FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Detección automática de mensaje de texto oculto en un archivo de audio digital

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor:

Bach. Salazar Aguilar Luis
ORCID: https://orcid.org/0000-0001-9080-3189

Asesor:

Dr. Ramos Moscol Mario Fernando
ORCID: https://orcid.org/0000-0003-3812-7384

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2023

APROBACIÓN DEL JURADO

DETECCIÓN AUTOMÁTICA DE MENSAJE DE TEXTO OCULTO EN UN ARCHIVO DE AUDIO DIGITAL

_____

Mg. Bravo Ruiz Jaime Arturo
Presidente

_____

Mg. Mejia Cabrera Heber Ivan
Secretario

_____

Dr. Tuesta Monteza Victor Alexci
Vocal

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la **DECLARACIÓN JURADA**, soy Salazar Aguilar Luis del Programa de Estudios de Ingeniería de Sistemas de la Universidad Señor de Sipán S.A.C, declaro bajo juramento que soy autor del trabajo titulado:

DETECCIÓN AUTOMÁTICA DE MENSAJE DE TEXTO OCULTO EN UN ARCHIVO DE AUDIO DIGITAL

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación con las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firma:

Salazar Aguilar, Luis	DNI: 45160743	

Pimentel, 01 de febrero de 2023.

Dedicatoria

Este trabajo está dedicado a mis hijos, Adriana, Joaquin y Valentino; a mis padres y a los que con su apoyo me impulsaron a lograr este objetivo.

Agradecimientos

En primer lugar, agradecer a mi primo Roy Reyna por su apoyo, paciencia y por creer en mí, al profesor Ivan Mejia y demás profesores de la carrera que a lo largo de mis estudios me impartieron los conocimientos necesarios para este logro. Finalmente, agradecer a la universidad.

RESUMEN

En la actualidad la delincuencia cibernética se adapta a los avances de la tecnología, donde el robo de información en entidades estales y privadas se realiza enviando información robada en archivos multimedia como videos, imágenes y audios, de tal modo siendo vulnerado los sistemas de seguridad. Para la investigación se recolectó una base de datos de audios compuesto por audios de música y audios de voz, de tal manera se implementó un algoritmo de ocultamiento basado en una técnica esteganográfica con el método LSB así obteniendo una base de datos de audios originales y stego audios que son etiquetados con y sin mensaje oculto posteriormente usado para el entrenamiento de un algoritmo de machine Learning supervisado usando el clasificador SVM para entrenar la base de datos etiquetada y así generar un modelo. Así mismo se implementó un algoritmo que es el proceso inverso del algoritmo LSB para la decodificación del mensaje oculto en los archivos que fueron detectados. La presente investigación tiene como resultado que el algoritmo SVM predice si hay mensaje oculto en un audio digital. La presente investigación tiene como objetivo analizar y detectar si hay mensaje de texto oculto.

Palabra Clave: WAV, LSB, detección, Algoritmo de clasificación, esteganografía.

ABSTRACT

Nowadays, cybercrime adapts to the advances in technology, where information theft in state and private entities is done by sending stolen information in multimedia files such as videos, images and audios, in such a way that security systems are violated. For the research a database of audios composed of music audios and voice audios was collected, in such a way a concealment algorithm was implemented based on a steganographic technique with the LSB method thus obtaining a database of original audios and stego audios that are labeled with and without hidden message later used for the training of a supervised machine learning algorithm using the SVM classifier to train the labeled database and thus generate a model. Likewise, an algorithm that is the inverse process of the LSB algorithm was implemented for decoding the hidden message in the files that were detected. The result of this research is that the SVM algorithm predicts if there is a hidden message in a digital audio. The present research aims to analyze and detect whether there is hidden text message.

Keyword: WAV, LSB, detection, classification algorithm, steganography.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

I. INTRODUCCIÓN

1.1. Realidad problemática

En la actualidad los delincuentes tecnológicos o ciberdelincuentes, adaptándose al desarrollo de la tecnología, trafican con información robada de empresas y entidades gubernamentales vulnerando los sistemas de seguridad. Entre las diversas formas de robo de información, existe el tráfico de información a través del ocultamiento de mensajes de texto en archivos multimedia, donde el mensaje de texto es embebido en un archivo de audio, imagen o video; para posteriormente ser extraído por el receptor.

En el año 2002, se descubrió una red que intercambiaba información de pornografía infantil mediante el uso de la esteganografía. Asimismo, el año 2008, en los Estados Unidos, el Departamento de Justicia fue atacado con técnicas esteganográficas, donde cierta información financiera fue robada a través de imágenes con mensaje oculto. Por otro lado, el año 2011, los investigadores del laboratorio de criptografía y seguridad de sistemas en Budapest en Hungría, descubrieron un software malicioso denominado Duqu. Este malware vulnera dispositivos con sistema operativo Microsoft Windows para robar información; el funcionamiento del malware consiste en encriptar la información y usando un método esteganográfico, incrusta la información encriptada en una imagen JPEG (Wendzel, Mazurczyk, Caviglione & Meier, 2014).

Del mismo modo, Rupa, Shaikh, & Chinta (2021), mencionaron que, en los últimos años, ha habido informes sobre el uso de la esteganografía en espionaje, ataques terroristas, delitos y otras actividades. En el año 2001, algunos medios de comunicación de los Estados Unidos, informaron sobre comunicaciones secretas entre miembros del grupo terrorista Al Qaeda utilizando esteganografía, y en mayo del 2011, un sospechoso de Al Qaeda fue arrestado en Berlín, siendo encontrado con una tarjeta de memoria, la cual después de ser descifrada, se descubrió un video pornográfico llamado "KickAss", el cual contenía 141 documentos de texto ocultos, que incluían una gran cantidad de informes de acción de Al Qaeda, planes futuros, etc.

Asimismo, actualmente, los medios digitales proporcionan una fuente vital de información no degradable y de fácil manipulación, sin embargo, la facilidad con la que se pueden modificar las imágenes digitales hace que la verificación de la integridad de la imagen sea importante. Por lo tanto, existe el riesgo de violación de los derechos de autor de multimedia posiblemente manipuladas, como consecuencia, el uso de la marca de agua es de gran utilidad para la identificación de imágenes, la autenticación, la verificación y la ocultación de datos (Nassar, Faragallah & El-Bendary, 2021).

De acuerdo con la publicación de James (2021) en el portal de Financesonline, en la actualidad los problemas relacionados con la seguridad de la información afectan a todas las infraestructuras de grandes empresas, tal es así que el 40% de las computadoras han sido víctimas de ciberataques (Kaspersky, 2017). Según el estudio realizado por Statista en el año 2020, entre los ciberataques más comunes se tiene que un 38% corresponde a suplantación de identidad o phishing, 32% a intrusiones de red o network intrusion y 12% a divulgación inadvertida. Además, según GlobeNewswire (2021) ha habido un incremento del 15% en los ciberataques en todo el mundo en comparación con el año 2019.

La técnica de Machine Learning, también conocida como aprendizaje automático, de acuerdo con Management Solutions (2018), ha evolucionado en diferentes ramas en la vida cotidiana, contribuyendo a la automatización digital. En la actualidad percibimos que cada actualización de tecnología se viene complementando con la técnica de machine Learning como por ejemplo el uso de un celular Smart que solo puede ser activado solo con el iris de los ojos, huellas dactilares e incluso con patrones de voz. Estas técnicas de aprendizaje automática son una potente herramienta para la implementación de algoritmos capaces de detectar información oculta en archivos multimedia.

En resumen, nos enfrentemos a una realidad problemática tecnológica, la cual esta basada en el robo de información y transmitida de manera oculta, embebida o encriptada a través de archivos multimedia. Este problema de ingeniería que se afronta es abordado en la presente investigación para detectar la existencia de información oculta en archivos de audio digital.

1.2. Trabajos Previos

Para los autores Ali, Mokhtar & George (2017), los métodos de esteganografía de audio modernos más conocidos son la codificación de fase (phase coding), la codificación de bits bajos (low-bit coding), el eco-ocultamiento (echo hidding), la codificación de paridad (parity coding), el dominio de wavelet (wavelet domain) y el espectro ensanchado (spread spectrum).

Según Alwan (2013), en su investigación titulada: A New Audio Steganography Technique For Hiding Text Message, propuso un nuevo método para ocultar texto en diferentes archivos de audio, usando el algoritmo LSB (Least Significant Bit), teniendo como resultado que el archivo de audio original y el archivo de audio con mensaje oculto no presentan diferencia significativa audible.

En este mismo sentido los autores Ali, Mokhtar & George (2017), publicaron un artículo, el cual propone un esquema conocido como ECA-BM, para mejorar el rendimiento de la esteganografía de audio, el cual contribuye en aumentar la capacidad de ocultación, mantiene la transparencia del portador y mejora la seguridad del modelo propuesto. De los resultados experimentales se concluyó que hubo un aumento significativo en la capacidad de ocultación en comparación con algunos estudios relacionados y, además, se conserva la fidelidad del archivo de audio y el archivo secreto reconstruido.

Según Mohtasham & Mosleh (2019), en su publicación titulada Steganalysis based on collaboration of fractal dimensions and convolutional neural networks, investigaron la detección de mensajes ocultos en audios digitales, para lo cual ocultaron información con los métodos de esteganografía Hide4PGP y StegHide, de tal manera que con el uso de los algoritmos clasificadores SVM uno de ellos el algoritmo de Machine Learning supervisado analizaron las métricas de rendimiento, obteniendo como resultados como la precisión con 57.8% seguido de la exactitud 76.67% y finalmente recall de 73.80%.

Según Ru, Zhang & Huang (2005), en su investigación titulada steganalysis of audio: attacking the steghide, estudiaron la técnica esteganográfica steghide, utilizando una base de datos con diferentes audios digitales entre audios

originales y audios originales con mensaje oculto, de tal manera que tomaron en cuenta la mitad de la base de datos para el entramiento para que puedan diferenciar entre las señales de audio limpias y las señales de audios con mensaje oculto, y los restante de audios de la base de datos para la prueba correspondiente, de los resultados que se obtuvieron muestran que tienen una óptima precisión y exactitud. Adicionalmente los autores concluyen en que el SVM se puede aplicar en otros distintos métodos esteganográficos

Los autores Sewisy, Mansour, Rida & Mohammed (2015), en su investigación proponen una técnica la cual sugiere que el mensaje de texto se codifica mediante el método de codificación Huffman y se integra en un archivo de audio mediante el algoritmo LSB, obteniéndose como resultado un nuevo archivo de audio y luego se contrasta mediante el uso de varios valores que incluyen; PSNR (relación señal / ruido pico) y SNR (relación señal / ruido). Posteriormente se esquematiza la frecuencia del archivo de audio antes y después del mensaje de texto oculto, obteniendo como conclusión que las pruebas indican que el método sugerido es comparativamente eficaz en archivos de texto cifrados incrustados en audio.

Azam, Ridzuan, Sayuti & Alsabhany (2019), en su investigación evaluaron los métodos de incrustación de 6 bits menos significativos (LSB) para capturar el rendimiento del archivo stego basado en la relación señal/ruido pico (PSNR). Los resultados de su investigación mostraron el efecto de cambiar el punto de incrustación inicial para el método de incrustación LSB, ante ello se propone un nuevo parámetro para maximizar la imperceptibilidad dado un valor de capacidad en el método de incrustación LSB; como conclusión, se puede considerar la selección del punto de incrustación inicial mientras se desarrolla un nuevo método de incrustación LSB que prioriza la incrustación en el nivel superior de LSB.

De acuerdo con Fateh, Rezvani & Irani (2021), la esteganografía basada en el bit menos significativo (LSB) puede dividirse en dos categorías: reemplazo del LSB (LSBR) y coincidencia del LSB (LSBM). En la esteganografía de imágenes, el LSBR reemplaza cada bit del mensaje secreto en los bits menos significativos del archivo portador. Por otro lado, el LSBM elige incrementar o

disminuir aleatoriamente algunos píxeles en la imagen portadora en la que sus bits significativos más bajos son diferentes de los bits apropiados en el mensaje secreto. En ambas categorías, el receptor puede simplemente extraer el mensaje secreto de los bits significativos más bajos en la imagen portadora con el mensaje oculto.

Los autores Zhang, Du & Li (2013), en su trabajo de investigación: Esteganálisis of LSB Matching in WAV Audio, investigaron un método de esteganálisis para buscar y analizar las coincidencias que se encuentran en un archivo multimedia esteganográfico, donde se consideran archivos de audios WAV de 16 bits, a los cuales se le introdujo información oculta usando la técnica LSB. La investigación determinó que el método propuesto logra un mejor rendimiento en la clasificación de los archivos de audio con contenido oculto.

Chávez & Gutiérrez (2020), investigaron sobre el ocultamiento de información confidencial en imágenes BMP y audio WAV mediante el método LSB. Para el ocultamiento de datos en un audio en formato WAV, utilizaron un audio con 3 segundos de duración, con una longitud de 94,226 bytes y un archivo de datos con una longitud de 10,448 bytes el cual fue ocultado en el archivo de audio de formato WAV. En el ocultamiento se cambió 40,457 bits menos significativos del audio original, lo cual representa un 5% del contenido y el 95 % restante no fue modificado.

Los autores Ghasemzadeh & Kayvanrad (2018), en su investigación titulada “Comprehensive Review of Audio Steganalysis”, estudiaron el esteganálisis de archivos de audio referenciando investigaciones de diferentes autores, considerando que sus principales objetivos fueron la extracción de características del audio, teniendo en cuenta los tipos de señales, el método comprimido y el método no comprimido. Así mismo, realizaron una comparación de diferentes métodos usando la misma base de datos tanto en métodos de esteganografía LSB, como no LSB en escenarios universales.

1.3. Teorías relacionadas al tema

1.3.1. Formato WAV

El formato WAV (Waveform Audio File), es un formato de audio digital originario de Microsoft Windows 3.1 y en la actualidad se mantiene con la extensión WAV. Es el formato más usado por los usuarios de Windows y es compatible con sistemas operativos como Linux, Mac entre otros. Además, una de las ventajas del formato WAV, es que puede ser usado para el tratamiento de sonido digital y puede ser grabado en diferentes calidades y tamaños. Por otro lado, el formato WAV puede convertirse en diferentes formatos tales como MP3, OGG, entre otros (Calvillo, 2005).

1.3.2. Esteganografía

El término esteganografía proveniente del griego “Steganos” que significa cubierto u oculto y “Graphos” que significa escritura; cuyo significado es escritura oculta. La esteganografía es una técnica científica para ocultar información secreta en un archivo multimedia que se puede enviar de manera segura a través de una red no segura, por lo que la esteganografía evita que los datos incrustados sean descifrados (Patel, Lad & Patel, 2021).

Según los autores Angulo, Ocampo & Blandon (2007), la esteganografía es una tecnología avanzada en el proceso de ocultación de información y la definen como el arte de disfrazar la información en archivos de imágenes sonido o en canales encubiertos haciendo uso de métodos y técnicas computacionales.

Para Singh, Moudgil, & Rani (2021), la esteganografía ayuda al ocultamiento de información dentro de un archivo multimedia el cual será transferido a través de la red, el archivo multimedia final que contiene el mensaje embebido es denominado “stego file”. Asimismo, el estudio de los ataques basados en esteganografía se denomina Estegoanálisis.

1.3.3. Criptografía

La palabra criptografía proviene de la unión de los términos griegos “Kryptos” que significa oculto y “Graphein” que significa escritura, y su definición es: “Arte de escribir con clave secreta o de un modo enigmático”. En la actualidad,

la criptografía se ha convertido en un conjunto de técnicas, cuyo objetivo consiste en la protección a través del ocultamiento de información frente a observadores no autorizados (Lucena, 2010).

Según Singh, Moudgil, & Rani (2021), la criptografía es ampliamente usado para ocultar mensajes, una característica particular de esta técnica es que el mensaje es visible pero no es posible leerlo o descifrarlo sin contar con la llave de cifrado. Asimismo, el estudio de los ataques basados en criptografía es denominado criptoanálisis.

Por otro lado, a diferencia de la esteganografía, en la criptografía se conoce la existencia del mensaje encriptado en un archivo multimedia, mientras que en la esteganografía nadie sospecha de un mensaje oculto (Kar, Nakka & Katangur, 2018).

1.3.4. Watermarking

De acuerdo con Nassar, Faragallah & El-Bendary (2021), se define como un proceso que incrusta un dato llamado marca de agua (watermark) en un archivo multimedia, el cual puede ser una imagen, audio o video, para que la marca de agua se pueda detectar o extraer posteriormente. El ocultamiento se puede realizar en el dominio espacial o en un dominio de transformación del archivo multimedia.

Los autores Kar, Nakka & Katangur (2018), mencionan que la marca de agua se usa para proteger la propiedad intelectual en un archivo multimedia. Por otro lado, es posible usar técnicas de marca de agua invisibles para aplicaciones esteganográficas y viceversa. A diferencia de la esteganografía, los archivos digitales con marca de agua pueden ser convertidos de un formato a otro pudiendo ser detectados de forma comprensible. Asimismo, a diferencia de la esteganografía el contenido de la marca de agua siempre es rastreable.

1.3.5. Least Significant Bit (LSB)

La codificación de bits menos significativo LSB, es una de las técnicas que se utiliza para ocultar información en archivos de audios digitales, sustituyendo el bit menos significativo del punto de muestro de un mensaje binario. De tal

manera el LSB es una de las técnicas que mayor cantidad de datos puede ocultar (Nehru & Dhar, 2012).

1.3.6. Machine Learning

Según Francés (2020), Machine Learning, es una rama de la inteligencia artificial “IA” que programa las máquinas mediante algoritmos perfeccionado su funcionamiento y aprendizaje.

En 1956 John McCarthy definió la inteligencia artificial, en una conferencia en el Dartmouth College, como “la ciencia y la tecnología para crear maquinas inteligentes”, la Comisión Europea en una definición más actualizada describe a la inteligencia artificial como “los sistemas que manifiestan un comportamiento inteligente, pues son capaces de analizar su entorno y pasar a la acción con cierto grado de autonomía con el fin de alcanzar objetivos específicos”. En la actualidad el objetivo de la inteligencia artificial es alcanzar la inteligencia a nivel humano (Francés, 2020).

Según Jung (2019), en su investigación titulada: A Study on Machine Learnig for Steganalysis, menciona que la técnica de Machine Learning es un método de aprendizaje automático. Además, menciona que esta técnica se puede clasificar como aprendizaje supervisado, aprendizaje no supervisado y aprendizaje reforzado.

El aprendizaje supervisado, consiste en aprender una función que mapea una base de datos de entrada con datos de salida conocidos, es decir los datos de entrada están etiquetados o clasificados. El aprendizaje no supervisado, consiste en aprender de una base de datos, de los cuales no se conoce el resultado de salida, es decir los datos de entrada no están etiquetados o clasificados. El aprendizaje reforzado, se refiere a la toma de decisiones para maximizar alguna noción de recompensa acumulativa (Jung, 2019).

1.3.7. Support Vector Machine (SVM)

El algoritmo SVM (Support Vector Machine), es un algoritmo de clasificación que divide las clases en regiones separadas por un hiperplano de N dimensiones. El caso más simple se muestra en la Figura 1, considerando 2 dimensiones en el que el hiperplano está definido por una línea recta que

divide las dos regiones de clasificación, es decir, una clasificación binaria donde el resultado de la clasificación tiene dos posibles resultados: (+1) para la clase A y (-1) para la clase B (Aguirre, 2017).

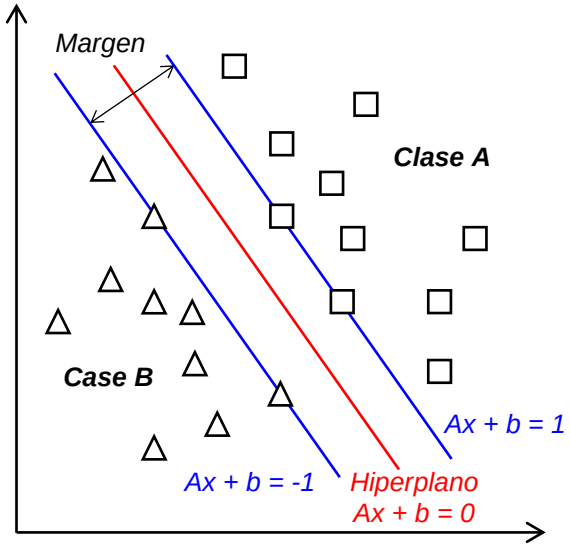

Figura 1. SVM 2D

1.3.8. Esteganálisis

La aplicación de métodos esteganográficos para poder ocultar información es de uso frecuente. Estos métodos utilizan archivos digitales como imágenes, audio o video, para incrustar la información oculta. El esteganálisis es una técnica que analiza los archivos multimedia para determinar la existencia de información oculta y posteriormente decodificar dicha información (Nehru & Dhar, 2012).

El esteganálisis es un método que tiene como objetivo detectar si algún archivo de multimedia tiene alguna información adicional oculta el cual podrá ser extraída posteriormente. El esteganálisis en audios digitales extrae las propiedades estadísticas vulnerando el ocultamiento esteganográfico (Zhang, Du & Li, 2013).

Según Zhang, Du & Li (2013), en su investigación usaron métodos de esteganálisis para poder extraer los datos ocultos de un archivo de audio, considerando la correlación entre el ruido y la señal de un audio esteganográfico que fue incrustado con el método LSB.

1.4. Formulación del problema

¿Cómo detectar mensajes de texto oculto en un archivo de audio digital?

1.5. Justificación e importancia del estudio

El desarrollo de la presente investigación se justifica debido a la necesidad de prevenir el tráfico de información oculta en archivos de audio, para lo cual se necesita implementar una herramienta de detección de mensajes ocultos a través del desarrollo de un algoritmo que sea capaz de identificar si existe un mensaje oculto en un archivo de audio. De esta manera, en el Perú, se puede prevenir y alertar a las autoridades competentes de posibles ciberataques que pongan en riesgo la seguridad de personas, empresas y el gobierno.

1.6. Hipótesis

Mediante la implementación de un algoritmo de clasificación, se puede detectar mensajes de texto oculto en un archivo de audio digital.

1.7. Objetivos

1.7.1. Objetivo general

Detectar de manera automática mensajes de texto oculto en un archivo de audio digital.

1.7.2. Objetivos específicos

a) Elaborar una base de datos con diferentes audios digitales en formato WAV.
b) Ocultar un mensaje de texto en los archivos de audio digital de la base de datos.
c) Implementar un algoritmo de detección previamente seleccionado.
d) Realizar simulaciones de detección y análisis de resultados.

II. MATERIAL Y MÉTODO

2.1. Tipo y diseño de investigación

La investigación según su finalidad fue de tipo aplicada, debido a que su ejecución aporto en la solución de la problemática detectada, al respecto Behar (2008), indicó que para que una investigación sea aplicada, el nuevo conocimiento debe aportar en la solución de problemas prácticos o concretos, y además, la investigación debe tener una orientación al problema específico.

Según su enfoque, la investigación es cuantitativa, al respecto los autores Hernández, Fernández y Baptista (2014), indicaron que las investigaciones cuantitativas recolectan datos numéricos para ser medidos y analizados estadísticamente con la finalidad de conocer su comportamiento, probar hipótesis o teorías basadas en medidas numéricas y análisis estadísticos.

La presente investigación corresponde a un diseño cuasi-experimental debido a que se realiza de manera intencionada, manipulando una variable independiente para examinar a través un procedimiento definido la existencia de información oculta dando como resultado una relación sobre una o más variables dependientes. (Hernández-Sampieri y Mendoza, 2018).

2.2. Población y muestra

2.2.1. Población

Según Valderrama (2015), la población es un conjunto de personas o cosas, tangibles o intangibles, con características comunes o relacionadas entre sí que son observables. Además, se debe tener en cuenta los elementos que los conforman, el periodo o tiempo y lugar donde se realiza la investigación. Asimismo, Lozano (2018), define la población como un conjunto de todos los individuos, personas u objetos, que poseen una o más características en común y forman parte de un estudio o investigación.

Para la presente investigación, la población está constituida por todos los audios digitales en formato WAV (Waveform Audio File Format), que pueden o no contener información oculta usando algún método esteganográfico.

2.2.2. Muestra

De acuerdo con Lozano (2018), la muestra es un grupo representativo de elementos de una población, reduciendo los costos y el tiempo de realización de la investigación, siendo posible generalizar los resultados de la muestra a toda la población en estudio. Del mismo modo Valderrama (2015), define la muestra como un subconjunto representativo de la población, el cual refleja fielmente las características de la población. La utilidad de la muestra depende de cómo fueron seleccionados sus elementos, si la muestra no es representativa de la población los resultados obtenidos son pocos fiables, es decir se tiene muestra sesgada.

Para la presente investigación se utilizó un método no estadístico intencionada de acuerdo al criterio del investigador, la cual está constituida por 2000 audios digitales en formato WAV (1000 con y 1000 sin información de mensaje oculto) para el entrenamiento y prueba del modelo de clasificación, y 1000 audios digitales en formato WAV (500 con y 500 sin información de mensaje oculto) para el aporte práctico.

2.3. Variables y Operacionalización

2.3.1. Variables

Según Behar (2008), define las variables como las características, propiedades o dimensiones de un fenómeno y que pueden asumir distintos valores, estas pueden ser clasificadas según la medición que se le realice, cuantitativas, cualitativas, independientes y dependientes.

Variable Independiente

Como variable independiente se tiene al algoritmo de clasificación.

Variable dependiente

Por otro lado, como variable dependiente se tiene la detección automática de un audio.

2.3.2. Operacionalización

La operacionalización de variables se describe en la Tabla 1

Tabla 1.
Operacionalización de variables

Variables	Dimensiones	Indicadores	Ítem	Técnica e instrumentos de recolección de datos
Variable Independiente Algoritmo de Clasificación	Tiempo de ejecución	Tiempo de ejecución	$T = T_{final} - T_{inicial}$	Observación y registro electrónico
	Consumo de memoria	Memoria consumida durante un proceso	$C_m = \sum_{j}^{n} \frac{cm_j}{n}$	
	Consumo de CPU	CPU consumido durante un proceso	$C_c = \sum_{j}^{n} \frac{cc_j}{n}$	
Variable Dependiente Detección de Mensajes	Rendimiento	Matriz de Confusión	$[CM]$	Observación y registro electrónico
		Exactitud	$E = \frac{VP + VN}{VP + VN + FP + FN}$	
		Precisión	$P = \frac{VP}{VP + FP}$	
		Recall	$R = \frac{VP}{VP + FN}$	
	Ocultamiento	SNR	$SNR = \frac{\bar{X}}{\sqrt{\frac{\sum(X - \bar{X})^2}{N}}}$	
		PSNR	$PSNR = 10 \log_{10}\left(\frac{X_{max}^2}{\sum(X - Y)^2 / N}\right)$	

(Fuente: Elaboración propia)

2.4. Técnicas e instrumentos de recolección de datos

En la presente investigación se desarrolló un algoritmo de Machine Learning supervisado, basado en el algoritmo SVM de clasificación para detectar audios en formato WAV que contengan información de texto oculto. Para dicho propósito se utilizó el lenguaje de programación Python en la plataforma Spyder.

La técnica usada en la presente investigación corresponde a la observación de los indicadores de interés para evaluar el algoritmo de clasificación, los cuales son el tiempo de ejecución, el consumo de memoria RAM y el consumo de CPU del equipo de cómputo durante la ejecución del algoritmo de clasificación. Asimismo, para evaluar la detección de los mensajes ocultos en un archivo de audio en formato WAV, se observó el rendimiento de la detección a través de la matriz de confusión, exactitud, precisión y recall de cada audio analizado con el algoritmo de detección. Por otro lado, se observó la calidad del ocultamiento con los indicadores SNR y PSNR.

Para la recolección de datos en la presente investigación se usó un instrumento de recolección de datos de registro electrónico, con la finalidad de documentar los respectivos resultados del algoritmo de clasificación y detección de mensajes de texto oculto en audios en formato digital.

2.4.1. Tiempo promedio de ejecución

El tiempo de ejecución es considerado desde el tiempo inicial, $T_{inicial}$, de la ejecución del algoritmo, considerando el preprocesamiento de la base de datos y la extracción de características, hasta el tiempo final, T_{final}, en que se analiza un archivo de audio, detectando la existencia o no de contenido oculto. El tiempo promedio calcula sumando el tiempo de ejecución de cada audio analizado entre la cantidad de audios que componen la base de datos (n).

$$T = \frac{1}{n}\sum_{i=1}^{n}(T_{final} - T_{inicial})$$

2.4.2. Grado de consumo de memoria RAM

En términos de grado de consumo de memoria RAM es la evaluación del consumo de memoria durante el proceso de ejecución del método propuesto.

$$C_m = \sum_{j}^{n} \frac{cm_j}{n}$$

2.4.3. Grado de consumo de CPU

En términos de grado de consumo de CPU evalúa el consumo del consumo del procesador durante el proceso de ejecución del método propuesto.

$$C_c = \sum_{j}^{n} \frac{cc_j}{n}$$

2.4.4. Matriz de confusión (CM)

La matriz de confusión es usada para medir el rendimiento de un modelo de predicción, y está compuesta por: VP, el total de verdaderos positivos, VN, el total de verdaderos negativos, FP, el total de falsos positivos y FN, el total de falsos negativos. La matriz de confusión se muestra a continuación:

CM = REAL		PREDICCIÓN: ORIGINAL	PREDICCIÓN: STEGO
	ORIGINAL	VN	FP
	STEGO	FN	VP

2.4.5. Exactitud (E)

Es la fracción de predicciones verdaderas respecto de las predicciones totales de todos los audios. La exactitud se puede calcular de la siguiente manera:

$$E = \frac{VP + VN}{VP + VN + FP + FN}$$

2.4.6. Precisión (P)

Es la fracción de predicciones verdaderas positivas de audios con mensaje oculto respecto del total de predicciones de audios detectados con contenido oculto.

$$P = \frac{VP}{VP + FP}$$

2.4.7. Recall (R)

Es la fracción de predicciones verdaderas positivas de audios con mensaje oculto respecto de la cantidad total de audios que realmente tienen mensaje oculto.

$$R = \frac{VP}{VP + FN}$$

2.4.8. Signal to Noise Ratio (SNR)

La Relación Señal Ruido de una señal puede ser calculado como la relación entre la media $\bar{X}$, y su desviación estándar. Esta relación se puede obtener para cualquier señal ya sea un audio original o un stego audio. El SNR se puede calcular con la siguiente expresión:

$$SNR = \frac{\bar{X}}{\sqrt{\frac{\sum(X - \bar{X})^2}{N}}}$$

2.4.9. Peak Signal to Noise Ratio (PSNR)

El Pico de Relación Señal Ruido es una medida de relación entre el audio original y el stego audio, el cual se puede interpretar como: a mayor valor de PSNR mayor la calidad del ocultamiento de información en el stego audio (Msallam, 2020). El PSNR se puede calcular con la siguiente expresión:

$$PSNR = 10\, log_{10}\left(\frac{X_{max}^2}{\sum(X - Y)^2/N}\right)$$

Donde: X_{max}, es el valor máximo posible de la señal, X, es la señal de audio original, Y, es la señal del stego audio y N, la longitud de las señales, considerando que ambas señales tienen la misma duración.

2.5. Procedimiento de análisis de datos

Diagrama de flujo de entrenamiento y prueba usando SVM para generar un modelo de detección es mostrado en la Figura 2.

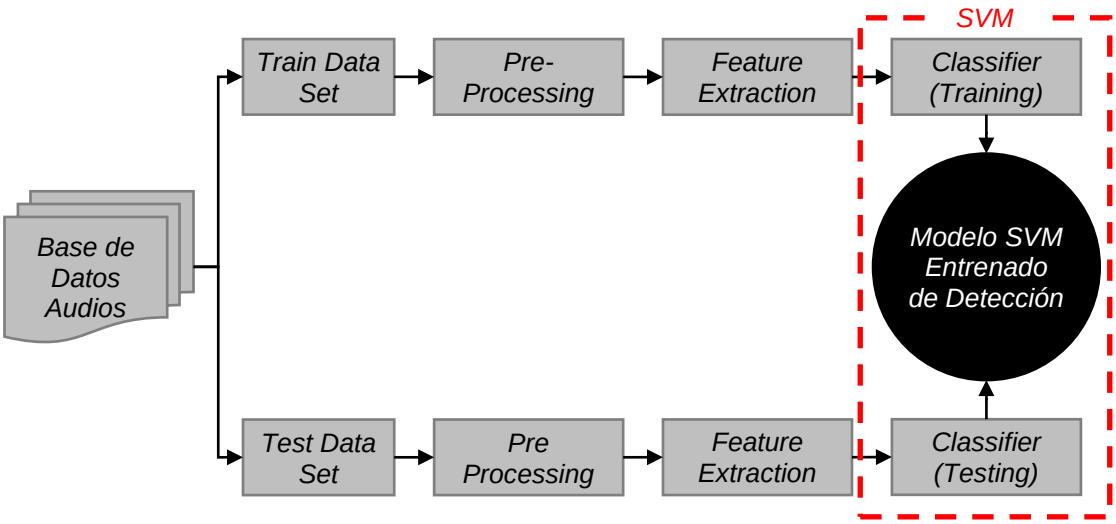

Figura 2. Diagrama de flujo de entrenamiento y test
(Fuente: Elaboración propia)

2.5.1. Base de datos de audios

La base de datos está compuesta por audios digitales en formato WAV, dichos audios son de dos tipos: audios originales, los cuales son audios que no tienen información oculta, y los stego audios, que corresponden a los mismos audios originales a los cuales se les incrustó un mensaje oculto.

Para el entrenamiento y prueba del modelo, se generó una base de datos de 2000 audios digitales en formato WAV compuesto por 1000 audios originales y 1000 stego audios, la cual se dividió en 2 grupos: el Training Data Set, compuesto por el 65% de la base de datos, con los cuales se realizó el entrenamiento del algoritmo SVM usado para la clasificación binaria, y el 35% restante de la base de datos fue usado para validar el modelo entrenado (Test Data Set). La clasificación está definida como: un audio que no contiene mensaje oculto (cero: 0) o un audio que si contiene mensaje oculto (uno: 1).

2.5.2. Pre-Procesamiento (Pre-Processing)

El pre-procesamiento de los audios, ya sea un audio original o stego audio consiste en realizar la descomposición de la señal usando la transformada de wavelet (ver Figura 3).

Seguidamente, se sustrae el ruido de la señal (ver Figura 4) y posteriormente se reconstruye la señal (ver Figura 5).

```
def wavelet_decomposition(audio_data):
    data = audio_data.copy()
    #print(f"data: {data}")
    wave = np.zeros_like(data)
    c = 1 if len(data) % 2 == 0 else 0
    for i in range((len(data)//2) - c):
        wave[int(2*i + 1)] = data[int(2*i + 1)] - ((data[int(2*i)] - data[int(2*i + 2)]) / 2)
        if i > 0:
            wave[int(2*i)] = data[int(2*i)] + ((wave[int(2*i - 1)] + wave[int(2*i + 1)] + 2) / 4)
    #print(f"wave: {wave}")
    count = 0
    for i in range(len(wave) // 2):
        count += wave[int(2*i + 1)]
    return wave, count/(len(wave)//2)
```

Figura 3. Algoritmo de descomposición de wavelet

(Fuente: Elaboración propia)

```
def cut_noise(wave, avg):
    for i in range(len(wave) // 2):
        num = wave[int(2*i + 1)]
        if num > avg:
            num -= avg
        if num < -avg:
            num += avg
        else:
            num = 0
        wave[int(2*i + 1)] = num
    #print(f"cut_wave: {wave}")
    return wave
```

Figura 4. Algoritmo de sustracción del ruido

(Fuente: Elaboración propia)

```
def wavelet_reconstruction(wave_data):
    wave = wave_data.copy()
    #print(f"wave: {wave}")
    wave2 = np.zeros_like(wave)
    for i in range(1, len(wave)//2):
        wave2[int(2*i)] = wave[int(2*i)] - ((wave[int(2*i - 1)] + wave[int(2*i + 1)] + 2) / 4)
    c = 1 if len(wave) % 2 == 0 else 0
    for i in range((len(wave)//2) - c):
        wave2[int(2*i + 1)] = wave[int(2*i + 1)] + ((wave2[int(2*i)] + wave2[int(2*i + 2)])
/ 2)
    #print(f"wave2: {wave2}")
    return wave2
```

Figura 5. Algoritmo de reconstrucción de wavelet

(Fuente: Elaboración propia)

2.5.3. Extracción de características (Feature Extraction)

La extracción de características conformará el data-set de parámetros de entrada “X” con etiquetas “y”, para cada audio analizado; cabe resaltar que los parámetros de entrada corresponden a las características de la relación entre

el ruido y la señal de un audio. La Figura 6 muestra el código para la extracción de características de cada audio.

```
# == Local Correlation ==
def local_corr(wave):
    a = wave[:]
    b = np.append(wave[1:], 0)
    return a - b
# == Single Markov ==
def single_markov(wave):
    suma = 0
    prevlsb = 0
    for i in range(len(wave)):
        lsb = wave[i] & 1
        suma += abs(lsb - prevlsb)
        prevlsb = lsb
    val = suma/len(wave)
    return val
# == Double Markov ==
def double_markov(wave):
    suma = 0
    count = 0
    prevlsb = 0
    double_prev = 0
    for i in range(len(wave)):
        lsb = wave[i] & 1
        if prevlsb == 1 & double_prev == 1:
            count += 1
            if lsb == 1:
                suma += 1
        if prevlsb == 0 & double_prev == 0:
            count += 1
            if lsb == 0:
                suma += 1
        double_prev = prevlsb
        prevlsb = lsb
    return suma / count
```

Figura 6. Código de extracción características
(Fuente: Elaboración propia)

2.5.4. Entrenamiento

Para el entrenamiento del modelo, se usó un algoritmo de Machine Learning basado en el modelo SVM para la clasificación de audios con y sin contenido oculto. La Figura 7 se muestra el script del entrenamiento del modelo.

```
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from joblib import dump
import os
import numpy as np
import definitions
path = os.path.join(definitions.FEAT_DATASET, 'dataset.npy')
with open(path, 'rb') as f:
    dataset = np.load(f)
X = dataset[:,0:-1]
y = dataset[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.35,
random_state=32)
clf = SVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy model: {acc}")
cm = confusion_matrix(y_test, y_pred)
print(f"Confusion Matrix: \n{cm}")
```

Figura 7. Script de entrenamiento

(Fuente: Elaboración propia)

2.5.5. Predicción

Para la predicción de la existencia de mensaje oculto en un audio digital analizado, se utilizó el modelo SVM entrenado. A continuación, se muestra el código de predicción en la Figura 8.

```
def make_prediction(dataset_dict, model):
    X = np.array(dataset_dict["X"])
    pred_dur = []
    pred = []
    for i in range(X.shape[0]):
        start_time = time.time()
        y = int(model.predict(X[i,:].reshape((1,4))))
        pred_dur.append(time.time() - start_time)
        pred.append(y)
    dataset_dict["pred"] =  pred
    dataset_dict["pred_dur"] = pred_dur
    return dataset_dict
proc_dataset = build_dataset(proc_audio_path, 0)
stego_dataset = build_dataset(stego_audio_path, 1)
proc_dataset = make_prediction(proc_dataset, clf)
stego_dataset = make_prediction(stego_dataset, clf)
y_test = proc_dataset["y"] + stego_dataset["y"]
y_pred = proc_dataset["pred"] + stego_dataset["pred"]
```

Figura 8. Código de predicción

(Fuente: Elaboración propia)

2.6. Criterios éticos

En este aspecto se toman en cuenta dos criterios principales la confidencialidad y derechos de autor, validado con el Código Deontológico del Colegio de Ingenieros del Perú.

Confidencialidad: La información recolectada para esta investigación no será publicada, de esta manera se protegerá la información recaudada y de las personas involucradas en ella.

Derechos del Autor: Toda documentación textual, gráfico, estadístico, etc. utilizada en esta investigación serán referenciadas y citadas de acuerdo con la norma APA sexta edición.

2.7. Criterios de rigor científico

Para esta investigación los criterios que se tomaron en cuenta son las siguientes: (Hernández, Fernández y Baptista, 2014)

La Credibilidad:

La credibilidad se fundamenta con investigaciones similares realizadas por otros autores, obteniendo resultados y conclusiones similares con la presente investigación.

Transferibilidad o Aplicabilidad:

La transferibilidad del conocimiento sobre el escenario o muestra analizada de la investigación busca aplicar las conclusiones a una muestra o escenarios similares, replicando de esta manera resultados consistentes con la presente investigación.

Confiabilidad:

En esta investigación la confiablidad se asocia a los resultados que son obtenidos de las pruebas que se realizaron, los mismos que pueden cambiar en función a la base de datos analizada.

Neutralidad:

La neutralidad garantiza que los resultados de la investigación no están sesgados por motivaciones, intereses o perspectivas del investigador.

III. RESULTADOS

3.1. Resultados en tablas y figuras

3.1.1. Entrenamiento y prueba del modelo

Para la presente investigación se generó una base de datos de 2000 audios digitales en formato WAV compuesto por 1000 audios originales y 1000 stego audios. Para el proceso de entrenamiento se consideró el 65% de audios de la base de datos que corresponde a 1300 audios y 35% de audios para el test o prueba del modelo entrenado que corresponde a 700 audios. Cabe resaltar que la elección de audios para el entrenamiento (Train Data Set) y prueba (Test Data Set) de la base de datos fue realizada de manera aleatoria.

En la Figura 9 se puede apreciar que el 55.1% de los audios que componen la base de datos analizada, equivalente a 551 audios, tienen un tiempo promedio de duración de 9 segundos y una desviación estándar de 1.205.

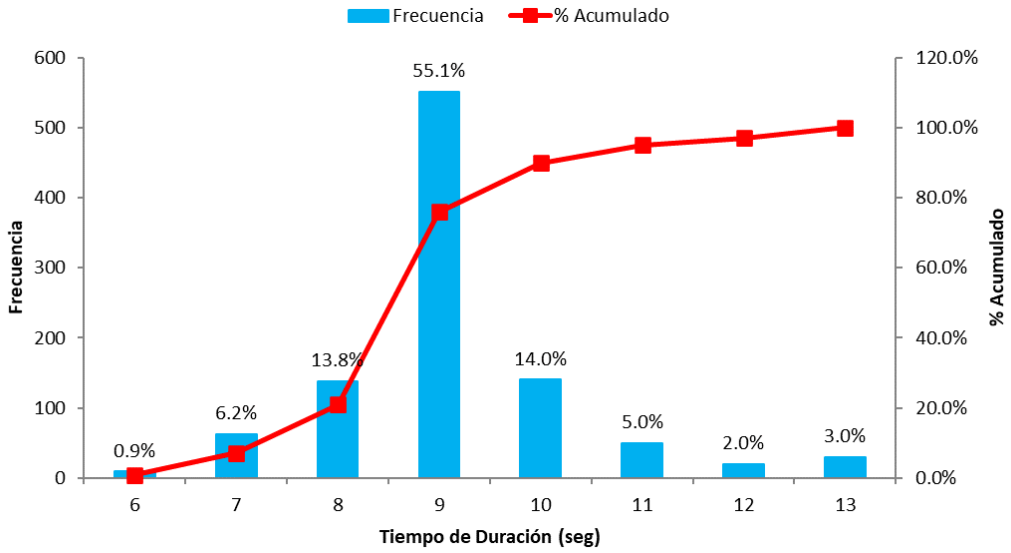

Figura 9. Histograma de tiempo de duración de audios
(Fuente: Elaboración propia)

Asimismo, en la Tabla 2 muestra las frecuencias y porcentajes acumulados de los tiempos de duración de la base de datos analizada.

Tabla 2.
Frecuencia y porcentaje acumulado de tiempos de duración de audios

Tiempo (seg)	Frecuencia	% Acumulado
6	9	0.9%
7	62	7.1%
8	138	20.9%
9	551	76.0%
10	140	90.0%
11	50	95.0%
12	20	97.0%
13	30	100.0%

(Fuente: Elaboración propia)

De los resultados obtenidos en la prueba del modelo usando el modelo entrenado, se obtuvo a siguiente matriz de confusión:

Tabla 3.
Matriz de confusión de la prueba de entrenamiento

		PREDICCIÓN	
		AUDIO ORIGINAL	STEGO AUDIO
REAL	AUDIO ORIGINAL	332	0
	STEGO AUDIO	0	368

(Fuente: Elaboración propia)

VN - VERDADERO NEGATIVO : **332** Audios originales detectados como originales.

VP - VERDADERO POSITIVO : **368** Stego audios detectados como Stego audios

FN - FALSO NEGATIVO : **0**

FP - FALSO POSITIVO : **0**

$$E = \frac{VP + VN}{VP + VN + FP + FN} = \frac{332 + 368}{332 + 368 + 0 + 0} = 1 = 100\%$$

$$P = \frac{VP}{VP + FP} = \frac{368}{368 + 0} = 1 = 100\%$$

$$R = \frac{VP}{VP + FN} = \frac{368}{368 + 0} = 1 = 100\%$$

De los valores obtenidos en la matriz de confusión para la prueba de entrenamiento del modelo, en la Figura 10 se observan las métricas de rendimiento de la prueba de entrenamiento de los 700 audios digitales en

formato WAV, en el cual se observa un 100.0% de Precisión, 100.0% de Exactitud y 100.0% de Recall.

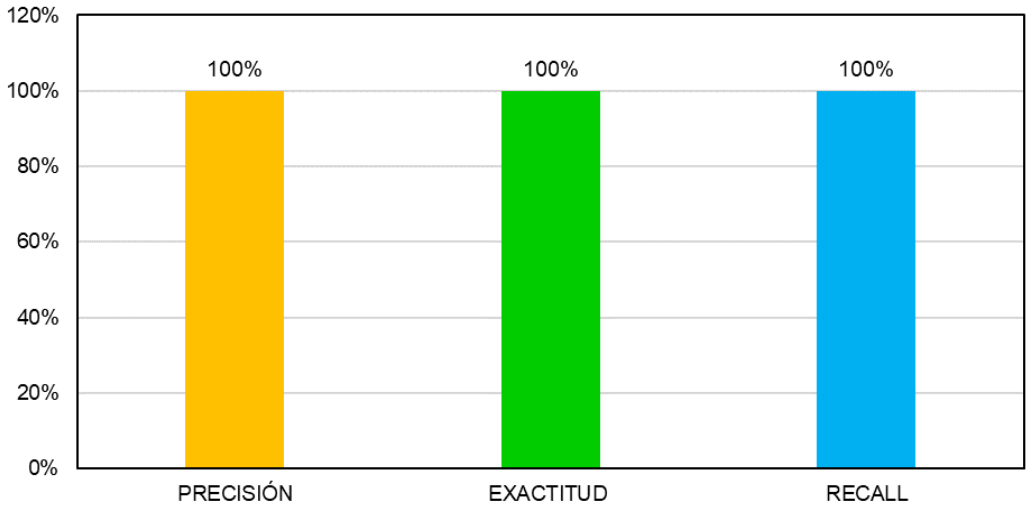

Figura 10 Métricas de rendimiento de audios de prueba del entrenamiento
(Fuente: Elaboración propia)

3.1.2. Métricas de ocultamiento

Del análisis de la base de datos, se obtuvo un valor de SNR para los audios originales y stego audios los cuales se muestran en la Figura 11, encontrándose que: el 92.6% de audios originales y 87.0% de stego audios tienen un SNR bajo del orden de 0.0125 (SNR promedio de 89.8%).

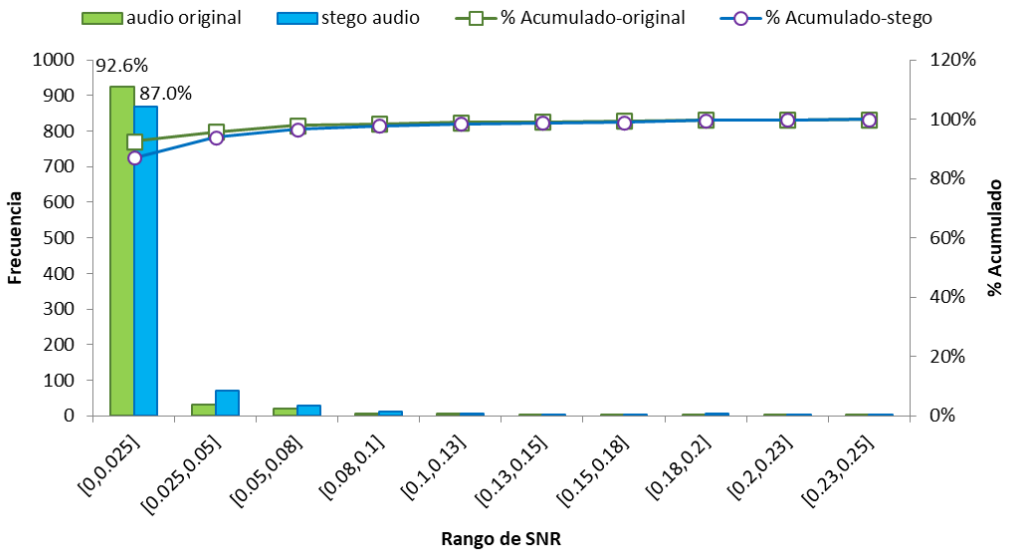

Figura 11. SNR de audios originales y stego audios
(Fuente: Elaboración propia)

Por otro lado, los valores de PSNR entre los audios originales y stego audios se muestran en la Figura 12, encontrándose que la mayoría, un 68.5% correspondiente a 685 audios, tienen un valor de PSNR de 72.25.

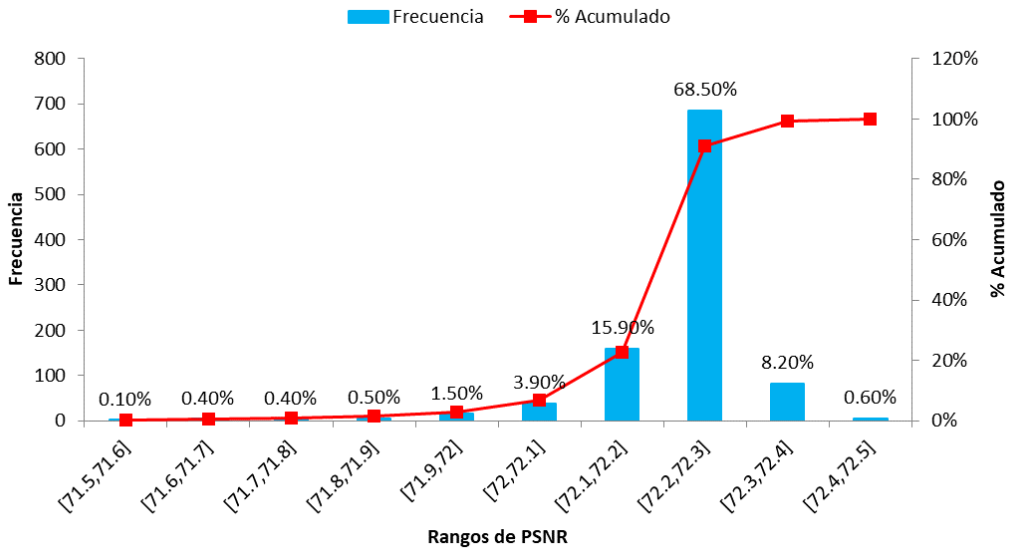

Figura 12. PSNR entre audio original y stego audio
(Fuente: Elaboración propia)

3.2. Discusión de Resultados

En términos de las métricas de rendimiento de la prueba de entrenamiento, la presente investigación obtuvo una precisión de 100%, exactitud de 100% y recall de 100%. Asimismo, tal como se muestra en el aporte práctico, usando una nueva base de datos, al usar el mismo modelo entrenado, los valores de las métricas de rendimiento fueron: precisión de 99.6%, exactitud de 99.8% y recall de 100%.

En comparación con la presente investigación, Ru, Zhang & Huang (2005) analizaron el algoritmo SVM para detectar contenido oculto en archivos de audios frente a métodos esteganográficos obteniendo los siguientes resultados: Para el método Hide4PGP se obtiene el 97.33% de exactitud, para el método Stegowav obtiene el 95.8% de exactitud y para el método S-Tool4 obtiene 98.09%.

Considerando que los autores Ru, Zhang & Huang (2005) utilizaron el algoritmo de clasificación SVM, obtuvieron en promedio una exactitud de 97.07% para diferentes métodos esteganográficos, la cual es ligeramente

menor en comparación con la presente investigación que se obtuvo 99.8% de exactitud frente al método esteganográfico LSB.

Asimismo, los autores Mohtasham & Mosleh (2019), usando el clasificador SVM para detección de mensajes ocultos en audios digitales, obtuvieron los siguientes resultados de las métricas de rendimiento entre ellas la precisión con 57.8% seguido de la exactitud 76.67% y finalmente recall de 73.80%.

De los resultados porcentuales de las métricas de rendimiento con el algoritmo de clasificación SVM de la presente investigación, frente al método esteganográfico LSB, se obtuvieron mejores resultados que los obtenidos por los autores Mohtasham & Mosleh (2019).

En términos de las métricas de ocultamiento, la presente investigación se obtuvo un valor máximo de PSNR de 71.57, un valor promedio de 72.22 y un valor mínimo de 72.46. Adicionalmente se puede resaltar que la desviación estándar de PSNR en la presente investigación fue de 0.089.

En contrastación con estos resultados, Sewisy et al. (2015) presenta un valor máximo de PSNR de 73.74, un valor promedio de 44.36, un valor mínimo de 12.22 y una desviación estándar de PSNR de 18.73.

Considerando que mientras mayor sea el valor de PSNR mejor es método de ocultamiento aplicado, por lo que se obtienen resultados similares en el valor máximo de PSNR, sin embargo, los resultados de la presente investigación muestran valores más estables del PSNR en comparación con los resultados de Sewisy et al. (2015), esto se puede evidenciar por el mejor valor de desviación estándar del PSNR.

Asimismo, en la investigación de Kar, Nakka & Katangur (2018), se obtuvieron un valor máximo de PSNR de 85.13, un valor promedio de 72.73, un valor mínimo de 61.41 y una desviación estándar de PSNR de 8.30.

Aunque el valor máximo de PSNR obtenido Kar, Nakka & Katangur (2018) es mayor que el obtenido en la presente investigación, se obtuvo un valor promedio de PSNR similar a lo obtenido por Kar, Nakka & Katangur (2018). Sin embargo, también se puede apreciar una mejor estabilidad de los valores

obtenidos en la presente investigación, al presentar una menor desviación estándar de PSNR.

Por otro lado, Rajput, Adhiya, & Patnaik (2017) encontraron un valor máximo de PSNR de 153.41, un valor promedio de 142.70, un valor mínimo de 131.53 y una desviación estándar de PSNR de 6.63.

En comparación con la presente investigación, Rajput, Adhiya, & Patnaik (2017) obtuvieron mejores valores de PSNR, indicando un mejor método de ocultamiento utilizado, sin embargo, los valores de PSNR obtenidos por la presente investigación cuentan con una menor desviación estándar, mostrando una mejor estabilidad del método LSB de ocultamiento utilizado.

3.3. Aporte práctico

3.3.1. Elaborar una base de datos

Para el aporte práctico, se generó una base de datos de audios digitales en formato WAV compuesto por 500 audios originales y 500 stego audios. En la Figura 13 se muestran los primeros 10 audios originales, los cuales corresponden a una fracción de un audio de música. Asimismo, se aprecia la duración y el tamaño de cada audio.

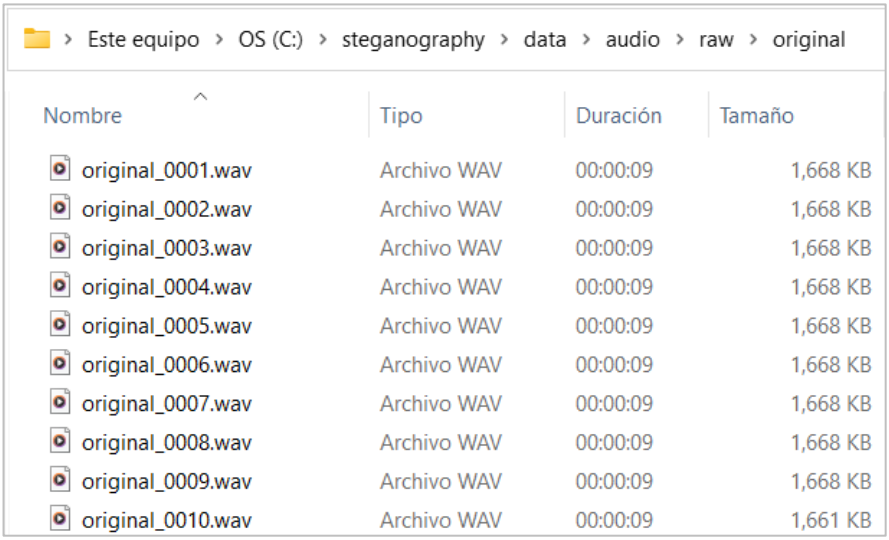

Figura 13. Base de datos de audios originales
(Fuente: Elaboración propia)

De los resultados obtenidos de los audios analizados, la Figura 14 muestra el histograma de los tiempos de duración de los audios originales y stego audios analizados. Asimismo, la Tabla 4 muestra la frecuencia y porcentaje acumulado para cada rango del histograma de tiempos de duración de cada audio.

Se puede apreciar que el 58.2% de los audios que componen la base de datos analizada, equivalente a 290 audios, tienen un tiempo promedio de duración de 11.06 segundos y una desviación estándar de 6.54.

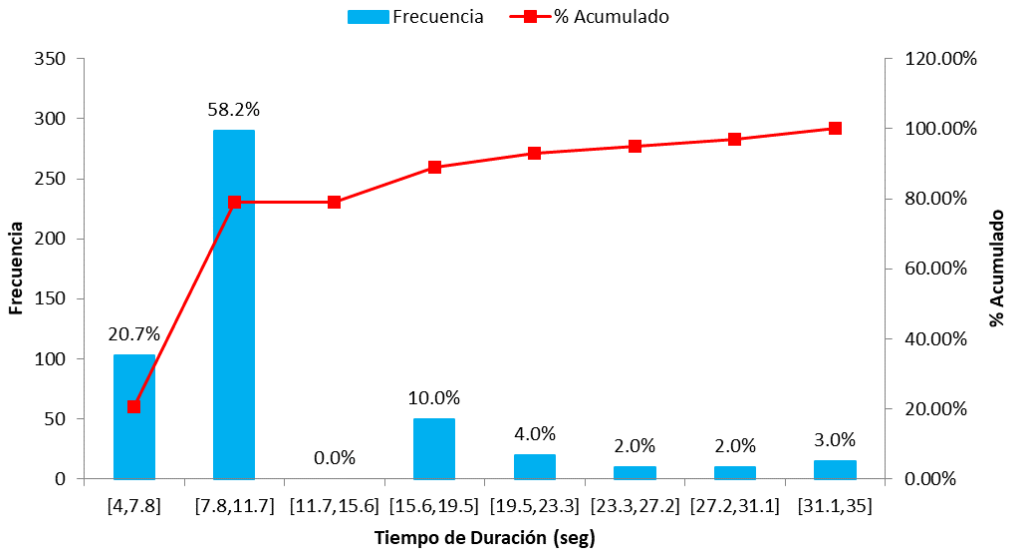

Figura 14. Histograma de tiempo de duración de audios

(Fuente: Elaboración propia)

Tabla 4.

Frecuencia y porcentaje acumulado de tiempos de duración de audios

Rango (seg)	Frecuencia	% Acumulado
[4,7.8]	103	20.72%
[7.8,11.7]	290	78.87%
[11.7,15.6]	0	78.87%
[15.6,19.5]	50	88.93%
[19.5,23.3]	20	92.96%
[23.3,27.2]	10	94.97%
[27.2,31.1]	10	96.98%
[31.1,35]	15	100.00%

(Fuente: Elaboración propia)

La base de datos de mensajes ocultos consta de 100 archivos de texto, en la Figura 15 se muestran los primeros 10 mensajes de texto, los cuales fueron usados como información oculta dentro de un archivo de audio original. Asimismo, la Tabla 5 muestra el contenido, cantidad de caracteres y tamaño de los 10 primeros archivos de texto.

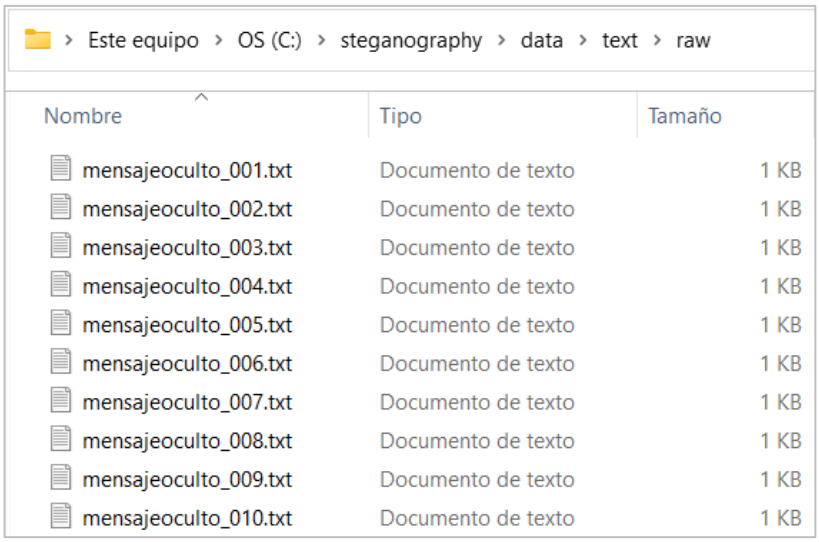

Figura 15. Base de datos de mensajes de texto

(Fuente: Elaboración propia)

Tabla 5.
Base de datos de archivos de texto

Archivo	**Contenido**	**Cantidad de Caracteres**	**Tamaño de Archivo**
Mensajeoculto_001.txt	NUMERO DE TARJETA: 4455 2255 6633 7788 CV: 151	46	1KB
mensajeoculto_002.txt	polo rojo pantalón verde.	25	1KB
mensajeoculto_003.txt	carro Toyota rojo polarizado D2Y321	35	1KB
mensajeoculto_004.txt	Puedo escribir los versos más tristes esta noche.	49	1KB
mensajeoculto_005.txt	Escribir, por ejemplo: La noche está estrellada, y tiritan, azules, los astros, a lo lejos.	91	1KB
mensajeoculto_006.txt	El viento de la noche gira en el cielo y canta. Puedo escribir los versos más tristes esta noche. Yo la quise, y a veces ella también me quiso. En las noches como ésta la tuve entre mis brazos. La besé tantas veces bajo el cielo infinito	237	1KB
mensajeoculto_007.txt	Somos libres, seámoslo siempre, y antes niegue sus luces el sol, que faltemos al voto solemne que la patria al Eterno elevó.	124	1KB
mensajeoculto_008.txt	En su cima los Andes sostengan la bandera o pendón bicolor, que a los siglos anuncie el esfuerzo que ser libres, por siempre nos dio.	259	1KB
mensajeoculto_009.txt	se nos fue, para la próxima.	28	1KB
mensajeoculto_010.txt	asfafadgasdgasdgasdgasdgasdgsadgasdg	37	1KB

(Fuente: Elaboración propia)

Los audios originales fueron procesados previo al ocultamiento del archivo de texto. El procesamiento de los archivos de audio digital consiste en la conversión del formato de audio original a int16, y en caso el audio tenga 2 canales (Estéreo) se convierte la señal a un solo canal (Mono). La Figura 16 muestra el fragmento del código fuente que se elaboró para el procesamiento del audio original, seguidamente en la Figura 17 se muestra los 10 primeros audios originales procesados.

```
    path = os.path.join(definitions.TEST_AUDIO,"original", "*.wav")
    audio_path = glob.glob(path)
    f_types = [np.float16, np.float32, np.float64]
    i_types = [np.int32, np.int64]
    for afp in audio_path:
        try:
            rate, data = read(afp)
        except Exception as e:
            print(f"Error reading: {afp}")
            print(e)
            try:
                data, rate = soundfile.read(afp)
            except Exception as e:
                print(f"Error reading with soundfile: {afp}")
                print(e)
                continue
        if data.dtype in f_types:
            data = util.float_to_int16(data)
        """if data.dtype in i_types:
            data = data.astype(np.int16)"""
        if data.ndim > 1:
            data = util.to_single_channel(data)
        head, tail = os.path.split(afp)
        filename, file_extension = os.path.splitext(tail)
        filename = filename.replace(" ", "_")
        cafp = os.path.join(definitions.TEST_AUDIO, "processed", filename +
    "_processed" + file_extension)
```

Figura 16. Fragmento de código fuente de procesamiento de audio.
(Fuente: Elaboración propia)

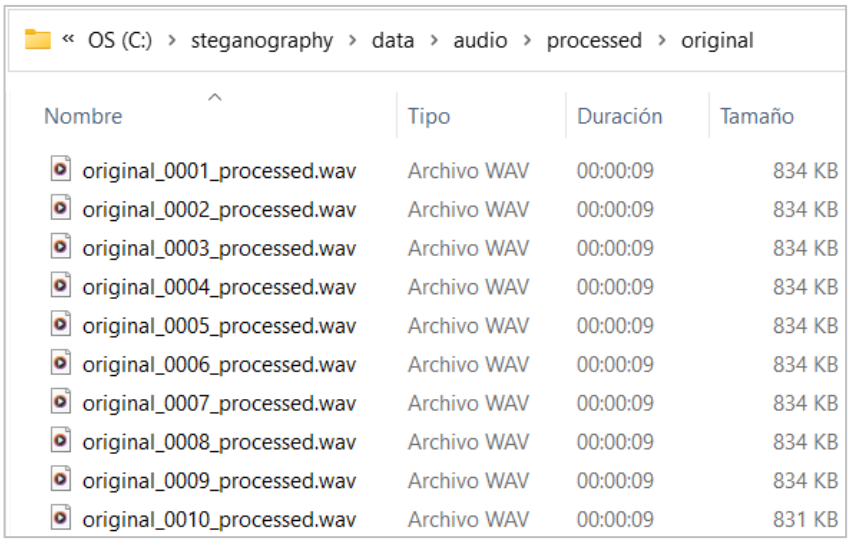

Figura 17. Base de datos de audios originales procesados

(Fuente: Elaboración propia)

3.3.2. Ocultar un mensaje de texto en los archivos de audio digital

Con los audios originales procesados, se procede a realizar el ocultamiento del mensaje de texto seleccionado de forma aleatoria de la base de datos, utilizando el método LSB, un fragmento del código de ocultamiento se muestra en la Figura 18. Como resultado del ocultamiento, se genera la misma cantidad de audios originales con información oculta (stego audio). La Figura 19 muestra los primeros 10 stego audios de la base de datos generada.

```
n = len(text_list)
for afp in audio_path:
    lsb_bytes = lsb.LSB_cover(afp, text_list[randrange(n)])
    head, tail = os.path.split(afp)
    filename, file_extension = os.path.splitext(tail)
    filename = filename.replace(" ", "_")
    cafp=os.path.join(definitions.TEST_AUDIO, "original_processed_stego",
filename + "_stego" + file_extension)
```

Figura 18. Fragmento de código fuente de ocultamiento de texto a audio.

(Fuente: Elaboración propia)

Finalmente, la base de datos de audios a ser analizada está compuesta por 500 audios originales procesados y 500 stego audios, es decir un total de 1000 audios, los cuales fueron utilizados para la detección de mensajes ocultos en archivos de audio digital en formato WAV, utilizando el modelo de predicción SVM entrenado.

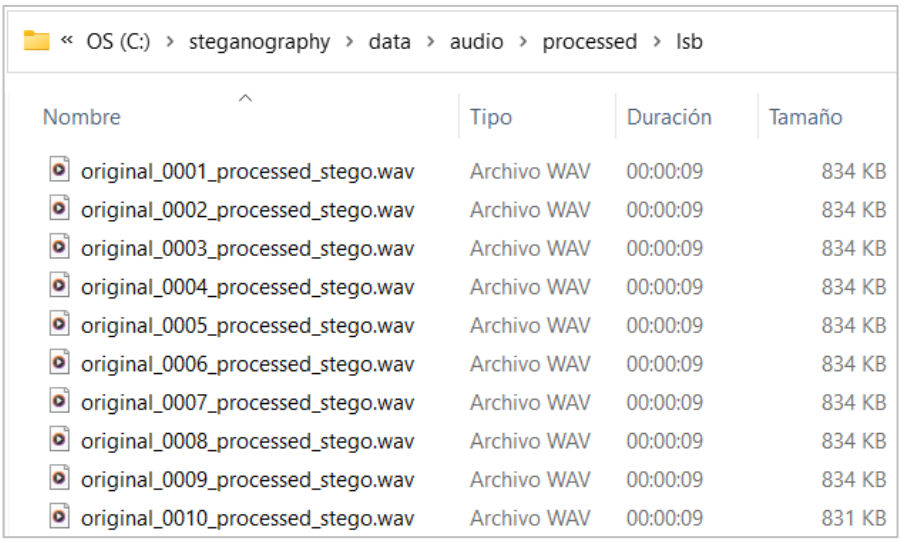

Figura 19. Base de datos de stego audios

(Fuente: Elaboración propia)

3.3.3. Implementar un algoritmo de detección

La implementación del algoritmo de detección consistió en el entrenamiento de un modelo de Machine Learning basado en el método SVM. El diagrama de flujo de entrenamiento y prueba usando el método SVM para generar un modelo de detección es mostrado en la Figura 20.

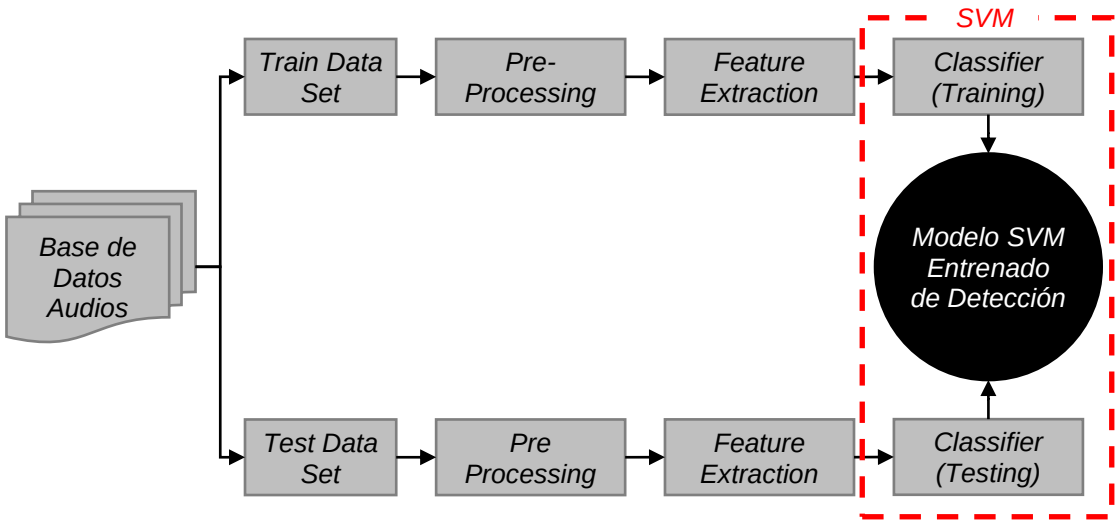

Figura 20. Diagrama de flujo del modelo de detección

(Fuente: Elaboración propia)

Cabe señalar que los resultados del entrenamiento y prueba del modelo de detección, se muestran en el item 3.1. Para el entrenamiento y prueba del modelo de detección se utilizó una base de datos de 2000 audios digitales en formato WAV, compuesta por 1000 audios originales y 1000 stego audios.

En la Figura 21 se muestra un fragmento del código fuente del algoritmo de detección.

```
    for afp in audio_path_list:
        start_time = time.time()
        row = get_feat(afp)
        feat_dur.append(time.time() - start_time)
        sec.append(get_audio_seconds(afp))
        X.append(row)
    return {"X": X, "y": [label]*len(X), "sec": sec, "feat_dur": feat_dur}
def make_prediction(dataset_dict, model):
    X = np.array(dataset_dict["X"])
    pred_dur = []
    pred = []
    for i in range(X.shape[0]):
        start_time = time.time()
        y = int(model.predict(X[i,:].reshape((1,4))))
        pred_dur.append(time.time() - start_time)
        pred.append(y)
    dataset_dict["pred"] =  pred
    dataset_dict["pred_dur"] = pred_dur
    return dataset_dict
proc_dataset = build_dataset(proc_audio_path, 0)
stego_dataset = build_dataset(stego_audio_path, 1)
proc_dataset = make_prediction(proc_dataset, clf)
stego_dataset = make_prediction(stego_dataset, clf)
y_test = proc_dataset["y"] + stego_dataset["y"]
y_pred = proc_dataset["pred"] + stego_dataset["pred"]
```

Figura 21. Fragmento de código fuente de algoritmo de detección.
(Fuente: Elaboración propia)

3.3.4. Realizar simulaciones de detección y análisis de resultados

Tal como se mencionó en el ítem 3.3.1, para la realización de simulaciones de detección, se generó una base de datos de 1000 audios digitales en formato WAV, para la aplicación práctica del modelo desarrollado, cabe resaltar que dichos audios no fueron analizados previamente por el modelo ya entrenado. La base de datos generada para la aplicación práctica está compuesta por 500 audios originales y 500 stego audios.

De los resultados obtenidos de los audios analizados, la Figura 22 muestra el histograma de los tiempos de ejecución de los audios originales y stego audios analizados. Asimismo, la Tabla 6 muestra la frecuencia y porcentaje acumulado para cada rango del histograma de tiempos de ejecución de audios.

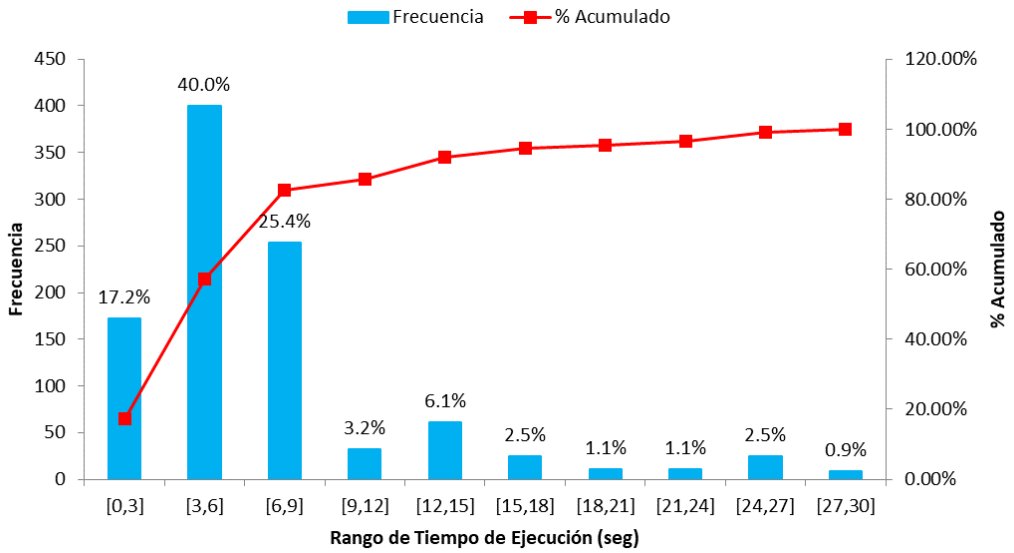

Figura 22. Histograma de tiempo de ejecución

(Fuente: Elaboración propia)

Tabla 6.
Frecuencia y porcentaje acumulado de tiempos de ejecución

Rango (seg)	Frecuencia	% Acumulado
[0,3]	172	17.20%
[3,6]	400	57.20%
[6,9]	254	82.60%
[9,12]	32	85.80%
[12,15]	61	91.90%
[15,18]	25	94.40%
[18,21]	11	95.50%
[21,24]	11	96.60%
[24,27]	25	99.10%
[27,30]	9	100.00%

(Fuente: Elaboración propia)

En términos del consumo del CPU Figura 23 muestra el histograma del consumo de CPU durante el procesamiento de los audios originales y stego audios analizados. De tal manera que, el 83.7% de audios analizados tienen un consumo de CPU entre 90% y 100%. Asimismo, la Tabla 7 muestra la frecuencia y porcentaje acumulado para cada rango del histograma de consumo de CPU durante el análisis de los audios.

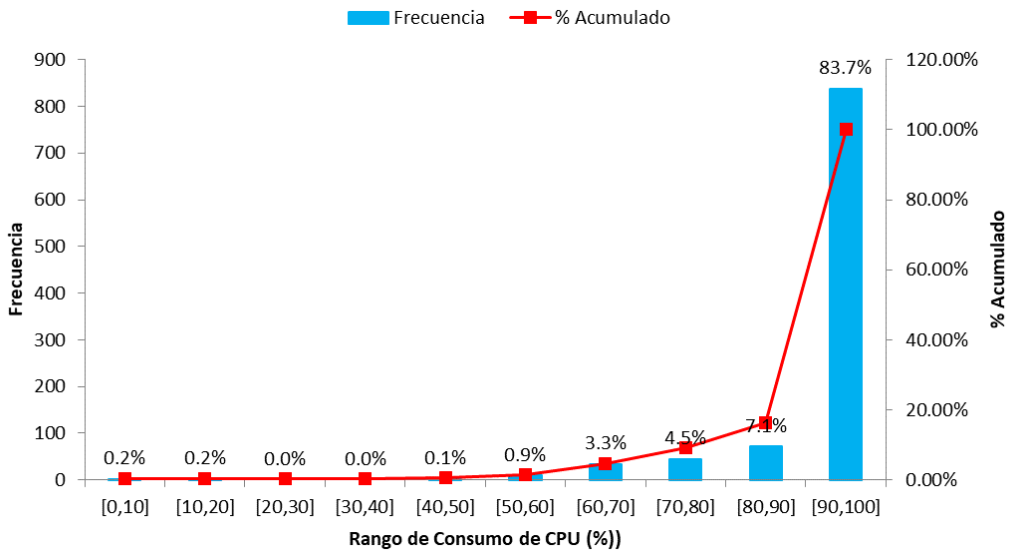

Figura 23. Histograma de consumo de CPU

(Fuente: Elaboración propia)

Tabla 7.
Frecuencia y porcentaje acumulado del consumo de CPU

Rango (seg)	Frecuencia	% Acumulado
[0,10]	2	0.20%
[10,20]	2	0.40%
[20,30]	0	0.40%
[30,40]	0	0.40%
[40,50]	1	0.50%
[50,60]	9	1.40%
[60,70]	33	4.70%
[70,80]	45	9.20%
[80,90]	71	16.30%
[90,100]	837	100.00%

(Fuente: Elaboración propia)

De mismo modo, en términos del consumo de memoria RAM, la Figura 24 muestra el histograma del consumo de memoria RAM durante el procesamiento de los audios originales y stego audios analizados. De tal manera que, el 78.4% de los audios analizados tienen un consumo de memoria RAM de 48MB a 54MB. Asimismo, la Tabla 8 muestra la frecuencia y porcentaje acumulado para cada rango del histograma de consumo de RAM durante el análisis de los audios.

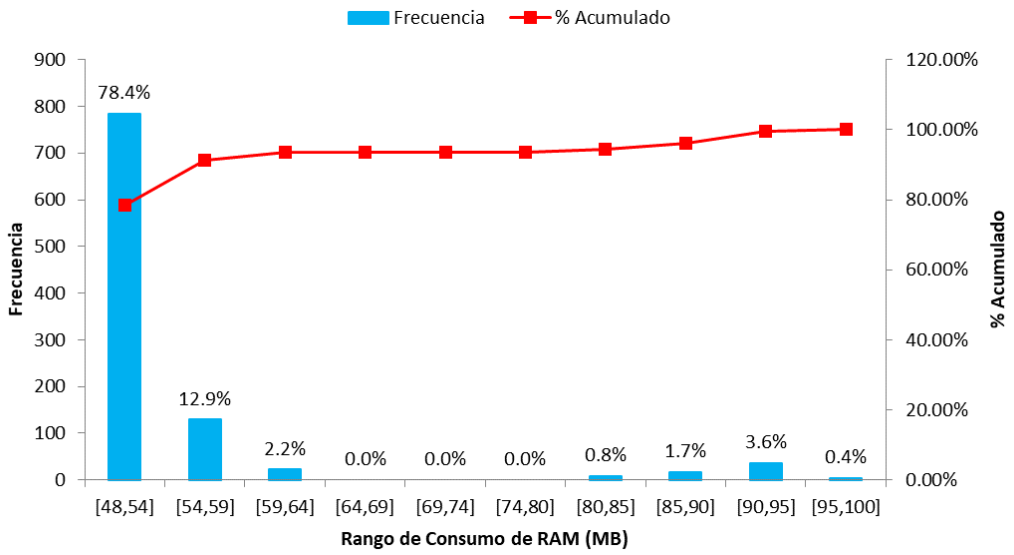

Figura 24 Consumo de RAM de Audios

(Fuente: Elaboración propia)

Tabla 8.
Frecuencia y porcentaje acumulado del consumo de RAM

Rango (seg)	Frecuencia	% Acumulado
[48,54]	784	78.40%
[54,59]	129	91.30%
[59,64]	22	93.50%
[64,69]	0	93.50%
[69,74]	0	93.50%
[74,80]	0	93.50%
[80,85]	8	94.30%
[85,90]	17	96.00%
[90,95]	36	99.60%
[95,100]	4	100.00%

(Fuente: Elaboración propia)

Por otro lado, para la obtención de las métricas de la detección de los audios de la base de datos, compuesta por 500 audios originales y 500 stego audios, se generó el código mostrado en la Figura 25.

De los resultados del análisis de detección de los audios de la base de datos analizada, se obtuvo la matriz de confusión mostrada en la Tabla 9.

```
cm = confusion_matrix(y_test, y_pred)
print(f"Confusion Matrix: \n{cm}")

acc = accuracy_score(y_test, y_pred)
print(f"Accuracy model: {acc}")

pre = precision_score(y_test, y_pred)
print(f"Precision Score: {pre}")

rec = recall_score(y_test, y_pred)
print(f"Recall Score: {rec}")
```

Figura 25. Fragmento de código fuente para las métricas de rendimiento
(Fuente: Elaboración propia)

Tabla 9.
Matriz de confusión de la prueba

		PREDICCIÓN	
		AUDIO ORIGINAL	STEGO AUDIO
REAL	AUDIO ORIGINAL	498	2
	STEGO AUDIO	0	500

(Fuente: Elaboración propia)

VN - VERDADERO NEGATIVO : **498** Audios originales detectados como originales.

VP - VERDADERO POSITIVO : **500** Stego audios detectados como Stego audios

FN - FALSO NEGATIVO : **0**

FP - FALSO POSITIVO : **2** Audios originales detectados como Stego Audios

$$E = \frac{VP + VN}{VP + VN + FP + FN} = \frac{500 + 498}{500 + 498 + 2 + 0} = 0.998 = 99.8\%$$

$$P = \frac{VP}{VP + FP} = \frac{500}{500 + 2} = 0.996 = 99.6\%$$

$$R = \frac{VP}{VP + FN} = \frac{500}{500 + 0} = 1 = 100\%$$

De los valores obtenidos en la matriz de confusión para los audios de la base de datos analizada, en la Figura 26 se observan las métricas de rendimiento de la detección de los 1000 audios digitales en formato WAV, en el cual se observa un 99.6% de Precisión, 99.8% de Exactitud y 100.0% de Recall.

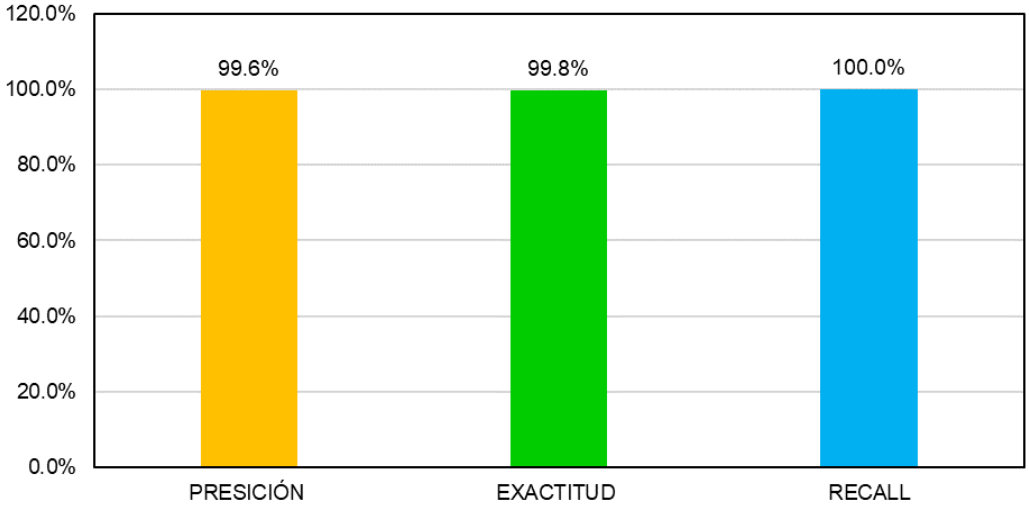

Figura 26 Métricas de Rendimiento de Audios
(Fuente: Elaboración propia)

Del procesamiento de los resultados de la base de datos analizada, utilizando el modelo SVM entrenado, se obtuvo una correlación entre la duración de los audios de componen la base de datos y los tiempos de detección de la existencia o no de información oculta en el audio analizado, independientemente si el audio es original o stego audio. Se puede observar que existe una tendencia lineal entre el tiempo de detección y la duración del audio analizado tal como se aprecia en la Figura 27.

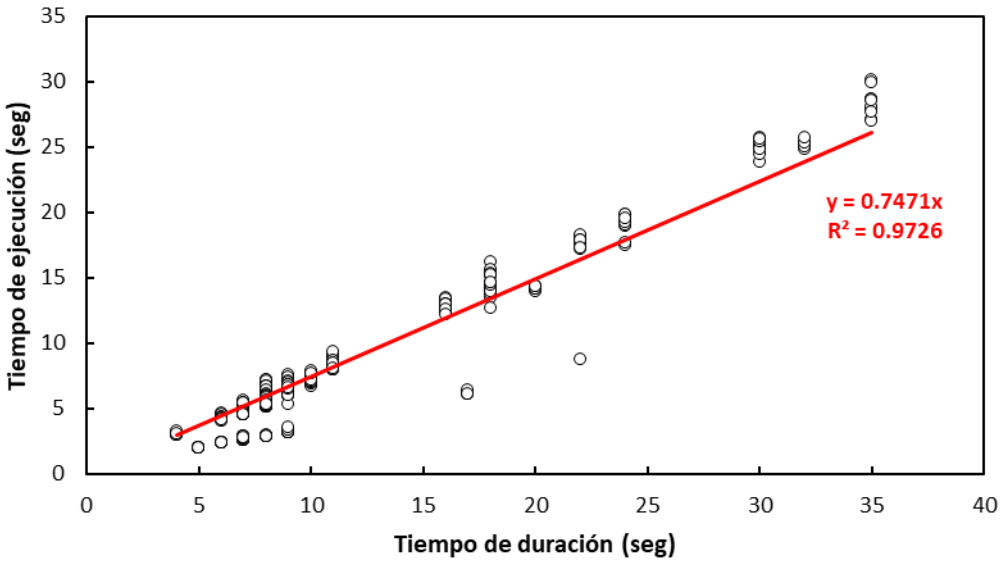

Figura 27. Tiempo de ejecución vs. Duración de audio
(Fuente: Elaboración propia)

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Para la presente investigación se realizó una base de datos con diferentes audios de música en formato WAV, entre audios originales y stego audios que son los audios originales con mensaje de texto.

Asimismo, para el desarrollo de esta investigación se estudió el método de ocultamiento LSB logrando ocultar mensaje de texto en archivo de audio WAV. Sin embargo, se analizó de manera audible la diferencia entre el audio original y el stego audio, encontrando un ruido de fondo, el cual evidencia la existencia de un mensaje oculto. Se recomienda estudiar más a profundidad el método de ocultamiento a fin de que el stego audio no presente ruidos perceptibles al oído humano.

En el marco de esta investigación, se implementó un modelo de aprendizaje automático para la detección de archivos de audio digital con mensaje oculto, basado en el método Support Vector Machine (SVM), con el cual se entrenó un modelo para la clasificación de archivos de audio digital. Para el entrenamiento se usó el 65% de la base de datos y para la prueba del entrenamiento el 35% restante.

Asimismo, con los resultados obtenidos el algoritmo SVM entrenado, proporciona un 99.6% precisión, 99.8% de exactitud y 100.0% de recall para la detección de mensajes ocultos en audios digitales en formato WAV, los cuales se consideran consistentes en comparación con investigaciones similares revisadas. Es importante mencionar que las métricas obtenidas en el aporte practico corresponden a la segunda base de datos utilizada, la cual es diferente de la base de datos con la que se entrenó el modelo.

Cabe resaltar, que el tiempo promedio de detección de audios fue de 7.9 segundos, sin embargo, la variación en el tiempo de detección de los audios analizados depende de la capacidad del equipo de cómputo usado, del tamaño y duración de los archivos de audio y del método de ocultamiento utilizado.

Para la realización de la presente investigación se utilizó un equipo de cómputo portátil (Laptop) con las siguientes características: Sistema Operativo Windows 11 Home 64 bits, Procesador i7-11800H @ 2.3Ghz (16 CPUs), Memoria RAM de 16GB y Disco Duro Sólido de 500 GB. Adicionalmente, para el desarrollo del programa, se utilizó el lenguaje de programación Python v3.9 en la plataforma Spyder v5.2.2. Las principales librerías utilizadas fueron Numpy, Soundfile, Sklearn, Scipy.

4.2. Recomendaciones

Para el análisis del ocultamiento de mensajes ocultos en archivos de audio digital, se consideró el uso de audios en formato WAV, se recomienda tomar en cuenta la duración de los audios para generar la base de datos ya que es directamente proporcional al tiempo de detección de los audios, lo cual conllevaría a un mayor consumo de los recursos usando el modelo de detección desarrollado.

A pesar de la presencia de un ruido audible en los audios con mensaje oculto, con la utilización del método LSB para el ocultamiento de información en archivos de audios digital en formato WAV. Las métricas de rendimiento fueron satisfactorias. Sin embargo, se recomienda extender la línea de investigación considerando diferentes métodos de ocultamiento para mejorar eliminar o reducir el ruido audible.

Los tiempos de ejecución obtenidos dependen directamente de la duración del audio analizado, este tiempo considera desde el preprocesamiento del audio, la extracción de características y la detección de la existencia de contenido oculto. Se recomienda implementar un algoritmo de optimización a fin de reducir el tiempo de ejecución del algoritmo implementado.

REFERENCIAS

Aguirre, F. (2017). *Desarrollo y análisis de clasificadores de señales de audio* [Tesis de Maestría, Universidad Politécnica de Valencia]
Recuperado de: https://riunet.upv.es/handle/10251/90005

Ahmad, T., Amrizal, M., Wibisono, W., & Ijtihadie, R. (2020). Hiding data in audio files: A smoothing-based approach to improve the quality of the stego audio. Heliyon, 6(3).
Recuperado de: https://doi.org/10.1016/j.heliyon.2020.e03464

Ali, A., Mokhtar, M. & George, L. (2017). Enhancing the Hiding Capacity of Audio Steganography Based on Block Mapping. *Journal of Theoretical & Applied Information Technology*, 95(7), 1441-1448. Recuperado de: https://www.researchgate.net/publication/316271884_Enhancing_the_hiding_capacity_of_audio_steganography_based_on_block_mapping

Angulo, C, Ocampo, S., & Blandon, L. (2007). Una mirada a la esteganografía. Scientia et technica, 13(37), 421-426.
Recuperado de: https://www.redalyc.org/pdf/849/84903772.pdf

Azam, M., Ridzuan, F., Sayuti, M., & Alsabhany, A. (2019). Balancing the Trade-Off Between Capacity and Imperceptibility for Least Significant Bit Audio Steganography Method: A New Parameter. *In 2019 IEEE Conference on Application, Information and Network Security (AINS)* (pp. 48-53).
Recuperado de: https://ieeexplore.ieee.org/document/8968707

Behar, D. (2008). *Metodología de la Investigación*. Editorial Shalom.
ISBN 978-959-212-783-7

Calvillo, S. (2005), análisis comparativo de los formatos de sonido digital: MP3, OGG VORBIS y YAMAHA VQF [Tesis de Grado, universidad de san Carlos de Guatemala]
Recuperado de: http://biblioteca.usac.edu.gt/tesis/08/08_0284_CS.pdf

Chávez, P & Gutiérrez, G (2020). Ocultamiento de Información Confidencial en Imágenes BMP y audio WAV mediante el Método LSB. *Revista de la Facultad de Contaduría y Ciencias Administrativas*, 5(10), 92-99.
Recuperado de: https://rfcca.umich.mx/index.php/rfcca/article/view/149

Fateh, M., Rezvani, M., & Irani, Y. (2021). A new method of coding for steganography based on LSB matching revisited. *Security and Communication Networks, 2021*.
Recuperado de: https://doi.org/10.1155/2021/6610678

Francés M., T. (2020). *Impacto del machine learning en el sistema financiero*. [Tesis de Grado, Universidad Pontificia Comillas]
Recuperado de: https://repositorio.comillas.edu/xmlui/handle/11531/42692

Hernández, R., Fernández, C. & Baptista, P. (2014). *Metodología de la investigación*. (6ª ed.). México. McGraw Hill.
ISBN: 978-1-4562-2396-0

Hernández-Sampieri, R., & Mendoza, C. (2018). *Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta*. México. McGrawHill.
ISBN: 978-1-4562-6096-5

James, A. (2021). *12 VoIP Trends for 2021/2022: Latest Predictions To Watch Out For*. FinancesOnline.
Recuperado de: https://financesonline.com/voip-trends

Jung, K. H. (2019). A study on machine learning for steganalysis. *In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing* (pp. 12-15).
Recuperado de: https://doi.org/10.1145/3310986.3311000

Kar, D. C., Nakka, A. M., & Katangur, A. K. (2018). A new statistical attack resilient steganography scheme for hiding messages in audio files. *International Journal of Information and Computer Security*, 10(2-3), 276-302.
Recuperado de: https://doi.org/10.1504/IJICS.2018.091472

Lucena, M. (2010). *Criptografía y Seguridad*.
Recuperado de: http://seguridad.unicauca.edu.co/criptografia/cripto.pdf

Mohtasham, V. & Mosleh, M. (2019). Audio Steganalysis based on collaboration of fractal dimensions and convolutional neural networks. *Multimedia Tools and Applications*, *78*(9), 11369-11386.
Recuperado de https://doi.org/10.1007/s11042-018-6702-1

Msallam, M. M. (2020). A Development of Least Significant Bit Steganography Technique. *IRAQI Journal of Computers, Communications, Control and Systems Engineering*, 20(1), 31-39.
Recuperado de: https://ijccce.uotechnology.edu.iq/article_168047.html

Nassar, S. S., Faragallah, O. S., & El-Bendary, M. A. (2021). Reliable Mark-Embedded Algorithm for Verifying Archived/Encrypted Image Contents in Presence Different Attacks with FEC Utilizing Consideration. *Wireless Personal Communications*, 1-25.
Recuperado de: https://doi.org/10.1007/s11277-021-08176-x

Nehru, G., & Dhar, P. (2012). A detailed look of audio steganography techniques using LSB and genetic algorithm approach. *International Journal of Computer Science Issues (IJCSI)*, 9(1), 402. Recuperado de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.6018&rep=rep1&type=pdf

Patel, R., Lad, K. & Patel, M. (2021). Study and investigation of video steganography over uncompressed and compressed domain: a comprehensive review. *Multimedia Systems*, 1-40.
Recuperado de: https://doi.org/10.1007/s00530-021-00763-z

Rajput, S. P., Adhiya, K. P., & Patnaik, G. K. (2017). An efficient audio steganography technique to hide text in audio. *In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)* (pp. 1-6). IEEE.
Recuperado de: https://ieeexplore.ieee.org/abstract/document/8463948

Rupa Ch., Shaikh, S., & Chinta, M. (2021). Multimedia Concealed Data Detection Using Quantitative Steganalysis. *International Journal of Digital Crime and Forensics (IJDCF)*, 13(5), 101-113.
Recuperado de: http://doi.org/10.4018/IJDCF.20210901.oa6

Sewisy, A., Mansour, R., Rida, S. & Mohammed, A. (2015). Hidden Text into Audio Files. *International Journal of Research Studies in Science, Engineering and Technology*, 2(5), 33-39.
Recuperado de: http://www.ijrsset.org/pdfs/v2-i5/5.pdf

Singh, A. P., Moudgil, S., & Rani, S. (2021). An Acquaintance to Text-Steganography and its Methods. In *Journal of Physics: Conference Series* (Vol. 1950, No. 1, p. 012005). IOP Publishing. Recuperado de: https://iopscience.iop.org/article/10.1088/1742-6596/1950/1/012005/meta

Solutions, M. (2018). Machine Learning, una pieza clave en la transformación de los modelos de negocio. *Management Solutions - España*. Recuperado de: https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf

Wendzel, S., Mazurczyk, W., Caviglione, L., & Meier, M. (2014). Hidden and uncontrolled–on the emergence of network steganographic threats. *In ISSE 2014 Securing Electronic Business Processes* (pp. 123-133). Springer Vieweg, Wiesbaden. Recuperado de: https://link.springer.com/chapter/10.1007/978-3-658-06708-3_9

Ru, X. M., Zhang, H. J. & Huang, X. (2005) Steganalysis of audio: attacking the Steghide. *International Conference on Machine Learning and Cybernetics*, 2005, pp. 3937-3942 Vol. 7.
Recuperado de doi:10.1109/icmlc.2005.1527626

Zhang, J., Du, F., & Li, S. (2013). Steganalysis of LSB Matching in WAV Audio. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*,1951-6851.
Recuperado de: https://doi.org/10.2991/iccsee.2013.258

ANEXOS

Anexo 1. Resolución de Proyecto de Tesis

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1000-2021/FIAU-USS

Pimentel, 11 de noviembre de 2021

VISTOS:
El Acta de reunión N°0610-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante Oficio N°0359-2021/FIAU-IS-USS de fecha 14 de octubre de 2021, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, según documentos de vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerda aprobar la modificación de los temas de Tesis a cargo de los estudiantes que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: MODIFICAR, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de **INGENIERÍA DE SISTEMAS** según se detalla en el anexo de la presente Resolución.

ARTÍCULO 2°: MODIFICAR, la Resolución de Facultad con la que se asigna Asesor especialista y/o Jurado evaluador en el extremo del tema de la tesis quedando tal como se detalla en el anexo de la presente Resolución.

ARTÍCULO 3°: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Mg. Victor Alexei Tuesta Montero
Decano (e) / Facultad De Ingeniería, Arquitectura Y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

MBA. María Noelia Sialer Rivera
Secretaria Académica / Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1000-2021/FIAU-USS

Pimentel, 11 de noviembre de 2021

ANEXO

N°	AUTOR(ES)	TEMA DE TESIS ACTUAL	TEMA DE TESIS ANTERIOR	RESOLUCIÓN PREVIA
1	SALAZAR AGUILAR LUIS	DETECCIÓN AUTOMÁTICA DE MENSAJE DE TEXTO OCULTO EN UN ARCHIVO DE AUDIO	EVALUACIÓN DE LOS ALGORITMOS HMM Y DTW PARA MEDIR LA EFICIENCIA EN EL TRÁFICO OCULTO DE VOZ IP	0672-2021/FIAU-USS
2	CARRERA SANCHEZ JOSE ANTONIO	DESARROLLO DE UN MODELO DE CIBERSEGURIDAD BASADO EN EL MARCO NIST V1.1. PARA LA DEFENSA CONTRA ATAQUES CIBERNÉTICOS A UNA UNIVERSIDAD NACIONAL PERUANA	EVALUACIÓN DE MARCOS DE TRABAJO PHP PARA EL DESARROLLO DE APLICACIONES WEB, BAJO LA NORMA ISO/IEC 25010, ENFOCADA A LA CALIDAD EN USO DEL PRODUCTO	0451-2021/FIAU-USS
3	THEOLOGITIS SANCHEZ DIMITRIS IOANNIS	PREDICCIÓN DE LA DEMANDA DE PRODUCTOS PARA PYMES DEDICADAS AL NEGOCIO RETAIL UTILIZANDO REDES NEURONALES ARTIFICIALES	PREDICCIÓN DE LA DEMANDA PARA EL ABASTECIMIENTO DEL INVENTARIO BASADO EN REDES NEURONALES ARTIFICIALES PARA PYMES DEDICADAS AL NEGOCIO RETAIL	0449-2021/FIAU-USS
4	ALFARO YESQUEN LILIANA ELIZABETH	IDENTIFICACIÓN AUTOMÁTICA DE INTENSIDAD DE CLOROFILA EN PLANTAS DE CAPSICUM ANNUUM GROUP MEDIENTE EL PROCESAMIENTO DE IMÁGENES DIGITALES	IDENTIFICACIÓN AUTOMÁTICA DE ESTRÉS HÍDRICO EN PLANTAS DE CAPSICUM ANNUUM GROUP MEDIANTE EL PROCESAMIENTO DE IMÁGENES DIGITALES	1818-2019/FIAU-USS
5	CIEZA CELIS JESUS ABELARDO OJEDA ROMERO ANTHONNY JHONATAN	EVALUACIÓN DEL DESEMPEÑO DE PROTOCOLOS DE SEGURIDAD PARA COMBATIR ATAQUES EN REDES INALÁMBRICAS WI-FI	EVALUACIÓN DEL DESEMPEÑO DE PROTOCOLOS DE SEGURIDAD DE REDES PARA COMBATIR VULNERABILIDADES EN REDES INALÁMBRICAS WI-FI	0700-2021/FIAU-USS
6	TAPIA LLATAS MANUEL AURELIO	COMPARACIÓN DE TÉCNICAS DE CLASIFICACIÓN AUTOMÁTICA PARA LA IDENTIFICACIÓN EFECTIVA DE MALWARE	COMPARACIÓN DE TÉCNICAS DE SISTEMAS INMUNES ARTIFICIALES EN LA IDENTIFICACIÓN DE MALWARE	2320-2020/FIAU
7	CALDERON YNOÑAN PAMELA DEL CARMEN PRIETO NEIRA FRANCK ALBERSON	INTEGRACIÓN DE UN ENFOQUE ÁGIL CON TÉCNICAS DE DISEÑO CENTRADO EN USUARIO (DCU) PARA LA MEJORA DE EXPERIENCIA DE USUARIO (UX)	DESARROLLO DE UN MÉTODO BAJO EL ENFOQUE ÁGIL EN ENTORNOS DE EXPERIENCIA DE USUARIO UI/UX PARA ASEGURAR LA USABILIDAD WEB	0445-2021/FIAU-USS
8	GUEVARA PEREZ ALEX HUMBERTO	COMPARACIÓN DE ALGORITMOS DE SEGMENTACIÓN DE IMÁGENES DIGITALES DE LAS HOJAS DE CAPSICUM ANNUUM ADQUIRIDAS EN AMBIENTE NO CONTROLADO	COMPARACIÓN DE ALGORITMOS DE SEGMENTACION PARA LA DETECCIÓN DE ENFERMEDAD OIDIOPSIS EN AMBIENTES NO CONTROLADOS EN CAPSICUM ANNUUM GROUP MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES	1823-2019/FIAU-USS
9	CASTRO MEDINA MIGUEL ANGEL	IMPLEMENTACIÓN DE UN MODELO AD HOC DE GESTIÓN DE SEGURIDAD DE LA	IMPLEMENTACIÓN DE UN MODELO AD HOC DE GESTIÓN DE LA SEGURIDAD	0700-2021/FIAU-USS

Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Anexo 2. Formato de matriz de confusión y métricas de rendimiento

		PREDICCIÓN	
		ORIGINAL	STEGO
REAL	ORIGINAL		
	STEGO		

ITEM	VALOR
VERDADERO POSITIVO (VP)	
FALSO POSITIVO (FP)	
VERDADERONEGATIVO (VN)	
FALSO NEGATIVO (FN)	

ITEM	VALOR
EXACTITUD	
PRECISIÓN	
RECALL	

VERDADERO POSITIVO (VP)		FALSO POSITIVO (FP)	
ITEM	VALOR	ITEM	VALOR
REALIDAD		REALIDAD	
PREDICCIÓN		PREDICCIÓN	
NUMERO DE RESULTADOS		NUMERO DE RESULTADOS	

VERDADERONEGATIVO (VN)		FALSO NEGATIVO (FN)	
ITEM	VALOR	ITEM	VALOR
REALIDAD		REALIDAD	
PREDICCIÓN		PREDICCIÓN	
NUMERO DE RESULTADOS		NUMERO DE RESULTADOS	