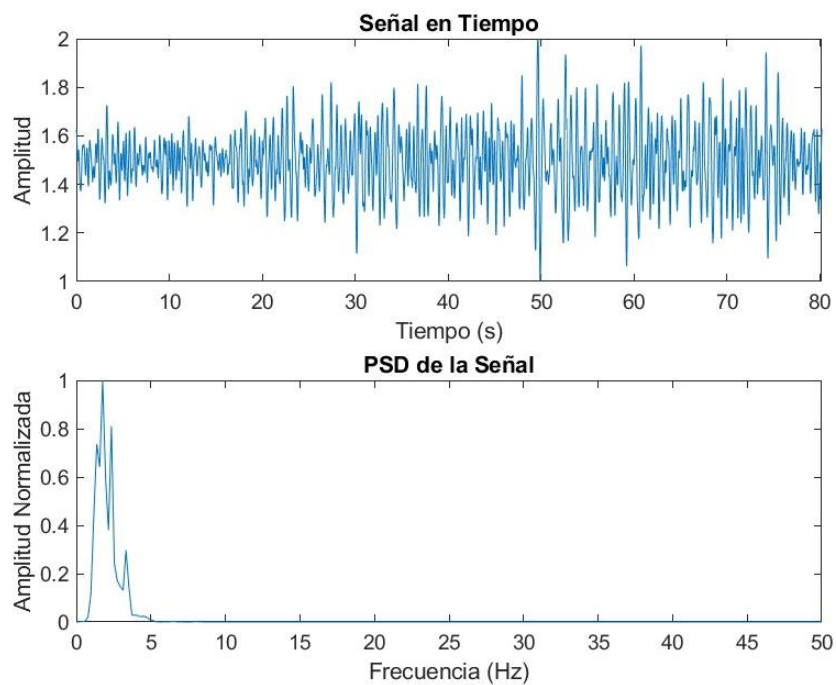
Figura 45*Señal microsísmica artificial TC y su PSD***Figura 46***Señal microsísmica real TC y su PSD*

Figura 47

Señal microsísmica artificial TR y su PSD

**Figura 48**

Señal microsísmica real TR y su PSD

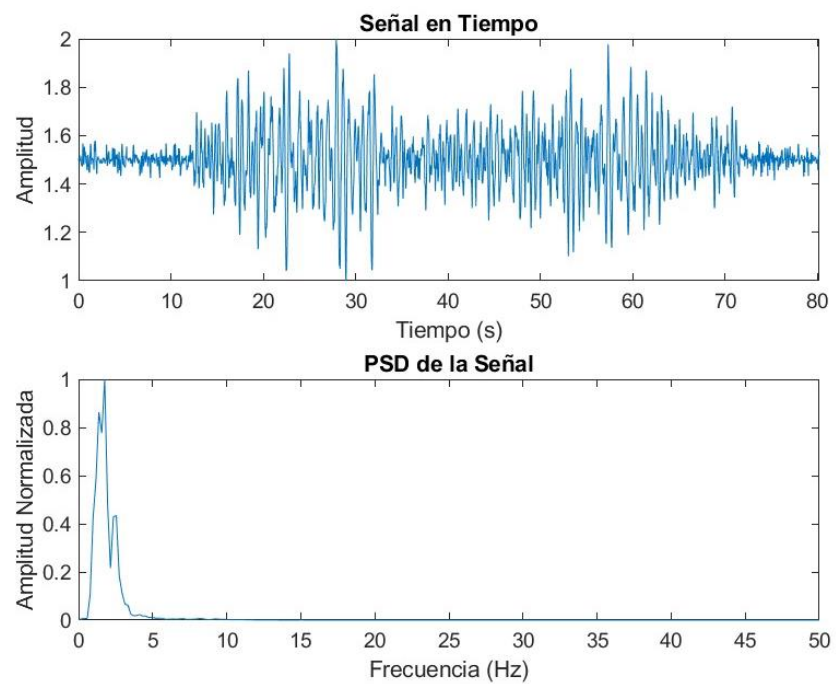
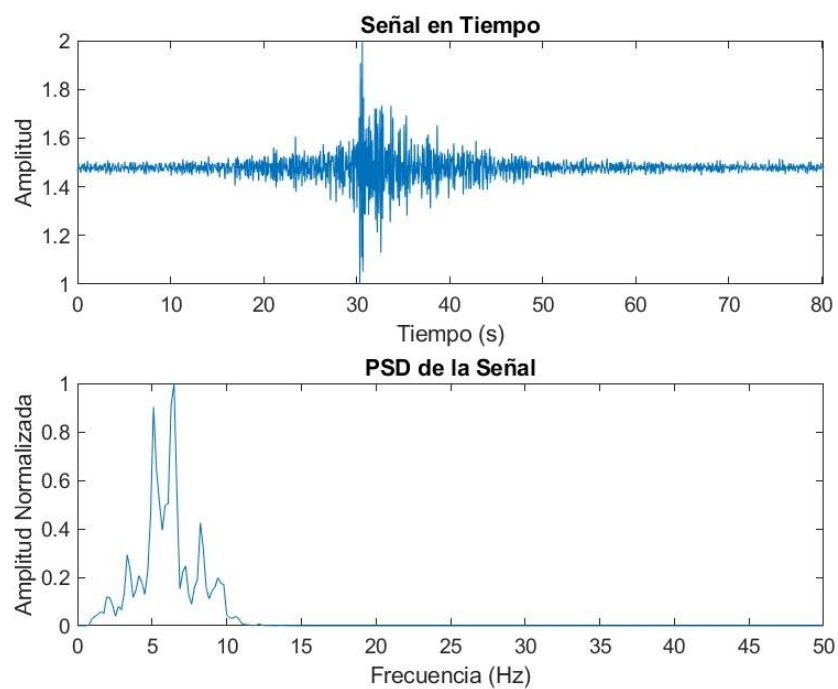
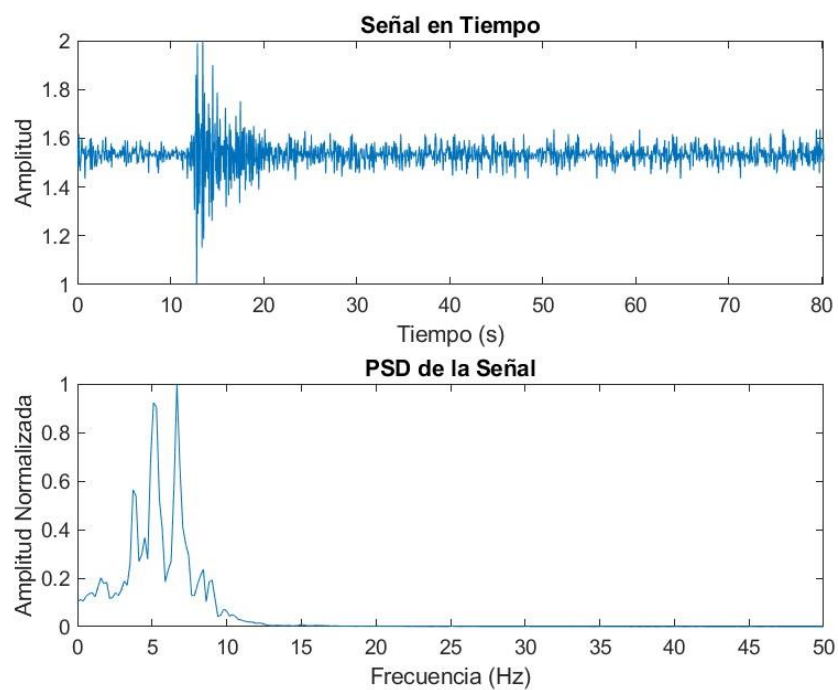


Figura 49*Señal microsísmica artificial VT y su PSD***Figura 50***Señal microsísmica real VT y su PSD*

Evaluación de señales sintéticas microsísmicas por medio de herramientas

Mediante la implementación de diversos clasificadores, se procede a categorizar las señales sintéticas del volcán Llaima por cada uno de los cuatro eventos generados. Se emplean tres métodos para llevar a cabo esta categorización, los cuales incorporan el análisis de métricas de evaluación como la P , la A , la R , la S y el BER.

Clasificación de microsismos por medio del uso del primer autoencoder

La primera herramienta empleada es un *autoencoder* el cual usa 257 características en frecuencia y 84 características extraídas de (Pérez et al., 2020), el cual tiene una exactitud de 81%. El análisis se lo realizó con 10000 señales sintéticas generadas con la aplicación desarrollada en este trabajo, la cual permite que 2500 sean señales microsísmicas de tipo LP, 2500 señales microsísmicas de tipo TC, 2500 señales microsísmicas de tipo TR y 2500 de tipo VT. Para el entrenamiento se usaron las señales reales y para el test las señales sintéticas. En la Figura 51 se observa la matriz de confusión para el primer *autoencoder*, en donde, el evento LP es representado por la clase 1, el evento TC por la clase 2, el evento TR por la clase 3 y el evento VT por la clase 4. En la Tabla 8 se muestran las métricas de evaluación del primer *autoencoder*.

Figura 51

Matriz de confusión de primer autoencoder

Confusion Matrix

Output Class	1	2067 20.7%	207 2.1%	115 1.1%	37 0.4%	85.2% 14.8%
	2	403 4.0%	2234 22.3%	5 0.1%	1043 10.4%	60.6% 39.4%
	3	8 0.01%	3 0.0%	2380 23.08%	0 0.0%	99.5% 0.5%
	4	22 0.2%	56 0.6%	0 0.0%	1420 14.2%	94.8% 5.2%
		82.7% 17.3%	89.4% 10.6%	95.2% 4.8%	56.8% 43.2%	81.0% 19.0%
	Target Class					
	1	2	3	4		

Tabla 8

Métricas de evaluación del primer autoencoder

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	92.08	85.20	82.68	94.28	0.10
TC	82.83	60.62	89.36	95.79	0.22
TR	98.69	99.54	95.20	98.42	0.01
VT	88.42	94.79	56.80	87.30	0.09
General	81.00	85.00	81.00	94.00	0.11

Nota. La tabla presenta las métricas A , P , R , S y BER, por medio del uso de un clasificador con el primer *autoencoder*, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Para el primer *autoencoder*, para señales LP, una cantidad considerable de señales se clasifican como TC, y viceversa. Cuando se trató de señales TC, las que se clasifican de manera incorrecta, en su mayoría lo hacen con LP. Al tratarse de señales TR, se equivoca con LP. La clasificación con más errores se produce al clasificar señales VT con señales TC. En consecuencia, las señales LP exhiben una tendencia a ser confundidas con las señales TC, mientras que las señales TR tienden a confundirse con las señales LP, y las señales VT muestran una confusión predominante con las señales TC.

Clasificación de microsismos por medio del uso del segundo autoencoder

La segunda herramienta empleada es un clasificador multiclase de alto desempeño que usa técnicas de selección dispersa de características y *autoencoders*. A través de las selecciones parciales se ha obtenido un total de 49 características, estructuradas de la siguiente forma: para un total de 257 características de la PSD se seleccionaron 26, para 16 características de coeficientes del modelo autoregresivo se seleccionaron 14. El modelo tiene una exactitud de 80.1%. El análisis se lo realizó con 10000 señales sintéticas generadas con la aplicación desarrollada en este trabajo, la cual permite que 2500 sean señales microsísmicas de tipo LP, 2500 señales microsísmicas de tipo TC, 2500 señales microsísmicas de tipo TR y 2500 de tipo VT. En la Figura 52 se observa la matriz de confusión para el segundo *autoencoder*, en donde, el evento LP es representado por la clase 1, el evento TC por la clase 2, el evento TR por la clase 3 y el evento VT por la clase 4. En la Tabla 9 se muestran las métricas de evaluación del primer *autoencoder*.

Figura 52

Matriz de confusión del segundo autoencoder

Confusion Matrix

Output Class	1	2499 25.0%	1781 17.8%	24 0.2%	0 0.0%	58.1% 41.9%
	2	0 0.0%	621 6.2%	54 0.5%	29 0.3%	88.2% 11.8%
	3	1 0.0%	0 0.0%	2422 24.2%	0 0.0%	100.0% 0.0%
	4	0 0.0%	98 1.0%	0 0.0%	2471 24.7%	96.2% 3.8%
		100.0% 0.0%	24.8% 75.2%	96.9% 3.1%	98.8% 1.2%	80.1% 19.9%
		Target Class				

Tabla 9

Métricas de evaluación del segundo autoencoder

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	81.94	58.06	99.96	99.98	0.21
TC	80.38	88.21	94.84	79.79	0.16
TR	99.21	99.96	96.88	98.97	0.01
VT	98.73	96.18	98.84	99.61	0.02
General	80.10	85.60	80.13	94.59	0.10

Nota. La tabla presenta las métricas A , P , R , S y BER, por medio del uso de un clasificador con el segundo *autoencoder*, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Para el segundo *autoencoder* la confusión con más peso es al clasificar TC que en su mayoría se clasifican como LP. Con el análisis de los dos *autoencoders*, los problemas se dan con las señales TC, esto se puede deber a la forma que tiene la señal ya que, si se observa en todo su rango, pareciera que la señal está constituida de varios tipos de microsismos.

Software para clasificación de microsismos por medio del uso de DT, k-NN, SVM y Votación.

La tercera aplicación utilizada es un sistema de clasificación de microsismos que utiliza distintas técnicas de Machine Learning como fueron DT, *k*-NN, SVM y el sistema implementado por los desarrolladores de la aplicación que consiste en un sistema de votación. Para la clasificación, el sistema no logra el éxito ya que con las distintas técnicas se obtuvo un máximo de 27.2% de exactitud.

En esta herramienta se ingresaron 10000 señales de las cuales 2500 eran señales microsísmicas de tipo LP, 2500 señales microsísmicas de tipo TC, 2500 señales microsísmicas de tipo TR y 2500 de tipo VT. En la Figura 53 se muestra la matriz de confusión que fue obtenida del clasificador al usar la técnica de DT con una exactitud de apenas el 27%, en la Tabla 10 se muestran sus métricas de evaluación. Para la clasificación con el uso de la técnica *k*-NN se puede observar la matriz de confusión en la Figura 54 con un 25.1% de exactitud, las demás métricas se las puede observar en la Tabla 11. Para el método SVM que fue usado como clasificador, la matriz de confusión corresponde la Figura 55 con un 25% de exactitud y sus métricas se observan en la Tabla 12. Finalmente, para la técnica de votación se tiene 27.2% en la exactitud observado en la Figura 56 y sus demás métricas en la Tabla 13. Finalmente, un resumen de los resultados con los distintos métodos de clasificación se muestra en la Tabla 14.

Figura 53

Matriz de confusión del sistema de clasificación por medio del uso de DT

Confusion Matrix

LP	2483 24.8%	67 0.7%	81 0.8%	2 0.0%	94.3% 5.7%
TC	7 0.1%	221 2.2%	2362 23.6%	14 0.1%	8.5% 91.5%
TR	1 0.0%	0 0.0%	0 0.0%	2484 24.8%	0.0% 100%
VT	9 0.1%	2212 22.1%	57 0.6%	0 0.0%	0.0% 100%
	99.3% 0.7%	8.8% 91.2%	0.0% 100%	0.0% 100%	27.0% 73.0%
	LP	TC	TR	VT	
	Target Class				

Tabla 10

Métricas de evaluación del clasificador por medio del uso de DT

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.33	94.30	99.32	99.77	0.03
TC	53.38	8.49	8.84	69.19	0.61
TR	50.15	0.00	0.00	66.73	0.67
VT	52.22	0.00	0.00	67.63	0.66
General	27.00	25.69	27.04	75.83	0.49

Nota. La tabla presenta las métricas A , P , R , S y BER, por medio del uso de un clasificador con el uso de la técnica DT, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Figura 54

Matriz de confusión del sistema de clasificación por medio del uso de *k*-NN

Confusion Matrix

LP	2498 25.0%	0 0.0%	158 1.6%	12 0.1%	93.6% 6.4%
TC	0 0.0%	15 0.1%	2315 23.2%	3 0.0%	0.6% 99.4%
TR	0 0.0%	0 0.0%	0 0.0%	2485 24.9%	0.0% 100%
VT	2 0.0%	2485 24.9%	27 0.3%	0 0.0%	0.0% 100%
	99.9% 0.1%	0.6% 99.4%	0.0% 100%	0.0% 100%	25.1% 74.9%
	LP	TC	TR	VT	

Target Class

Tabla 11

Métricas de evaluación del clasificador por medio del uso de *k*-NN

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.28	93.63	99.92	99.97	0.03
TC	51.97	0.64	0.60	67.59	0.65
TR	50.15	0.00	0.00	0.67	0.67
VT	49.86	0.00	0.00	0.67	0.67
General	25.10	23.57	25.13	42.22	0.51

Nota. La tabla presenta las métricas *A*, *P*, *R*, *S* y BER, por medio del uso de un clasificador con el uso de la técnica *k*-NN, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Figura 55

Matriz de confusión del sistema de clasificación por medio del uso de SVM

Confusion Matrix

LP	2498 25.0%	0 0.0%	130 1.3%	3 0.0%	94.9% 5.1%
TC	0 0.0%	0 0.0%	2336 23.4%	2 0.0%	0.0% 100%
TR	0 0.0%	0 0.0%	0 0.0%	2495 24.9%	0.0% 100%
VT	2 0.0%	2500 25.0%	34 0.3%	0 0.0%	0.0% 100%
	99.9% 0.1%	0.0% 100%	0.0% 100%	0.0% 100%	25.0% 75.0%
	LP	TC	TR	VT	

Target Class

Tabla 12

Métricas de evaluación del clasificador por medio del uso de SVM

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.65	94.94	99.92	99.97	0.03
TC	51.62	0.00	0.00	67.37	0.67
TR	50.05	0.00	0.00	66.68	0.67
VT	49.64	0.00	0.00	66.51	0.67
General	25.00	23.74	24.98	75.13	0.51

Nota. La tabla presenta las métricas A , P , R , S y BER, por medio del uso de un clasificador con el uso de la técnica SVM, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Figura 56

Matriz de confusión del sistema de clasificación por medio del uso de Votación

Confusion Matrix

LP	2498 25.0%	0 0.0%	122 1.2%	5 0.1%	95.2% 4.8%
TC	0 0.0%	221 2.2%	2334 23.3%	15 0.1%	8.6% 91.4%
TR	0 0.0%	0 0.0%	0 0.0%	2480 24.8%	0.0% 100%
VT	2 0.0%	2279 22.8%	44 0.4%	0 0.0%	0.0% 100%
	99.9% 0.1%	8.8% 91.2%	0.0% 100%	0.0% 100%	27.2% 72.8%
	LP	TC	TR	VT	
	Target Class				

Tabla 13

Métricas de evaluación del clasificador por medio del uso de Votación

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.71	95.16	99.92	99.97	0.02
TC	53.72	8.60	8.84	69.33	0.61
TR	50.20	0.00	0.00	66.76	0.67
VT	51.75	0.00	0.00	67.43	0.67
General	27.20	25.94	27.19	75.87	0.49

Nota. La tabla presenta las métricas A , P , R , S y BER, por medio del uso de un clasificador con el uso de la técnica de Votación, para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

En sistema de clasificación de microsismos donde se aplican distintas técnicas de Machine Learning, al considerar DT y querer clasificar señales TC, se clasifican como VT. Para las señales TR, en su totalidad se clasifican erróneamente como señales TC y para señales VT, en su totalidad se clasifican como TR. La tendencia para todos los métodos que restan, es la misma. La clasificación errónea se realiza entre los mismos tipos de señales. Igual que, en el análisis que se hace en los *autoencoders*, las señales TC y TR parecieran estar constituidas por distintos tipos de microsismos. Por lo tanto, la clasificación con el uso de solo la señal en el dominio del tiempo no es recomendada y se deben considerar también las características frecuenciales para mejorar la clasificación.

Tabla 14

Resultados de las herramientas empleadas

Herramienta	A (%)	R (%)	P (%)	S (%)	BER
Primer Autoencoder	81.00	85.00	81.00	94.00	0.11
Segundo Autoencoder	80.10	85.60	80.13	94.59	0.10
Sistema de clasificación DT	27.00	25.69	27.04	75.83	0.49
Sistema de clasificación k-NN	25.10	23.57	25.13	42.22	0.51
Sistema de clasificación SVM	25.00	23.74	24.98	75.13	0.51
Sistema de clasificación Votación	27.20	25.94	27.19	75.87	0.49

Nota. La tabla presenta las métricas *A, P, R, S* y BER, por medio del uso de dos técnicas, el uso de dos *autoencoders* diferentes y el sistema de clasificación que internamente usa cuatro técnicas para evaluar las distintas señales microsísmicas LP, TC, TR y VT.

Capítulo IV

Conclusiones y Trabajos Futuros

Conclusiones

Para este trabajo de integración curricular, se utilizó la base de datos que fue adaptada para que pueda trabajarse en el software de MATLAB®. La base de datos usada se denomina LlaimaSigV1 que fue recolectado de la estación LAV que monitorea el volcán Llaima, y los eventos etiquetados por especialistas del OVDAS. Contiene un total de 3592 eventos de cuatro clases diferentes de las cuales se tiene 304 VT, 1310 LP, 490 TR y 1488 TC. Las señales con las que se trabajó son todas las disponibles en la base de datos. Lo que se hizo de primera instancia es filtrar todas las señales de los eventos. Esto se hace debido al choque de las olas del mar contra la superficie terrestre, esto da como consecuencia que las señales presentan un componente frecuencial en 0.2 Hz. Esta característica conlleva al enmascaramiento de la señal real. Por lo tanto, se implementó un filtro FIR pasa banda. Seguido de esto, cada señal se estandarizó con la técnica z-score y se aplicó normalización de tipo máximo, se obtuvo además la respuesta en frecuencia por medio del uso de la FFT y finalmente se normalizó entre -1 y 1. Con estas señales y las etiquetas previamente obtenidas de cada tipo de evento, se empezó la etapa de diseño y entrenamiento de ambas redes que constituyen el modelo CGAN.

En la etapa en la que se diseñó la CGAN multiclase, se encontró que las redes componentes, es decir, tanto el generador como el discriminador, requerían de parámetros específicos. Estos parámetros incluyeron determinar el número de filtros, la profundidad de las redes en función del número de capas utilizadas, el tamaño del *kernel*, el *stride* y la elección de la función de activación, la cual está intrínsecamente relacionada con la naturaleza de cada capa. Al afinar estos parámetros se logró obtener un modelo final, el cual tuvo como objetivo la generación de señales sintéticas de magnitudes frecuenciales de los cuatro diferentes tipos de

eventos. La variación de distintos parámetros y sus respectivos resultados obtenidos fueron monitoreados gracias a la ayuda de herramientas diseñadas para este fin.

En la etapa de entrenamiento, con la ayuda de la monitorización, se comprobó que, al utilizar excesiva cantidad de filtros, el trabajo computacional es demasiado e incluso inmanejable para las limitaciones de las máquinas. Además, provoca un error de convergencia, es decir, la red no converge, por tanto, se fijó el número de filtros en un valor intermedio. También se probó el entrenamiento con un número de épocas significativo, en el cual se da otro error conocido como sobre entrenamiento. Hasta cierto punto el aprendizaje es correcto hasta que la convergencia se dispara hasta otros valores. Con este precedente se determinó que las épocas deberían ser menor a 100, por tanto, se toma 90 épocas y con esto se varía el valor de *minibatch*, se determinó que el mejor es 128. Finalmente, se llevó a cabo un entrenamiento adicional que incorporaba una variación en uno de los parámetros del optimizador de *Adam*. Como resultado de este ajuste, la red no requirió de valores elevados de épocas ni de *minibatch size*; en este caso, bastaron 13 y 5 respectivamente. El resultado final fue que el uso de recursos computacionales disminuyó y, además, se logró una convergencia más eficiente. Se observó que las magnitudes frecuenciales obtenidas eran semejantes entre sí. Por ende, se presentó un caso de modo colapso que se intentó solventar mediante varias técnicas, pero lamentablemente no se logró debido a la naturaleza de la base de datos. Sin embargo, al reconstruir la señal con las fases originales y las magnitudes sintéticas; luego llevarlo al dominio del tiempo, las señales correspondientes a los cuatro microsismos se obtienen como resultado una distinción clara.

Con la red CGAN entrenada, se creó una aplicación por medio del uso de *App Designer* de MATLAB® 2022. La aplicación permite la generación de señale sintéticas etiquetadas de los cuatro tipos de microsismos. En la aplicación se puede ingresar la cantidad de señales deseadas, las cuales para cada evento se generan 1/4 del total. La tabla obtenida está en *.mat*,

en donde se mantuvo el formato de la base de datos original que contiene: procedencia de la señal, la estación base que la monitorea, la frecuencia de muestreo, el componente, el año y el mes, la etiqueta del evento, la duración del evento, el punto de inicio y fin del evento y finalmente la señal sintética temporal. La aplicación también proporciona un gráfico de cada uno de los tipos de señales en el tiempo con su PSD respectiva.

De las herramientas para analizar las señales se obtuvo las exactitudes de la clasificación para los eventos LP, TC, TR y VT, con la utilización del primer *Autoencoder* el valor de 81%, segundo *Autoencoder* con 80.10%. Para el sistema de clasificación por medio del uso de DT 27%, cuando este sistema usó *k*-NN 25.10%, al haber usado SVM se obtuvo 25% y al haber usado el sistema de votación 27.20%. Los mejores resultados se dieron con el uso del primer *Autoencoder*, debido a que no solo se toman las características de tiempo si no también se lo hace con el uso de características de frecuencia. Los resultados obtenidos para exactitud mayor a 80% y con BER menor a 0.11 no aseguran que sean señales malas, pero tampoco que se deban usar para aumentar las bases de datos en estudios futuros que utilicen *Deep Learning*. Para solventar este problema y poder usar las señales generadas por el modelo planteado en este trabajo, se sugiere realizarlo bajo la utilización de un clasificador previo para garantizar y conservar solo las señales confiables.

Trabajos Futuros

Las herramientas utilizadas en la clasificación de microsismos no tienen altos valores de exactitud al usar la aplicación desarrollada en un trabajo paralelo a este en donde se usan varias técnicas de *Machine Learning*. De igual manera, el resultado al usar *autoencoders* no proporciona un porcentaje de error aceptable. Esto se debe a la naturaleza particular de la base de datos, en la cual las señales poseen un intervalo de confianza estrecho. Incluso al observar el diagrama de dispersión, las señales aparentan ser perfectas, es decir, exhiben un carácter académico. Por tanto, al entrenar la CGAN, no existen las suficientes características para hacer

la clasificación únicamente por medio del uso de las señales de los eventos en el dominio en el tiempo, para los *autoencoders* se tienen características adicionales que dan mejores resultados. Para mejorar estos resultados, en el entrenamiento de la CGAN se debe trabajar con bases de datos que proporcionen mayor variabilidad, ya que es importante, para técnicas de *Deep Learning*, trabajar con gran cantidad de señales, pero se debe considerar que tendrá mayor costo computacional.

Con el fin de reducir el costo computacional y lograr mejoras en los resultados, se espera implementar técnicas como *Transfer Learning*. Esto se basa en la premisa de utilizar redes previamente entrenadas en tareas de clasificación binaria. Se contempla la posibilidad de ajustar los parámetros internos de las redes generadora y discriminadora para adaptar el modelo CGAN a un enfoque multiclase, incluso se considera la duración de las nuevas señales. Otras técnicas de *Transfer Learning* que pueden ser tomadas en cuenta es hacerlo con redes entrenadas como *AlexNet* o *GoogleNet*, debido a que estas han sido entrenadas con una gran cantidad de imágenes y los resultados que se esperan serán mejores. Como en el anterior caso, para utilizar estas redes ya entrenadas, se debe considerar las modificaciones en las capas iniciales o finales e incluso el tamaño de entrada de las imágenes. En el presente trabajo se utilizaron las fases reales de cada uno de los microsismos, pero se podrían generar estas fases de manera artificial para ver qué tipo de resultados se tendrían.

Referencias

- Canário, J. P., Mello, R., Rios, R., Huenupan, F., & Curilem, M. (2020). *Data for: In-Depth Comparison of Deep Artificial Neural Network Architectures on Seismic Events Classification*. (Mendeley Data) Recuperado el 18 de Marzo de 2023, de <https://data.mendeley.com/datasets/dv8nwdd36k/1>
- Canário, P., Fernandes, R., & Curilem, M. (27 de Septiembre de 2021). *Data In Brief*. Recuperado el 2023 de Marzo de 2023, de Llama volcano dataset: In-depth comparison of deep artificial neural network architectures on seismic events classification: [https://www.data-in-brief.com/article/S2352-3409\(20\)30521-7/fulltext](https://www.data-in-brief.com/article/S2352-3409(20)30521-7/fulltext)
- Canário, P., Mello, R., Curilem, M., Huenupan, F., & Rios, R. (1 de Septiembre de 2020). *neural network architectures on seismic events classification*. (sciencedirect) Recuperado el 18 de Marzo de 2023, de <https://www.sciencedirect.com/science/article/abs/pii/S0377027319306171>
- Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). *Synthetic data augmentation using GAN for improved liver lesion classification*. (IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)) Recuperado el 19 de Marzo de 2023, de <https://ieeexplore.ieee.org/abstract/document/8363576>
- Galarza, C., & Vega, J. (15 de Julio de 2022). *Generación de señales sintéticas de eventos sísmo-volcánicos del volcán Cotopaxi a través de un modelo de red neuronal adversario generativo condicional*. (Repositorio ESPE) Recuperado el 19 de Marzo de 2023, de <http://repositorio.espe.edu.ec/bitstream/21000/31542/1/T-ESPE-052391.pdf>
- IGN. (s.f.). *Técnicas de vigilancia volcánica*. (Instituto Geográfico Nacional) Recuperado el 18 de Marzo de 2023, de https://www.ign.es/resources/docs/IGNCnig/TecnicasDeVigilancia_v8.pdf

- Jaramillo, C., León, R., Lara, R., Benítez, D., & Ruiz, M. (2014). *A New Structure for Sequential Detection and Maximum Entropy Spectral Estimator for Characterization of Volcanic Seismic Signals*. (IEEE Latin-America Conference on Communications) Recuperado el 19 de Marzo de 2023, de <https://ieeexplore.ieee.org/document/7041847>
- Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (11 de 6 de 2020). Training Generative Adversarial Networks with Limited Data. *Advances in Neural Information Processing Systems, 2020-December*, 37. doi:<https://doi.org/10.48550/arXiv.2006.06676>
- Kingma, D., & Ba, J. L. (2015). ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. *3rd International Conference on Learning Representations, ICLR 2015*(<https://arxiv.org/abs/1412.6980v9>), 1-15.
- Lara, F., Lara, R., Larco, J., Carrera, E., & León, R. (2021). *A deep learning approach for automatic recognition of seismo-volcanic events at the Cotopaxi volcano*. (Journal of Volcanology and Geothermal Research) Recuperado el 19 de Marzo de 2023, de <https://www.sciencedirect.com/science/article/abs/pii/S0377027320305783>
- Lara, R., Benítez, D., Carrera, E., Ruiz, M., & Álvarez, J. (2019). *Automatic Recognition of Long Period Events From Volcano Tectonic Earthquakes at Cotopaxi Volcano*. (IEEE Transactions on Geoscience and Remote Sensing) Recuperado el 19 de Marzo de 2023, de <https://ieeexplore.ieee.org/document/7479481>
- Lara, R., Benítez, D., Carrera, V., Ruiz, M., & Álvarez, J. (2016). *Feature selection of seismic waveforms for long period event detection at Cotopaxi Volcano*. (Journal of Volcanology and Geothermal Research) Recuperado el 19 de Marzo de 2023, de <https://www.sciencedirect.com/science/article/abs/pii/S0377027316300099>
- Lara, R., Bernal, P., Saltos, & Benítez, D. (2015). *Time and Frequency Feature Selection for Seismic Events from Cotopaxi Volcano*. (Asia-Pacific Conference on Computer Aided

- System Engineering) Recuperado el 19 de Marzo de 2023, de <https://ieeexplore.ieee.org/document/7287007>
- Lara, R., Larco, J., Benítez, D., 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) Recuperado el 19 de Marzo de 2023, de <https://www.sciencedirect.com/science/article/abs/pii/S0377027320303590>
- Lara, R., Rodriguez, M., & Larco, J. (1 de Agosto de 2020). *A real-time microearthquakes-detector based on voice activity detection and endpoint detection: An approach to Cotopaxi Volcano*. (sciencedirect) Recuperado el 19 de Marzo de 2023, de <https://www.sciencedirect.com/science/article/abs/pii/S0377027319305517>
- Loey, M., Manogaran, G., & Nour, K. (2020). *A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images*. (Neural Comput & Applic) Recuperado el 19 de Marzo de 2023, de <https://doi.org/10.1007/s00521-020-05437-x>
- Ma, F., Li, Y., Ni, S., Huang, S., & Zhang, L. (5 de Enero de 2022). *Data Augmentation for Audio-Visual Emotion Recognition with an Efficient Multimodal Conditional GAN*. (MDPI) Recuperado el 19 de Marzo de 2023, de <https://www.mdpi.com/2076-3417/12/1/527>
- Maas, A., Hannun, A., & Ng, A. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. 6.
- MathWorks. (2023). *Generate Synthetic Signals Using Conditional GAN*. Recuperado el 14 de Mayo de 2023, de <https://la.mathworks.com/help/signal/ug/generate-synthetic-pump-signals-using-conditional-generative-adversarial-network.html>

- MathWorks. (2023). *mathworks*. Obtenido de Monitor GAN Training Progress and Identify Common Failure Modes: <https://la.mathworks.com/help/deeplearning/ug/monitor-gan-training-progress-and-identify-common-failure-modes.html>
- ORACLE. (s.f.). *Puntuación Z*. (Oracle) Recuperado el 03 de Mayo de 2023, de https://docs.oracle.com/cloud/help/es/pbcs_common/PFUSU/insights_metrics_Z-Score.htm#PFUSU-GUID-640CEBD1-33A2-4B3C-BD81-EB283F82D879
- Pérez, B. (26 de Enero de 2022). *Generación de señales sintéticas de eventos sísmo-volcánicos del volcán Cotopaxi mediante el método de Bootstrap*. Obtenido de repositorio ESPE: <https://repositorio.espe.edu.ec/bitstream/21000/28429/1/T-ESPE-019175.pdf>
- 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>
- Radford, A. M. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 1-16.
- sernageomin. (2023). (Servicio Nacional de Geología y Minería) Recuperado el 18 de Marzo de 2023, de <https://rnvv.sernageomin.cl/glosario/>
- Servicio Nacional de Geología y Minería. (30 de Diciembre de 2019). *RNVV*. Obtenido de Sismo VT (Volcano-Tectónico): <https://rnvv.sernageomin.cl/faq-items/sismo-vt-volcano-tectonico/>

Servicio Nacional de Geología y Minería. (30 de Diciembre de 2019). *RNVV*. Obtenido de Sismo LP (Largo periodo): <https://rnvv.sernageomin.cl/faq-items/sismo-lp-largo-periodo/>

Servicio Nacional de Geología y Minería. (2023). *sernageomin*. Obtenido de RNVV: <https://rnvv.sernageomin.cl/volcan-llaima/>

The MathWorks. (2023). *Generate Synthetic Signals Using Conditional GAN*. Recuperado el 01 de Mayo de 2023, de <https://la.mathworks.com/help/signal/ug/generate-synthetic-pump-signals-using-conditional-generative-adversarial-network.html>

Yépez, C. (Octubre de 2022). *Generación de señales volcánicas artificiales del Volcán Cotopaxi usando modelos GAN (Generative Adversarial Networks) : Generación de señales volcánicas artificiales del tipo VT (Volcano-Tectonic) y LP (Long Period) usando un modelo cGAN*. (bibdigital EPN) Recuperado el 19 de Marzo de 2023, de <https://bibdigital.epn.edu.ec/bitstream/15000/23206/1/CD%2012627.pdf>

Zapata, J. (Octubre de 2022). *GENERACIÓN DE SEÑALES VOLCÁNICAS ARTIFICIALES DE TIPO LP (LONG-PERIOD) Y VT (VOLCANO-TECTONIC) A PARTIR DE UNA BASE DE DATOS DEL VOLCÁN COTOPAXI USANDO LA TÉCNICA DE BOOTSTRAPPING*. Obtenido de bibdigital EPN: <https://bibdigital.epn.edu.ec/bitstream/15000/23276/1/CD%2012691.pdf>