



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**“DESARROLLO Y ANÁLISIS DE SISTEMAS DE ESTIMACIÓN Y
DETECCIÓN DE OBJETIVOS DE RADAR MEDIANTE ALGORITMOS DE
MACHINE Y DEEP LEARNING”**

**AUTORES: AGUIRRE SILVA, HERNÁN GUILLERMO
GALLEGOS MOYA, SANTIAGO ANDRÉS**

DIRECTOR: ING. CARRERA ERAZO, ENRIQUE VINICIO

SANGOLQUÍ

2019

CERTIFICACIÓN



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"Desarrollo y análisis de sistemas de estimación y detección de objetivos de radar mediante algoritmos de Machine y Deep Learning"** realizado por los señores **Aguirre Silva, Hernán Guillermo y Gallegos Moya, Santiago Andrés**, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, junio 2019

Firma:



Ing. Enrique Vinicio Carrera Erazo

C. C: 1708792104

AUTORÍA DE RESPONSABILIDAD

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORÍA DE RESPONSABILIDAD

Yo, *Aguirre Silva, Hernán Guillermo*, declaro que el contenido, ideas y criterios del trabajo de titulación: **"Desarrollo y análisis de sistemas de estimación y detección de objetivos de radar mediante algoritmos de Machine y Deep Learning"** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas. Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, junio 2019

Firma:

Hernán Guillermo Aguirre Silva

C.C.: 1718131277

AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORÍA DE RESPONSABILIDAD

Yo, **Gallegos Moya, Santiago Andrés**, declaro que el contenido, ideas y criterios del trabajo de titulación: "**Desarrollo y análisis de sistemas de estimación y detección de objetivos de radar mediante algoritmos de Machine y Deep Learning**" es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas. Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, junio 2019

Firma:

Santiago Andrés Gallegos Moya

C.C.: 1804283685

AUTORIZACIÓN



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Aguirre Silva, Hernán Guillermo** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "**Desarrollo y análisis de sistemas de estimación y detección de objetivos de radar mediante algoritmos de Machine y Deep Learning**", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, junio 2019

Firma

Hernán Guillermo Aguirre Silva

C.C.: 1718131277

AUTORIZACIÓN



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, *Gallegos Moya, Santiago Andrés* autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "Desarrollo y análisis de sistemas de estimación y detección de objetivos de radar mediante algoritmos de Machine y Deep Learning", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, junio 2019

Firma

Santiago Andrés Gallegos Moya

C.C: 1804283685

DEDICATORIA

El presente trabajo lo dedico a todas aquellas personas que aportaron con un granito de arena de forma directa e indirecta en mi vida en la construcción y formación de mi ser. A quienes me apoyaron día tras día en los momentos felices y amargos para poder superar mis inquietudes y problemas. A todos aquellos que forman parte de mi vida y mi familia.

Hernán Guillermo Aguirre Silva

El presente trabajo es el fruto de años de trabajo en mi vida universitaria y requirió de mucho esfuerzo y dedicación y fue posible gracias a la ayuda de todas las personas que desinteresadamente me brindaron su apoyo durante toda esta travesía, dedico este logro a todos aquellos que fueron un pilar para que yo pueda seguir adelante y encontrarme en este punto de mi vida.

Santiago Andrés Gallegos Moya

AGRADECIMIENTOS

A mis padres y hermana que siempre estuvieron allí en las buenas y en las malas, apoyándome a tomar las mejores decisiones de mi vida.

Al amor de mi vida que junto a ella he superado muchas barreras que me potencializó como persona.

A mis amigos que de manera incondicional me sostuvieron para generar los resultados que tengo hoy por hoy.

A mi compañero Santiago Gallegos, por ser y estar durante el transcurso de la carrera universitaria, por su apoyo hacia las obligaciones y por el desarrollo del presente trabajo.

Al Ingeniero Vinicio Carrera, por ser guía durante el desarrollo del trabajo de titulación y por su apoyo incondicional ante todas las dudas que surgieron en el camino.

Hernán Guillermo Aguirre Silva

A mi familia, que son mi sustento en todo aspecto de mi vida y sin ellos nada de esto sería posible.

A la Universidad, por llevarme por el camino del conocimiento hasta esta la culminación de mi carrera.

A mis amigos, que vivieron día tras día esta aventura a mi lado.

A mi compañero Hernán Aguirre, por su apoyo incondicional a lo largo de la carrera y de este trabajo de titulación.

A mi tutor de tesis el Ingeniero Vinicio Carrera, por guiarme a través del presente trabajo con sus vastos conocimientos y su experiencia.

Santiago Andrés Gallegos Moya

INDICE DE CONTENIDOS

CERTIFICACIÓN.....	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN	iv
DEDICATORIA.....	vi
AGRADECIMIENTOS	vii
INDICE DE CONTENIDOS.....	viii
INDICE DE TABLAS	x
INDICE DE FIGURAS	xi
LISTADO DE ABREVIATURAS.....	xii
RESUMEN.....	xiii
ABSTRACT	xiv
1. CAPÍTULO I: INTRODUCCIÓN	1
1.1. Antecedentes	1
1.2. Justificación.....	1
1.3. Alcance del Proyecto	2
1.4. Objetivos	3
1.4.1. General	3
1.4.2. Específicos.....	3
2. CAPITULO II: MARCO TEÓRICO	4
2.1. Sistema de Radar	4
2.2. Preprocesamiento de la Señal.....	4
2.3. Características extraídas para Machine Learning	5
2.4. Machine Learning	6
2.5. Deep Learning.....	8
2.6. Algoritmo Adam.....	14
2.7. Métodos de validación	14
2.8. Parámetros de rendimiento	15
3. CAPITULO III: METODOLOGÍA	18
3.1. Implementación del Sistema	18

3.1.1. Descripción General.....	18
3.2. Señal de Radar.....	19
3.2.1. Base de datos	19
3.3. Técnicas con Machine Learning	21
3.3.1. Extracción de características	21
3.3.2. Entrenamiento del sistema.....	22
3.4. Técnicas con Deep Learning	22
3.5. Análisis de Resultados	24
4. CAPÍTULO IV: RESULTADOS	25
4.1. Resultados con Machine Learning	25
4.2. Resultados con Deep Learning	29
4.3. Comparación entre Machine Learning y Deep Learning.....	34
4.4. Comparación de Machine, Deep Learning y algoritmos bayesianos.....	36
5. CAPÍTULO V: CONCLUSIONES.....	38
6. TRABAJOS FUTUROS	39
7. BIBLIOGRAFÍA.....	40

INDICE DE TABLAS

Tabla 1	<i>Clasificación de objetivos de un sistema por acierto y error.....</i>	16
Tabla 2	<i>Características en el dominio del tiempo y frecuencia extraídas.....</i>	21
Tabla 3	<i>Parámetros de entrenamiento con cinco millones de muestras.</i>	25
Tabla 4	<i>Parámetros de entrenamiento con dos millones de muestras.....</i>	26
Tabla 5	<i>Falsos positivos existentes en el rango de -5 dB a 3 dB.....</i>	28
Tabla 6	<i>Parámetros de entrenamiento con dos millones de muestras.....</i>	28
Tabla 7	<i>Parámetros de entrada utilizados para cinco millones de muestras.....</i>	29
Tabla 8	<i>Parámetros de entrada utilizados para dos millones de muestras</i>	30
Tabla 9	<i>Parámetros de entrada utilizados para las pruebas de Hidden Layers</i>	32
Tabla 10	<i>Parámetros de entrada utilizados para dos millones de muestras (Amplitud).....</i>	34
Tabla 11	<i>Comparación entre Machine y Deep Learning con 5 millones de muestras... ..</i>	35
Tabla 12	<i>Tiempos de compilación de los algoritmos de Machine y Deep Learning.</i>	35
Tabla 13	<i>Comparación de la probabilidad de detección entre algoritmos de Machine, Deep Learning y AR LMS MEAN en los distintos niveles de SNR.....</i>	36

INDICE DE FIGURAS

Figura 1. Técnicas de aprendizaje de Machine Learning.....	7
Figura 2. Esquema básico de una red Neuronal.....	9
Figura 3. Esquema de una red de Deep Learning	10
Figura 4. Minimización del error por el gradiente	11
Figura 5. Comportamiento de una red neuronal con alta tasa de aprendizaje.....	11
Figura 6. Cálculo del gradiente en sistemas con tasa de aprendizaje adaptativo.....	12
Figura 7. División de datos en un sistema de validación k-fold k=10.....	15
Figura 8. Diagrama de bloques de técnicas con machine learning.....	18
Figura 9. Diagrama de bloques de técnicas con Deep Learning	18
Figura 10. Señal de radar con presencia aleatoria de objetivo	19
Figura 11. Etiquetado de la muestra de señal donde existe objetivo	19
Figura 12. Etiquetado en una señal de radar donde existe y no existe objetivo	20
Figura 13. Conjunto de datos con etiqueta generadas	21
Figura 14. Extracción de características a partir de la ventana de señal	22
Figura 15. Hidden Layers dentro de la red neuronal clasificadora	22
Figura 16. Probabilidad de detección vs SNR obtenidos en el bloque de pruebas 2.....	26
Figura 17. Probabilidad de falsa alarma vs SNR obtenidos en el bloque de pruebas 2.	27
Figura 18. Probabilidad de detección vs SNR bloque de pruebas 2 (Deep Learning) ...	30
Figura 19. Probabilidad de falsa alarma vs SNR bloque de pruebas 2(Deep Learning)	31
Figura 20. Probabilidad de detección con Hidden Layers y tasa de aprendizaje fija	32
Figura 21. Probabilidad de falsa alarma con Hidden Layers y tasa de aprendizaje fija .	33
Figura 22. Resultados obtenidos con algoritmos anteriores a la investigación.....	36

LISTADO DE ABREVIATURAS

CIAM	Centro de Investigación de Aplicaciones Militares
AR	Auto Regresivo
GPU	<i>Graphics Processing Unit</i> (Unidad de Procesamiento Gráfico)
DNN	<i>Deep Neural Network</i> (Red Neuronal Profunda)
RNN	<i>Recurrent Neural Network</i> (Red Neuronal Recurrente)
LTSM	<i>Long-Short-Term-Memory</i> (Memoria de corto y largo plazo)
SGDM	<i>Stochastic Gradient Descent with Momentum</i> (Descenso del Gradiente Estocástico con Momentum)
RMSPROP	<i>Root Mean Square Prop</i> (Apoyo de Raíz Media Cuadrática)
VP	Verdaderos Positivos
FP	Falsos Positivos
FN	Falsos Negativos
VN	Verdaderos Negativos
SNR	<i>Signal to Noise Relationship</i> (Relación Señal a Ruido)

RESUMEN

La detección de objetivos de radar es un proceso que requiere de la recepción, procesamiento y análisis de las señales a analizar, donde es necesario cumplir con parámetros de alto rendimiento para garantizar la capacidad de respuesta ante posibles amenazas. Los sistemas actuales basan su funcionamiento en filtros y algoritmos que si bien cumplen su función presentan falencias frente a niveles de relación señal a ruido (SNR) bajos, situación que es común cuando se manejan señales a grandes distancias en un ambiente no controlado, generando complicaciones en el correcto funcionamiento de los sistemas. El avance de los algoritmos de Machine Learning y Deep Learning ha dado paso a sistemas de entrenamiento supervisados que reflejan mejores resultados trabajando como clasificadores. El objetivo del presente trabajo es comparar el rendimiento del algoritmo bayesiano tradicional (AR LMS MEAN) con los algoritmos de Machine Learning (Algoritmo de Bosque Aleatorio) y Deep Learning (Redes Neuronales LSTM). Los resultados muestran una mejora considerable en niveles de SNR bajos, cumpliendo con las exigencias de un sistema de detección de objetivos de radar óptimo, demostrando que si bien el entrenamiento de los algoritmos desarrollados tiene una duración mayor su desempeño es una mejora considerable con respecto a las técnicas antes utilizadas.

PALABRAS CLAVE

- **MACHINE LEARNING**
- **DEEP LEARNING**
- **REDES NEURONALES**
- **ALGORITMO DE BOSQUE ALEATORIO**
- **LONG-SHORT TERM MEMORY**
- **RADAR Y RADIONAVEGACIÓN**
- **RADIONAVEGACIÓN**
- **AVIÓNICA**
- **PATRULLA DE RADAR**
- **DETECCIÓN DE OBJETOS**

ABSTRACT

Radar target detection is a process which requires the reception, processing and analysis of the obtained signals to analyze, where it is necessary to fulfill high performance parameters to guarantee the response capacity against threats in the environment. Current systems base their operation in filters and algorithms that despite fulfilling the main purpose they present lack of performance when having low Signal to Noise Ratios (SNR), situation that is common when analyzing signals at a large distance and in not controlled environments, generating malfunctioning problems in the systems. The improvement in Machine Learning and Deep Learning algorithms have led to supervised training systems that reflect better results as classifiers. The objective of the present work is to compare the classic Bayesian algorithm (AR LMS MEAN) performance with the ones used in Machine Learning (Random Forest Algorithm) and Deep Learning (LSTM Neural Nets). The results show a considerable improvement in low SNR levels, fulfilling the requirements for an optimum radar objective detection system, demonstrating that even if the developed algorithms training processes take more time, their performances are a significant advance regarding the traditional methods.

KEYWORDS

- **MACHINE LEARNING**
- **DEEP LEARNING**
- **NEURAL NETWORKS**
- **RANDOM FOREST ALGORITHM**
- **LONG-SHORT TERM MEMORY**
- **RADAR AND RADIONAVIGATION**
- **RADIONAVIGATION**
- **AVIONICS**
- **RADAR CUTTER**
- **OBJECT DETECTION**

1. CAPÍTULO I: INTRODUCCIÓN

1.1. Antecedentes

El avance del procesamiento digital de señales permite el estudio de la detección de señales de radar en los últimos años, varios algoritmos de detección desarrollados en este tiempo (Kelly & Forsythe, 1989) permite contar con un modelo matemático para el análisis del ruido que siempre está presente en las señales. El ruido impide tener una representación ideal de la señal deseada en el sistema, siendo imposible la detección de los objetivos, dando lugar a una imprecisión (Gini & Farina, 2002). Para superar este problema han sido investigadas técnicas de detección adaptativas y otras más sofisticadas con el propósito de minimizar las interferencias, obteniendo mejores resultados en la detección de objetivos de radar con modelos capaces el futuro comportamiento de las señales (Bandinera, De Maio, Greco, & Ricci, 2007).

A pesar de los avances mencionados, es necesario un diseño y supervisión constante de los sistemas de detección de radar para su correcto funcionamiento, siendo esto una gran limitante para generar autonomía dentro de los sistemas (LeCun, Bengio, & Geoffrey, 2015). Para cubrir esta falencia se han desarrollado los algoritmos de Machine Learning y Deep Learning que buscan mediante su implementación reducir drásticamente la intervención humana dentro de la detección de objetivos, estos sistemas busquen extraer características de señales para determinar la presencia de objetivos de radar, más robustos y eficientes frente a la variabilidad de la señal en el tiempo y a relaciones de señal a ruido bajas.

1.2. Justificación

En la actualidad la detección de señales de radar se restringe a sistemas que requieren ser permanentemente supervisados para su correcto funcionamiento, teniendo de esta manera un procesamiento de la señal mínimo, un tiempo de resolución alto e incertidumbre frente a casos especiales presentes debido a los factores que alteran las señales constantemente, siendo esta la principal desventaja (Manson, Yonel, & Yazici, 2017).

Las técnicas de Machine Learning y Deep Learning buscan superar estos inconvenientes mediante algoritmos capaces de manejar una gran cantidad de señales de radar (Sinminyan & Zisserman, 2014) y procesarlas de tal manera que se pueda

generar una respuesta sin la necesidad de una supervisión constante del sistema; al ser un método de aprendizaje supervisado únicamente es necesario un entrenamiento previo de los sistemas mencionados para que estos posteriormente funcionen de manera autónoma y autosustentable.

La implementación de un sistema autodidacta y autosustentable para la detección de objetivos de radar para aplicaciones militares de las Fuerzas Armadas es importante para poder verificar constantemente el estado actual de objetivos que ingresen o salgan de zona ecuatorial y estar preparados para cualquier circunstancia, generando un tiempo de respuesta menor ante un objetivo que pueda representar una amenaza. El desarrollo de un sistema fiable que cumpla con parámetros de altas exigencias de rendimiento, como es el caso de los radares con aplicaciones militares, se garantiza la seguridad del estado ante las diferentes situaciones que se pueden suscitar en el territorio ecuatoriano.

Por lo expuesto anteriormente, el desarrollo e implementación de algoritmos de Machine Learning y Deep Learning para la estimación de parámetros y una mejor detección de objetivos de señales de radar, se constituye un método computacionalmente eficiente para el procesamiento de señales, obteniendo mejores resultados que las técnicas tradicionales de detección y clasificación (Jokanovic, Amin, & Ahmad, 2016).

1.3. Alcance del Proyecto

Para la elaboración del presente trabajo, se utiliza un simulador de radar aprobado por el Centro de Investigación de Aplicaciones Militares (CIAM), que genera bases de datos de las señales de radar, con diferentes niveles de relación señal a ruido, en las que se determine con antelación si es que existe o no objetivo de radar en las mismas. Considerando que se empleará algoritmos de Machine Learning, se requiere realizar un preprocesamiento antes de que inicie el entrenamiento del sistema. El preprocesamiento consiste en realizar extracción de características tanto de tiempo como de frecuencia.

En el caso de Deep Learning no se requiere ningún preprocesamiento, la señal de entrada es la información almacenada en las bases de datos, la misma que modifica conforme su paso por la red neuronal, los pesos internos, para que de esta manera el sistema aprenda dentro de sus capas a diferenciar una clase de señal de otra, extrayendo por sí mismo las propiedades que las caracterizan.

Cada entrenamiento en ambos algoritmos se los corrobora mediante un método de validación cruzado llamado '*k-fold*', garantizando de esta manera resultados fiables.

Una vez se obtengan los resultados con ambos algoritmos, se comparará las métricas de rendimiento de ambos. De esta manera se concluirá cuál de los dos métodos empleados es el más efectivo y más eficaz.

1.4. Objetivos

1.4.1. General

Desarrollar un sistema de detección de objetivos de radar mediante la utilización de algoritmos de Machine Learning y Deep Learning.

1.4.2. Específicos

- Investigar y analizar algoritmos existentes de Machine Learning y Deep Learning para poder aplicarlos a sistemas de radar.
- Generar una base de datos de señales de radar para el entrenamiento adecuado de los algoritmos de Machine y Deep Learning.
- Etiquetar las ventanas generadas de las señales de radar con información de presencia o ausencia de objetivo.
- Realizar la extracción de características correspondiente para utilizarlas en el método de entrenamiento de Machine Learning.
- Determinar los parámetros de entrenamiento óptimos en los métodos de Machine Learning y Deep Learning.
- Analizar y comparar los resultados obtenidos de los algoritmos de Machine Learning y Deep Learning para determinar el rendimiento de estos y seleccionar el más efectivo, además de comparar los resultados obtenidos con las técnicas bayesianas tradicionales (AR LMS MEAN).

2. CAPITULO II: MARCO TEÓRICO

En este capítulo se presentará la fundamentación teórica del presente trabajo empezando con la definición de un sistema de radar y algunas de sus características de operación, posterior a ello se explicará en que consiste el preprocesamiento inicial necesario para el entrenamiento del sistema, así como también se realizará una breve explicación de lo que consiste los entrenamientos de Machine Learning y Deep Learning. Finalmente se explicará las diferentes métricas utilizadas en el trabajo para poder presentar los resultados.

2.1. Sistema de Radar

Un radar es un sistema electromagnético diseñado para la detección, localización y reconocimiento de objetivos. El sistema transmite señales electromagnéticas y recibe ecos de los blancos con el fin de extraer información para ser analizada y procesada (Barton, 2005).

Uno de los problemas que presenta un sistema de radar para la detección de objetivos es el Clutter al cual se lo define como ecos no deseados siempre presentes en la señal receptada. Estos pueden provenir de la tierra, mar, lluvia, animales o auroras boreales (Barton, 2005).

La relación existente entre la señal de radar y la señal de Clutter, también llamada relación señal a ruido (SNR), es utilizada para cuantificar la cantidad de ruido existente en la señal. Esta se expresa comúnmente en decibelios (dB). (Bolton, 1995)

$$SNR = 10 \log \left(\frac{\text{potencia_señal}}{\text{potencia_ruido}} \right) \text{ dB}$$

2.2. Preprocesamiento de la Señal

Es un sistema previo a la manipulación y análisis de la señal de entrada, en este caso, realizar una extracción de características de la señal de radar para que ingrese a la fase de entrenamiento con algoritmos de Machine Learning (Cuartas, Madrigal, & Torres, 2013), esto se explicará más adelante.

La extracción de características se comprende como la extracción de diferentes propiedades, que pueden ser en función de tiempo y frecuencia, de un conjunto de datos de entrada (Guyon, Gunn, Nikraves, & Zadeh, 2006).

Dentro de la literatura se presenta varios casos en los cuales se extrae diferentes características en aplicaciones de radar, por ejemplo, en (Álvarez Pastor, 2016) se extraen características estadísticas de media y desviación estándar. En (Zhang, Rong,

Hu, & Jin, 2004) por otro lado realizan extracción de características utilizando la entropía de la señal.

A continuación, se presenta las características a extraer; la mayoría de estas son fundamentadas en (Upton & Cook, 1996):

2.3. Características extraídas para Machine Learning

- **Media**

Es igual a la suma de todas las observaciones dividida para el número total de las mismas.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

- **Desviación Estándar**

Es igual a la raíz cuadrada de la varianza. Indica el promedio de que tanto se encuentran dispersos los datos con respecto a la media.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

- **Desviación Media Absoluta**

Es igual a la sumatoria del valor absoluto de la diferencia entre las observaciones y la media, dividido para el número total de observaciones.

$$MAD = \frac{\sum_{i=1}^n |x_i - \bar{x}|}{n}$$

- **Valor Máximo**

Es el valor máximo de todas las observaciones.

$$MAX = \max\{x(i)\}$$

- **Valor Mínimo**

Es el valor mínimo de todas las observaciones:

$$MIN = \min\{x(i)\}$$

- **Energía**

La energía de un vector de valores se la puede definir como:

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

- **Rango Intercuartílico**

Es igual a la diferencia entre el cuartil más grande con respecto al más pequeño. Así como la mediana subdivide los datos en dos mitades, los cuartiles la subdividen los datos

en cuartos. Dado esto, existen tres cuartiles, el cuartil más bajo (Q1), el mediano (Q2) y el cuartil más alto (Q3) (Upton & Cook, 1996).

$$IQR = Q_3 - Q_1$$

- **Entropía**

Llamada también entropía de Shannon y entropía de la información. Mide la cantidad de incertidumbre presente en un conjunto de datos (Mataix & Rivadulla, 2002).

$$H = - \sum_{i=1}^n x_i \log_2 x_i$$

- **Modelo Autorregresivo**

El modelo autorregresivo (AR) se trata de un modelo predecible con respecto a observaciones inmediatas anteriores (Gras, 2001).

La forma discreta de un modelo autorregresivo $y(t)$ de orden p es representada por la relación.

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + \dots + a_p y(n-p) + \epsilon(n)$$

Donde $\epsilon(n)$ es ruido blanco (Emery & Thomson, 2004).

En el presente trabajo se empleará el método de Burg, este es un estimador para encontrar los coeficientes computacionales de un modelo autorregresivo $y(n)$.

Es el método que presenta el error cuadrático medio más bajo para los parámetros estimados del modelo AR. El método de Burg es el estimador preferido como estimador de parámetros AR (Broesen, 2006).

- **Frecuencia Media**

Es el promedio ponderado de todas las frecuencias que se presenta en un conjunto de datos. Esta es la única característica en función de la frecuencia a ser utilizada, las demás mencionadas son en función del tiempo.

2.4. Machine Learning

Machine Learning o Aprendizaje de Máquina es Inteligencia Artificial que permite a un sistema aprender de diferentes datos de entrada en lugar de una programación explícita por parte de un usuario. Esta utiliza diversos algoritmos que iterativamente aprenden de los datos de entrada para poder así predecir resultados futuros (Hurwitz & Kirsch, 2018).

El aprendizaje automático emplea dos tipos de técnicas para obtener resultados. El primero es el aprendizaje de máquina supervisado, este entrena un modelo de datos conociendo previamente la entrada y la salida, con la finalidad de predecir salidas futuras. El segundo es el aprendizaje de máquina no supervisado, este trabaja solamente con

datos de entrada conocidos, con el objetivo de determinar patrones o agrupar elementos similares (Bonaccorso, 2017). Esto se lo muestra en la **Figura 1**.



Figura 1. Técnicas de aprendizaje de Machine Learning.

La técnica de aprendizaje supervisada emplea algoritmos de Clasificación y Regresión. Los algoritmos de clasificación se utilizan para datos discretos y los algoritmos de regresión para tipos de datos continuos. En el otro caso, la técnica de aprendizaje no supervisada utiliza algoritmos de clustering, los cuales se emplean para encontrar patrones o agrupaciones ocultos dentro de la base de datos de entrada (Bonaccorso, 2017).

Para el presente trabajo al tratarse de un sistema en el cual se conoce tanto la entrada de datos como la salida se empleará un aprendizaje supervisado. Las clases mencionadas son discretos, por lo tanto, se utilizará algoritmos de clasificación. Hay que tomar en cuenta que la cantidad de datos a ser entrenados es significativa, esto se lo explica más adelante, por lo tanto, el tiempo de entrenamiento también será significativo. Esto conduce a que el clasificador a utilizar debe ser rápido en cuanto a la predicción, los algoritmos capaces de esto son los árboles de decisión (MathWorks, 2018). Una de las principales desventajas de estos algoritmos es que son propensos a un sobreajuste de los datos de entrenamiento. Para solucionar este inconveniente se utiliza el algoritmo de: Árboles de decisión empaquetados.

También llamado Algoritmos del bosque aleatorio, fue creado por Breiman L (MathWorks, 2018). Es un algoritmo de machine Learning supervisado. Utiliza múltiples arboles de decisión para fusionarlos y obtener una predicción más exacta y estable. Puede ser utilizada tanto en problemas de clasificación como de regresión, es decir para problemas discretos y continuos.

El algoritmo en lugar de hallar la característica más importante para entrenar el sistema cuando se divide un nodo busca la mejor característica entre un subconjunto aleatorio de características.

Como se mencionó el método del bosque aleatorio es un conjunto de árboles de decisión, pero son diferentes entre sí.

Un árbol de decisión al entrenar un grupo de datos con características y etiquetas este creará reglas que serán utilizadas para realizar predicciones. El método del bosque aleatorio selecciona varias observaciones y características para construir diversos árboles de decisión y con esto promediar el resultado.

Otra diferencia es que este método evita el sobre entrenamiento en la gran mayoría de casos a diferencia de un árbol de decisión profundo (Donges, 2018).

Una alternativa para realizar el entrenamiento del sistema dentro del software MATLAB® es la aplicación de *Classification Learner*.

2.5. Deep Learning

Deep Learning es una herramienta derivada de Machine Learning orientada al aprendizaje por computadora con diferentes niveles de extracción que ha sido utilizada en la última década en aplicaciones como clasificadores de señales, detección y reconocimiento de objetos, voz, etc. Las técnicas convencionales de Machine Learning se ven limitadas por el procesamiento que estas requieren dentro de su entrenamiento, por lo que no pueden procesar los datos en su estado natural y requieren del diseño de un extractor de características para cada tipo diferente de señal, Deep Learning por el contrario realiza este proceso de manera autónoma determinando aquellas características relevantes y transformando la información en bruto a una representación más adecuada para que un clasificador pueda determinar la salida del sistema (LeCun, Bengio, & Geoffrey, 2015). Una ventaja al utilizar redes neuronales en Deep Learning es que estos sistemas se muestran robustos frente a la variabilidad de las señales de entrada aún si estas nunca han sido mostradas a la red previamente, esto se logra en los sistemas tradicionales con un buen diseño de extractores de características, lo cual como se mencionó anteriormente no es necesario en Deep Learning.

La desventaja principal de Deep Learning frente a Machine Learning es el coste computacional, sin embargo su avance ha sido posible debido al progreso en conjunto de las Unidades de Procesamiento Gráfico (GPU) con compañías como NVIDIA y Mobileye que han utilizado las técnicas de Deep Learning para el desarrollo de visión por computadora y reconocimiento de objetos, dándole al hardware la posibilidad de manejar

enormes cantidades de datos necesarias para el entrenamiento de la red neuronal (LeCun, Bengio, & Geoffrey, 2015).

Deep Learning basa su concepto en las redes neuronales las cuales dividen la capa de entrada y la capa de salida por medio de Hidden Layers (Hinton, y otros, 2012) cómo se puede observar en la **Figura 2**, estas capas intermediarias son las encargadas de reemplazar la extracción de características y el clasificador diseñado manualmente en Machine Learning sino que se generan como procedimientos de propósito general, definiendo los parámetros de entrenamiento necesarios para el entrenamiento solamente.

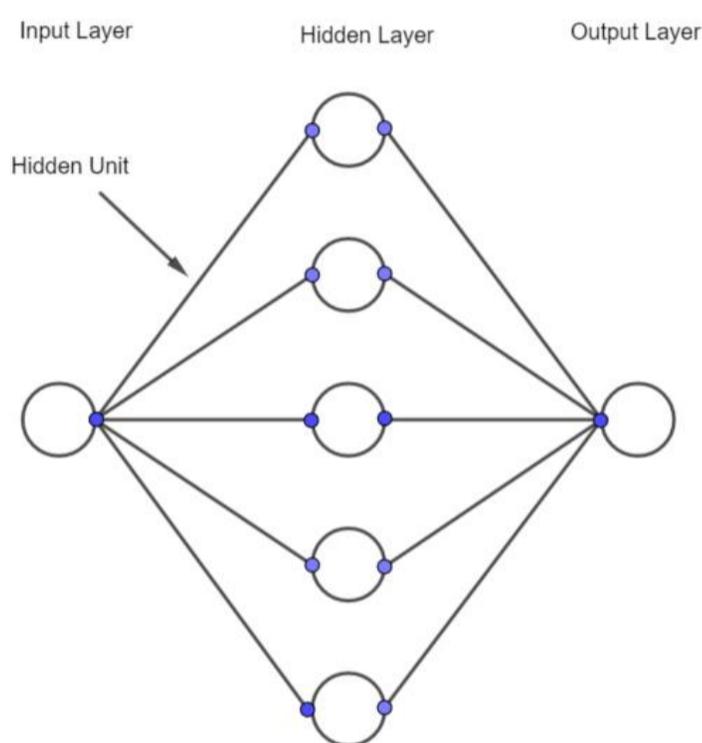


Figura 2. Esquema básico de una red Neuronal

Una red neuronal se denomina de Deep Learning si su complejidad, es decir el número de Hidden Layers, aumenta de tal manera que, si bien el entrenamiento dentro de esta red representará un coste computacional mayor, esto se reflejará de igual manera en el desempeño del sistema debido a que la extracción de las características para la clasificación de las señales de entrada se realizará en mayor medida y de una manera más sofisticada, dado lugar a mejores resultados. Las Hidden layers como se puede observar en la **Figura 3** se concatenan en cascada siendo la salida de cada Hidden Layer la entrada de la siguiente, las conexiones entre cada una de estas capas se denominan Hidden Units y poseen valores denominados pesos que son modificados conforme el paso de los datos por cada una de las capas (Tiros, Bueno, García, & Benítez, 2018). La complejidad del sistema dependerá esencialmente del número de Hidden Layers y

Hidden Units utilizados en el sistema, viéndose esto reflejado tanto en una mejor extracción de características como en un procesamiento más complejo y extenso del sistema.

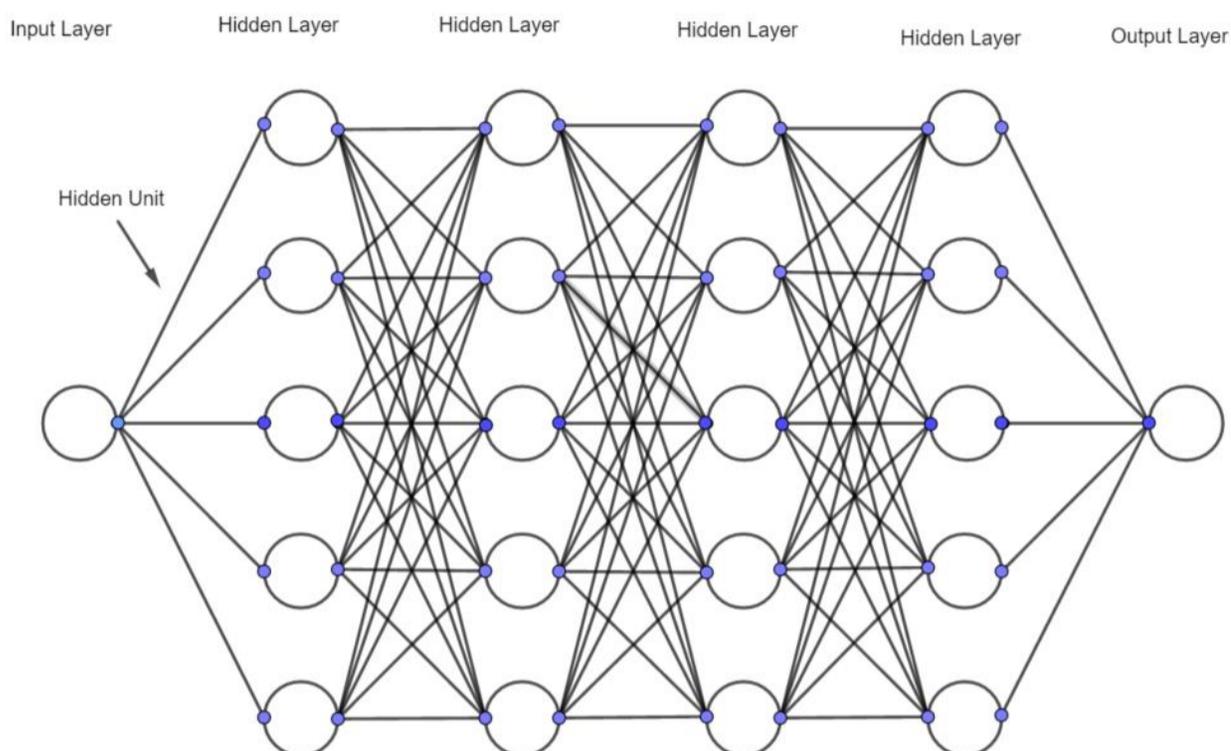


Figura 3. Esquema de una red de Deep Learning

El número de Hidden Layers y de Hidden Units que se encuentran en la red neuronal determinan el rendimiento y la complejidad de esta y por ende la fiabilidad de la respuesta de salida, sin embargo, el número de capas utilizadas es proporcional al procesamiento que se requerirá tanto el proceso de entrenamiento como la utilización de la red en su funcionamiento por lo que es necesario crear un balance entre la respuesta del sistema y su rendimiento en tiempo.

Estos pesos ubicados en las Hidden Units determinan la salida del sistema y por lo tanto el rendimiento de este. Se crea una función que mide el error que existe entre los elementos de entrada y los de salida siendo el parámetro que se busca minimizar con el ajuste de los pesos. El algoritmo de aprendizaje calcula un vector gradiente que aumentará o disminuirá en función de pequeñas modificaciones en los pesos, buscando dirigirse en la dirección que minimice el error en cada paso de la información por la red neuronal (LeCun, Bengio, & Geoffrey, 2015), esto se realiza disminuyendo el valor del gradiente hasta que se aproxime a cero y por lo tanto se encuentre el error mínimo como se puede observar en la **Figura 4.**

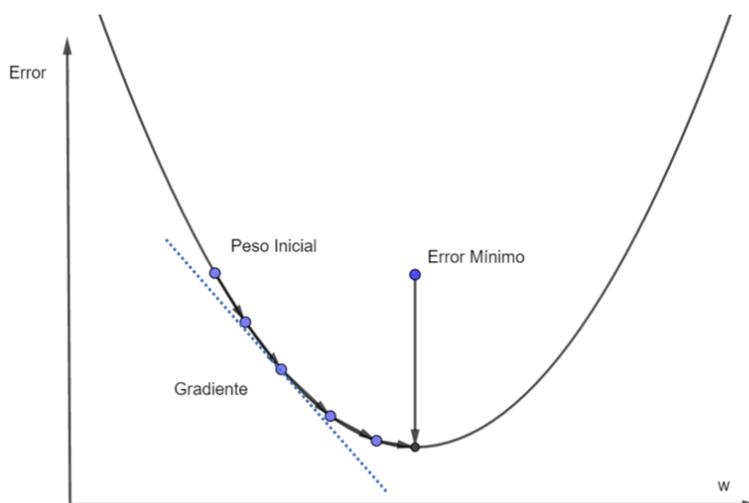


Figura 4. Minimización del error por el gradiente

La tasa de aprendizaje es uno de los factores más importantes dentro del aprendizaje de Deep Learning y se define como la rapidez con la que la red modifica los pasos dentro del cálculo en las Hidden Units para acercarse al error mínimo determinando el rendimiento del sistema desde ordenado y óptimo hasta oscilatorio y caótico (Sutskever, Martens, Dahl, & Hinton, 2013); si se utiliza una tasa de aprendizaje baja la constancia de los resultados es notable, pero la poca rapidez con la que estos son procesados lleva a que los resultados sean de un rendimiento menor, aunque constante como se puede observar en la **Figura 4**. Una tasa de aprendizaje alta genera un rendimiento más inmediato en los resultados, sin embargo es inminente la inestabilidad debido a que una tasa de aprendizaje alta busca de forma más rápida el valle del gradiente por lo que esta tiende a generar rebotes dentro de la curva pudiendo incluso divergir en el error como se observa en la **Figura 5** (Martens, 2010),

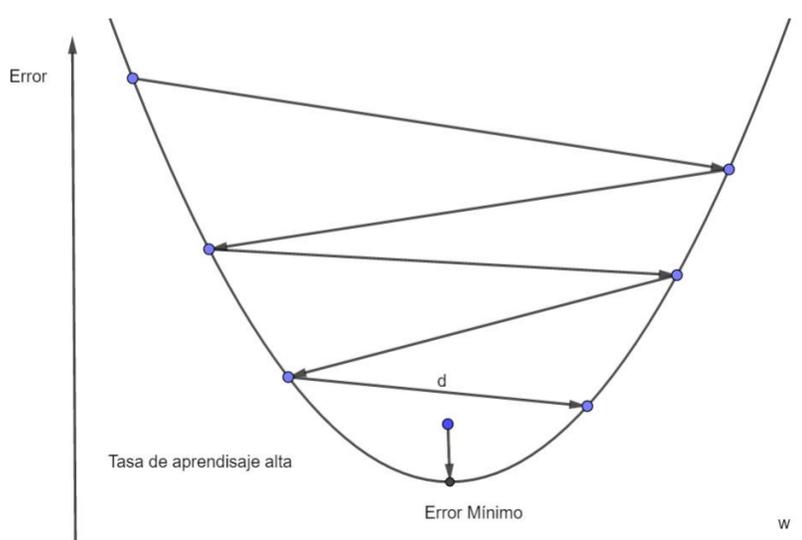


Figura 5. Comportamiento de una red neuronal con alta tasa de aprendizaje

La alternativa para solucionar este problema en el cálculo del error y ajuste de los pesos de la red de Deep Learning es una tasa de aprendizaje adaptativa, buscando la minimización del error durante el principio del entrenamiento, determinando los ajustes de manera acelerada y posteriormente al acercarse el gradiente a su valor de cero la tasa de aprendizaje disminuye para que de esta manera se busque de la manera más estable sin generar inestabilidad ni rebotes en la curva como se muestra en la **Figura 6**.

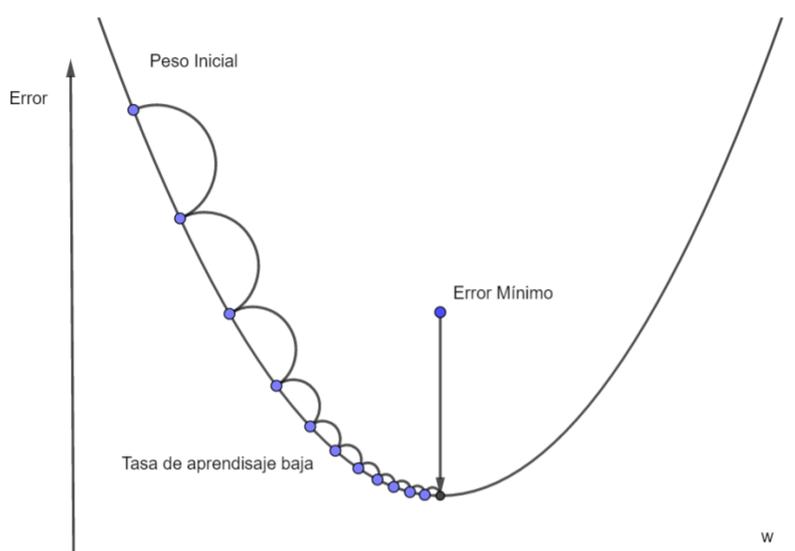


Figura 6. Cálculo del gradiente en sistemas con tasa de aprendizaje adaptativo

Debido al modo de extracción de características y ajuste de pesos que se maneja dentro de Deep Learning en la mayoría de las ocasiones es necesario que los datos sean reutilizados varias veces para que los pesos iniciales sean reajustados y de esta manera la red pueda extraer características que no fueron detectadas en un principio, se denomina época (epoch) a cada vez que todos los datos atraviesan por la red desde la capa de entrada hasta la capa de salida, cada época adicional si bien puede mejorar el rendimiento del sistema representa un tiempo adicional en el entrenamiento total, por lo que es necesario establecer un balance entre tiempo y rendimiento.

Las bases de datos utilizadas para el entrenamiento de Deep Learning son de tamaño considerable por lo que toda la base de datos no puede pasar de una sola vez a través de la red, es por esto por lo que existe el parámetro de batch size o tamaño de lote que representa el tamaño en el cual el bloque de datos se separa en sub-bloques para su procesamiento de una manera optimizada. Se denomina iteración al número de veces que los lotes pasan por la red para completar una época siendo un indicador del avance del entrenamiento.

El número de clases en la red neuronal representa los diferentes grupos en los cuales se puede clasificar, mientras el clasificador tenga más opciones en las cuales se puede categorizar los datos ingresados, la red neuronal tendrá que tener en cuenta estos grupos haciendo de esta manera más compleja la red y su funcionamiento.

El avance de las redes neuronales han dado paso a la evolución de las redes neuronales profundas (DNN) y redes neuronales recurrentes (RNN) hacia sistemas que tienen una memoria dentro de sus Hidden Layers, estas nuevas redes denominadas Long-Short Term Memory (LSTM) han demostrado tener mejor desempeño en experimentos de secuencias temporales con respecto a sus predecesoras (Sundermeyer, Schluter, & Ney, 2012), uno de los principales problemas de las DNN y RNN consiste en el problema del desvanecimiento del gradiente ya que al no tener una memoria previa de los datos que atravesaron la red el gradiente puede tender a aumentar o decaer exponencialmente haciendo que el sistema se vuelva inestable, por lo que dentro de las redes neuronales LSTM existe la implementación de bloques de memoria en una Hidden Layer recurrente o capa LSTM, esta es la encargada de mantener el multiplicador que afecta al gradiente, que tiene a ser mayor o menor a uno, a mantenerse en la unidad volviendo al sistema más estable (Sak, Senior, & Beaufays, 2014), con una tasa de aprendizaje adecuada en conjunto con una capa de LSTM se tiende a una convergencia estable con resultados de rendimiento mayores con respecto a las DNN y RNN tradicionales.

Para mejorar el rendimiento de una estructura LSTM, se utiliza una capa denominada Fully Connected en el sistema, esta es la encargada de transformar la información recibida de la capa LSTM y obtener una representación de las características de mayor nivel u orden, haciendo que la clasificación de los datos se realice de una manera más efectiva (Sainath, Vinyals, Senior, & Sak, 2015).

La capa Softmax utilizada posteriormente a la capa Fully Connected se utiliza para determinar la probabilidad de la clase, la cual será utilizada en la capa de salida denominada clasificador para determinar la salida del sistema (Le, Oparin, Allauzen, Gauvain, & Francois, 2011).

2.6. Algoritmo Adam

La optimización estocástica de gradientes ha dado paso a que se desarrolle un método utilizado en Deep Learning denominado *Adam*, el cual ha demostrado el ser más efectivo que las técnicas precedentes que utilizan el cálculo del gradiente y actualización de parámetros como lo son el momentum de descenso de gradiente estocástico (SGDM) usado en grandes bases de datos (Ruder, 2016) y el apoyo de raíz media cuadrática (RMSprop) (Kurbiel & Khaleghian, 2017), teniendo Adam como avance principal el uso de tasas de aprendizaje adaptativo y el cálculo del gradiente con una optimización en la memoria requerida para los procedimientos además de una robustez ante secuencias de datos con bajos niveles de relación señal a ruido y una convergencia más acelerada hacia un rendimiento estable (Kingma & Lei Ba, 2014).

Adam utiliza en la base de su algoritmo tres parámetros esenciales de inicialización: la tasa de aprendizaje, y los momentos de primer y segundo orden del gradiente, siendo la media y varianza respectivamente. El cálculo de la dirección del gradiente con estos parámetros se realiza en subconjuntos o lotes de los datos utilizados en su entrenamiento y se adaptan constantemente en cada iteración corrigiendo la tendencia del gradiente y con ello la predisposición del sistema hacia la convergencia en sus resultados, disminuyendo el error conforme sigue el entrenamiento. La corrección de esta tendencia es una mejora con respecto a los métodos anteriores previniendo la divergencia de los resultados finales.

2.7. Métodos de validación

Dentro del desarrollo de técnicas de Machine Learning y Deep Learning que utilizan para su entrenamiento grandes grupos de datos es necesario crear una validación de estos de tal manera que los resultados obtenidos en sus parámetros de rendimiento se reflejen de manera real, para eliminar la variabilidad en los resultados entonces existen técnicas de validación que reducen el problema y orientan al sistema a una media de resultados. La validación cruzada *k-fold* ha demostrado ser efectiva para estimar parámetros de precisión cuando se utiliza grandes bases de datos disminuyendo la posibilidad de obtener resultados demasiado optimistas en contraste con la realidad de

los mismos, lo cual se puede observar en clasificadores como los son los métodos Bayesianos y del vecino más próximo (Wong, 2015).

La validación cruzada de *k-fold* tiene como principios esenciales el dividir los datos en *k* grupos de igual tamaño, los cuales tienen la misma probabilidad de ser escogidos en el proceso y tratarse como una observación independiente de los otros grupos en cada experimento realizado como se muestra en la **Figura 7** **Figura 1**. El grupo seleccionado servirá como medidor del rendimiento de los parámetros del sistema y se registraran los parámetros obtenidos en el experimento, este proceso se realiza con los *k* grupos obtenidos de la división del grupo original de datos y se realiza un promedio de los resultados obtenidos a fin de encontrar una media de estos y reducir la variabilidad que puede generarse en un solo experimento.

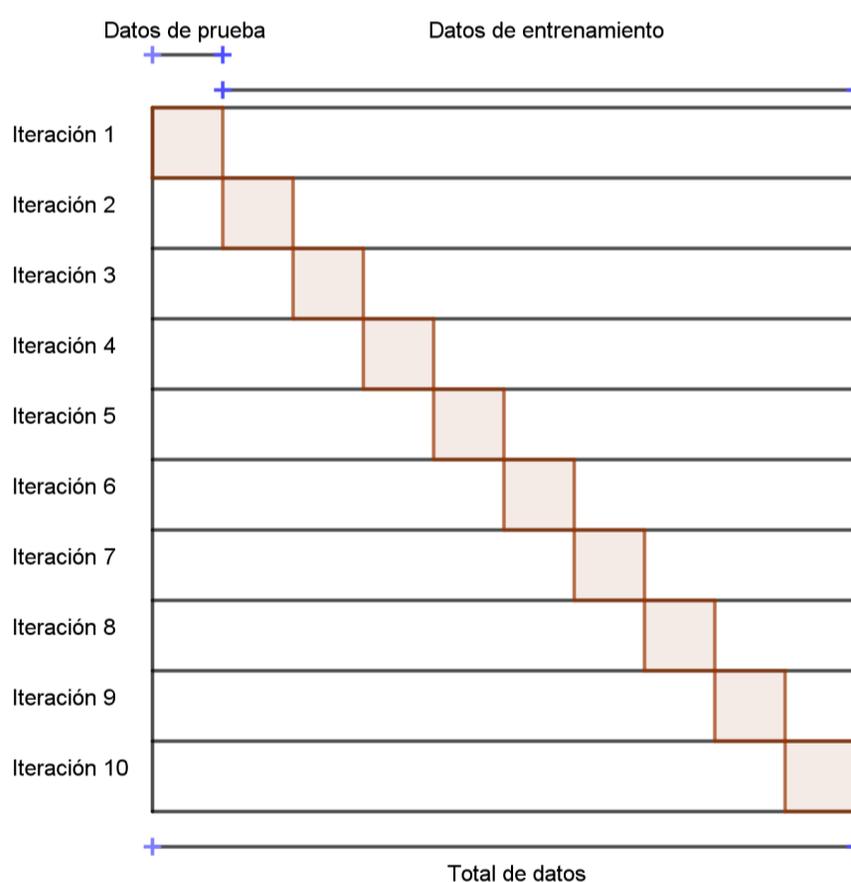


Figura 7. División de datos en un sistema de validación *k-fold* $k=10$

2.8. Parámetros de rendimiento

Para poder determinar el funcionamiento de un sistema como el caso de Machine Learning y Deep Learning es necesario tener parámetros de validez que muestren una medida cuantitativa de su precisión. Esta validez se mide a través de la sensibilidad y la especificidad, parámetros utilizados en clasificadores donde se determina la existencia o no de un objetivo a medir. Estos parámetros se pueden observar la **Tabla 1**

Tabla 1
Clasificación de objetivos de un sistema por acierto y error

	<i>Presencia de un objetivo</i>	<i>Ausencia de un objetivo</i>
<i>Prueba Positiva</i>	Verdadero Positivo (VP)	Falso Positivo (FP)
<i>Prueba Negativa</i>	Falso Negativo (FN)	Verdadero Negativo (VN)

Cada uno de los elementos de la **Tabla 1** se definen a continuación.

Verdaderos positivos (VP): Aquellos datos que son positivos en su respuesta y son detectados de manera acertada.

Falsos positivos (FP): Aquellos datos que son negativos en su respuesta y son detectados de manera acertada.

Falsos negativos (FN): Aquellos datos que son positivos en su respuesta y son detectados de manera equivocada.

Verdaderos negativos (VN): Aquellos datos que son negativos en su respuesta y son detectados de manera equivocada.

Se define a la sensibilidad como la capacidad del sistema de determinar de manera acertada la detección de un objetivo en caso de existir, es decir el número de casos positivos detectados acertadamente sobre el número total de casos positivos reales (Parikh, Mathai, Parikh, & Thomas, 2008), en el área de radar la sensibilidad se define también como probabilidad de detección.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

Se define a la especificidad como la capacidad del sistema de determinar la ausencia de un objetivo en el caso de no existir, es decir el número de casos negativos detectados acertadamente sobre el número total de casos negativos reales (Parikh, Mathai, Parikh, & Thomas, 2008).

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

En contraste a la especificidad se define en el área de radar la probabilidad de falsa alarma como

$$\textit{Probabilidad de Falsa Alarma} = 1 - \textit{Especificidad}$$

Siendo la probabilidad de detectar un objetivo sin existir alguno.

3. CAPITULO III: METODOLOGÍA

3.1. Implementación del Sistema

3.1.1. Descripción General

Actualmente en el Ecuador, las Fuerzas Armadas emplean el sistema de radar OERKILON, este se encuentra compuesto por el director de tiro SKYGUARD y las piezas de artillería. El director de tiro SKYGUARD utiliza un conjunto de datos de 6400 muestras correspondientes a los 360° de giro del radar, de las cuales 43 muestras corresponden a la presencia de un objetivo (Lara & Ortiz, 2019).

En el presente trabajo, la detección de objetivos presentes en las señales de radar se las ha propuesto mediante dos técnicas:

- Técnica con Algoritmos de Machine Learning
- Técnica con Algoritmos de Deep Learning

Ambas técnicas se han implementado en el software MATLAB® R2018a.

Se presenta a continuación, en la **Figura 8**, el diagrama de bloques de los procesos realizados para la obtención de los resultados con el método de entrenamiento de Machine Learning.

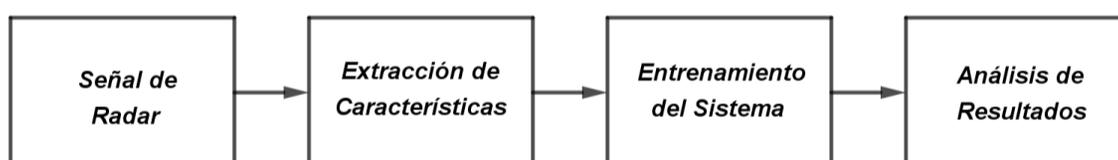


Figura 8. Diagrama de bloques de técnicas con machine learning

En la **Figura 9** se presenta el diagrama de bloques de los procesos realizados para la obtención de los resultados con métodos de entrenamiento de Deep Learning. Como se puede observar, en Deep Learning se evita la extracción de características del sistema, es decir, no existe ningún tipo de preprocesamiento de la señal de entrada para que se pueda entrenar el sistema.



Figura 9. Diagrama de bloques de técnicas con Deep Learning

La generación de la señal de radar en conjunto con la creación de las bases de datos correspondientes para las pruebas son las mismas tanto para Machine Learning como para Deep Learning.

3.2. Señal de Radar

Las señales de entrada utilizadas son generadas a partir de un simulador de señales de radar aprobada por el Centro de Investigación de Aplicaciones Militares (CIAM) con las especificaciones de la cantidad de muestras ya mencionadas.

Utilizando como parámetros del simulador un coeficiente de correlación de 0.9 y un SNR que varía desde -5 dB hasta 15 dB en pasos de 1 dBm clásico de clutter de suelo.

Se obtiene las señales de radar en las componentes de fase y cuadratura (parte real y parte imaginaria). Cada simulación consta de 6400 muestras dentro de las cuales 43 muestras consecutivas, ubicadas de manera aleatoria, indican la existencia de un objetivo, esto se lo representa en la **Figura 10**.

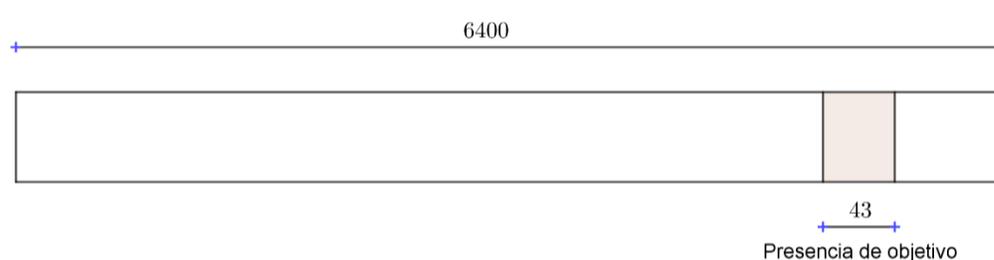


Figura 10. Señal de radar con presencia aleatoria de objetivo

En la salida del simulador se presenta los datos del radar seguido de la posición en donde empieza el objetivo, este último indicador es indispensable para poder realizar el etiquetado de cada ventana y la obtención de la base de datos.

3.2.1. Base de datos

Por cada simulación realizada se extrae las 43 muestras en las que se ubica el objetivo de radar y se la etiqueta de manera positiva como se lo representa en la **Figura 11**.

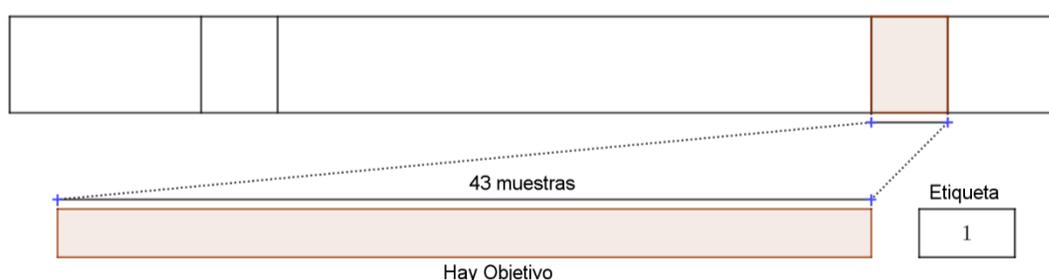


Figura 11. Etiquetado de la muestra de señal donde existe objetivo

Dando a conocer que en esa ventana existe la presencia de un objetivo. Posterior a ellos se extrae 43 muestras aleatorias consecutivas en las que no existe la presencia de

un objetivo y se la etiqueta de manera negativa, se representa este proceso en la **Figura 12**.

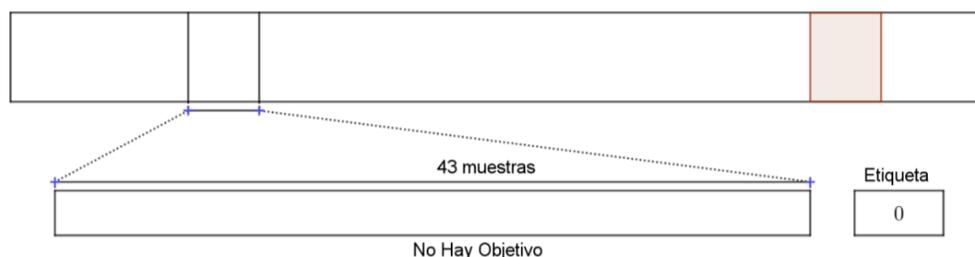


Figura 12. Etiquetado en una señal de radar donde existe y no existe objetivo

Cabe mencionar que en ningún momento se utiliza una ventana en la que exista y no exista la presencia de un objetivo, por lo que el etiquetado positivo se lo coloca únicamente en la ventana de 43 muestras en las que si hay objetivo y el etiquetado negativo solamente cuando no existe señal de objetivo.

El proceso se lo ha realizado hasta generar un conjunto de datos, que consta de la componente de fase (parte real) y de la componente de cuadratura (parte imaginaria), ver **Figura 13**.

Se ha elaborado una base de datos que consta de varios conjuntos de datos diferentes. Los conjuntos de datos se dividen de la siguiente manera: Un conjunto de cinco millones de elementos con valores con SNR de 0 dB Siendo este un valor neutral dentro del rango de SNRs utilizados. Catorce conjuntos que contienen dos millones de elementos y se ha variado los siguientes SNR: -5 dB , -4 dB , -3 dB , -2 dB , -1 dB , 0 dB , 1 dB , 3 dB , 5 dB , 8 dB , 10 dB , 12 dB , 14 dB y 15 dB . Este rango de SNR presenta el simulador de señales de radar de manera predeterminada. Los primeros valores de SNR presentados desde -5 dB hasta 0 dB se encuentran en pasos de a 1 dB , ya que son valores críticos en los cuales la potencia de la señal de ruido es mayor o igual a la potencia de la señal de radar y es importante determinar la tendencia del resultado. Para el caso de los otros rangos, desde 1 dB hasta 15 dB , se los ha dividido en pasos más amplios de 2 dB y hasta de 3 dB , ya que al ser la potencia de la señal mayor a la de ruido no causa mucho inconveniente al momento de predecir un resultado por parte del sistema.

Por último, un conjunto de datos que posee elementos de amplitud obtenida de la parte real e imaginaria se ha obtenido con SNR de 0 dB.

La cantidad de etiquetas tanto positiva como negativas son la mitad de los elementos totales, garantizando el equilibrio entre ventanas que poseen o no señal de objetivo.

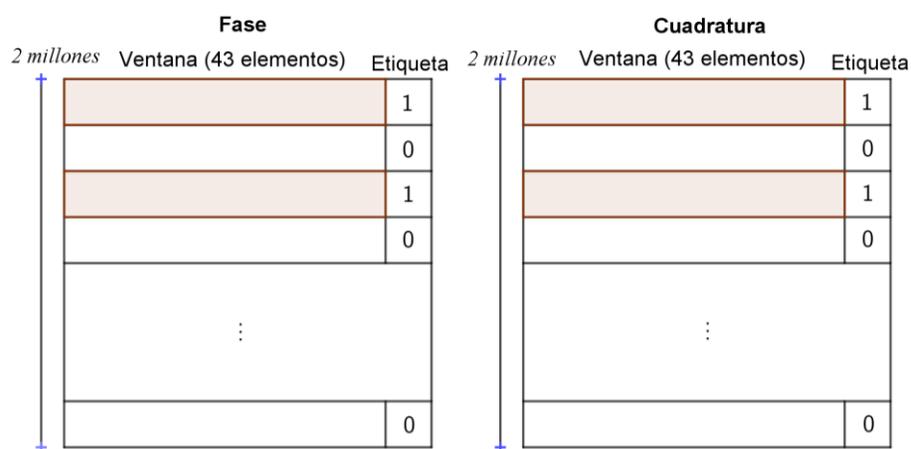


Figura 13. Conjunto de datos con etiqueta generadas

Dado que por cada simulación de radar se extrae una ventana positiva y una ventana negativa, las bases de datos se formarán como en la **Figura 13**. Además, como se especificará más adelante, al utilizar un método de validación cruzada de *k-fold* con valor de 10. El bloque de muestras a entrenar y a realizar las pruebas se las tomará de manera aleatoria.

3.3. Técnicas con Machine Learning

3.3.1. Extracción de características

Dadas las bases de datos, por cada ventana se extraen diferentes características con el objetivo de clasificar si existe o no objetivo con los algoritmos de Machine Learning. En este caso se ha utilizado 13 características que se dividen en dominio del tiempo y dominio de la frecuencia, mencionadas en la **Tabla 2** y representadas en la **Figura 14**.

Tabla 2

Características en el dominio del tiempo y frecuencia extraídas

Características en el dominio del tiempo	
ID	Característica
f1	Media
f2	Desviación Estándar
f3	Desviación Media Absoluta
f4	Valor Máximo
f5	Valor Mínimo
f6	Energía
f7	Rango Intercuartílico
f8	Entropía
f9	Modelo Autorregresivo
f10	
f11	
f12	
Características en el dominio de la frecuencia	
ID	Característica
f13	Frecuencia Media

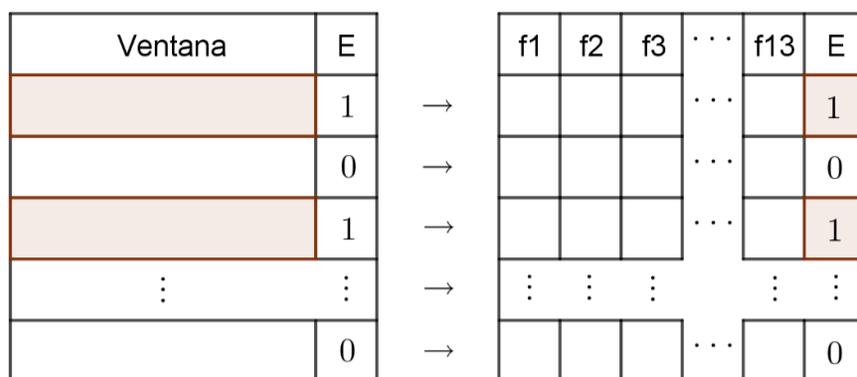


Figura 14. Extracción de características a partir de la ventana de señal

Se ha utilizado las características de las bases de datos de fase y cuadratura en una sola base de datos, llegando a tener un total de 26 características.

3.3.2. Entrenamiento del sistema

Una vez generadas las características de las bases de datos se ha utilizado el método de árboles de decisión empaquetado. Este método se encuentra en la aplicación de *Classification Learner* del software MATLAB. Seleccionando las variables de entrada y de salida, es decir, 26 características y etiquetas respectivamente.

3.4. Técnicas con Deep Learning

En comparación a Machine Learning, la extracción de características de las señales a ser analizadas se realizar de manera automática mediante las capas escondidas (Hidden Layers) que procesan el banco de datos utilizados y se define los pesos a utilizar durante el entrenamiento y que determinarán la salida del sistema y por ende la existencia o no de un objetivo de radar.

Al igual que con Machine Learning se utilizarán las bases de datos mencionadas anteriormente y se utilizará el método de validación cruzada *k-fold* obteniendo resultados en los valores de SNR de -5dBs a 15dBs.

El entrenamiento se lleva a cabo con un clasificador de tipo 'Adam' que determina los pesos del clasificador convolucional. La estructura de las Hidden Layers utilizadas en el clasificador implementado en Matlab será la que se observa en la **Figura 15**, los datos pasan a través de la capa Sequence Input para ingresar a la red de manera secuencial, las características más relevantes son extraídas por las siguientes tres capas y la capa final del clasificador determina la salida del sistema.



Figura 15. Hidden Layers dentro de la red neuronal clasificadora

Las épocas utilizadas, es decir el número de veces que los datos pasarán por la red neuronal es determinada mediante las diferentes pruebas utilizadas en función del aumento del rendimiento y la optimización del tiempo, mediante la observación del proceso de entrenamiento y la precisión que se obtiene con cada una de las iteraciones se puede evitar un sobre entrenamiento de la red neuronal lo cual no solamente aumenta el tiempo de entrenamiento sino además evita que la saturación dentro de la red afecte a su desempeño. Finalmente, el tamaño de los lotes en los que se divide el banco de datos total para ingresar por la red es determinado por la estabilización del sistema en el tiempo y la restricción de la memoria del hardware utilizado.

El número de capas escondidas (Hidden Units) al ser el parámetro que más influye en la complejidad de la red neuronal dentro del sistema se determinan mediante varios experimentos observando el cambio del rendimiento en función de la complejidad de la red, llegando a obtener un número de este parámetro que genere resultados positivos para la detección de objetivos de radar y al igual que en el caso de las épocas no llegue a sobresaturar el sistema en su complejidad y por lo tanto desempeño de ejecución.

El parámetro que afecta de manera significativa a la adaptación de la red neuronal en el proceso de entrenamiento es la tasa de aprendizaje utilizada, es decir la rapidez con la cual la red determina los pesos dentro de las capas del sistema, este parámetro varía de 0 a 1 y puede ser estático o variable. Una tasa de aprendizaje alta genera resultados de precisión más rápidos en el tiempo por lo que el número de épocas necesario sea menor, sin embargo una tasa de aprendizaje alta refleja una inestabilidad en el rendimiento final y por el contrario una tasa de aprendizaje menor si bien genera una estabilidad en los resultados su rendimiento final en los parámetros de precisión es menor por lo que para los presentes experimentos realizados se utilizó una tasa de aprendizaje adaptativo de manera descendente por un factor multiplicativo en un cierto número de épocas beneficiándose de esta manera de un rendimiento alto en los resultados y una constancia e invariabilidad de los mismos en el tiempo de entrenamiento.

Con los parámetros antes descritos y mediante los experimentos realizados los resultados obtenidos serán como se mencionó anteriormente comparados con Machine Learning para reflejar la diferencia entre ambos métodos de análisis.

3.5. Análisis de Resultados

Para el método de verificación se ha utilizado la validación cruzada *k-fold* a un valor de 10. Al obtener los resultados, se ha verificado los valores de probabilidad de detección y probabilidad de falsa alarma.

También se comparará los resultados obtenidos de las técnicas de Machine Learning y Deep Learning, adicionalmente se comparará ambos resultados con los obtenidos con los métodos bayesianos registrados en: (Lara & Ortiz, 2019).

4. CAPÍTULO IV: RESULTADOS

4.1. Resultados con Machine Learning

Para la obtención de los resultados con algoritmos de Machine Learning se ha empleado el método de *árboles de decisión empaquetados* del grupo de *Ensemble Classifiers*, utilizando la aplicación de *Classification Learner* del software Matlab.

Dentro de la aplicación se ha especificado un primer caso de entrenamiento que contiene cinco millones de ventanas con su respectiva etiqueta, 26 características referentes a la fase y cuadratura de una muestra de señal de radar, un SNR de 0 dB y un método de validación *k-fold*, donde *k* es igual a 10. Esto se resume en la **Tabla 3**.

Tabla 3

Parámetros de entrenamiento con cinco millones de muestras.

Parámetro	Valor
SNR (dB)	0
Número de datos	5000000
Número de	26
Tasa de validación <i>k-fold</i>	10

Del entrenamiento se ha obtenido: Una probabilidad de detección del 100% y una probabilidad de falsa alarma de 0.0000008. Es decir, de cada 5 millones de ventanas el sistema predijo 2 de manera incorrecta. Esto se ve reflejado como 2 falsos positivos (FP) y 0 falsos negativos (FN)

Para este entrenamiento el tiempo de procesamiento fue aproximadamente de 24 horas. Debido a este tiempo que conlleva realizar el entrenamiento de un conjunto de datos de cinco millones de ventanas, además de las limitaciones físicas presentadas por el hardware de las computadoras portátiles utilizadas, se ha optado por reducir el número de datos a dos millones, con lo cual se puede mostrar la tendencia que presenta el sistema dependiendo del valor de SNR especificado en la base de datos.

Al igual que en el primer caso se ha empleado 26 características referentes a la fase y cuadratura de una muestra de señal de radar y un método de validación *k-fold*, donde *k* es igual a 10. La diferencia en este nuevo caso es que se utiliza dos millones de muestras por cada conjunto de datos y SNR variable de -5 dB a 15 dB. Esto se lo resume en la

Tabla 4

Tabla 4

Parámetros de entrenamiento con dos millones de muestras.

Parámetro	Valor						
SNR (dB)	-5	-4	-3	-2	-1	0	1
	3	5	8	10	12	14	15
Número de datos	2000000 por cada SNR						
Número de	26						
Tasa de validación <i>k-fold</i>	10						

Para estos nuevos catorce entrenamientos se ha obtenido los resultados de la probabilidad de detección graficados en la **Figura 16**. Donde se puede obtener las siguientes observaciones:

Desde -5 dB a -2 dB la probabilidad de detección tiene una tendencia ascendente desde un 99,9956% hasta un 99,9996%. Para este rango en general, y específicamente para -5 dB, valor en el cual se obtuvo el resultado más bajo de 99,9956%, se dice que de cada dos millones de ventanas el sistema predijo en 47 casos de forma incorrecta. Esto se ve reflejado como 3 falsos positivos (FP) y 44 falsos negativos (FN).

Dentro del mismo rango se observa también que entre menor sea la SNR, la probabilidad de detección también es menor. Y conforme la SNR suba, probabilidad de detección también lo hará. Esto es lógico, debido a que conforme la SNR es menor, la señal será mucho más baja en comparación al ruido existente. Por lo que el entrenamiento del sistema presentará más errores.

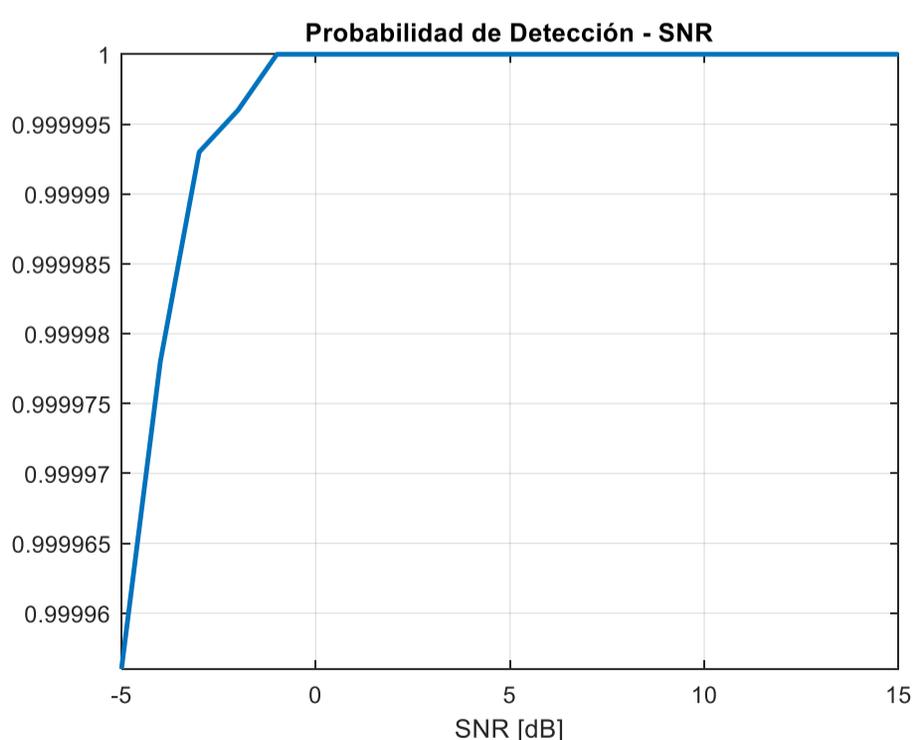


Figura 16. Probabilidad de detección vs SNR obtenidos en el bloque de pruebas 2.

Con respecto al rango de -1 dB a 15 dB , la probabilidad de detección se mantiene en 100%. Lo que quiere decir que, de los dos millones de pruebas realizadas, todos han sido predicho de manera correcta. Esto se refleja de tal manera que los falsos positivos (FP) y falsos negativos (FN) son nulos.

Los resultados obtenidos de la probabilidad de falsa alarma con respecto al SNR se muestran graficados en la **Figura 17**.

En este caso, a diferencia de la probabilidad de detección. Se observa que desde -5 dB hasta 3 dB existe variación en el resultado. Se esperaría que exista una tendencia descendente ya que a menor SNR, mayor cantidad de falsa alarma existiría y a mayor SNR, menor probabilidad de falsa alarma estaría presente.

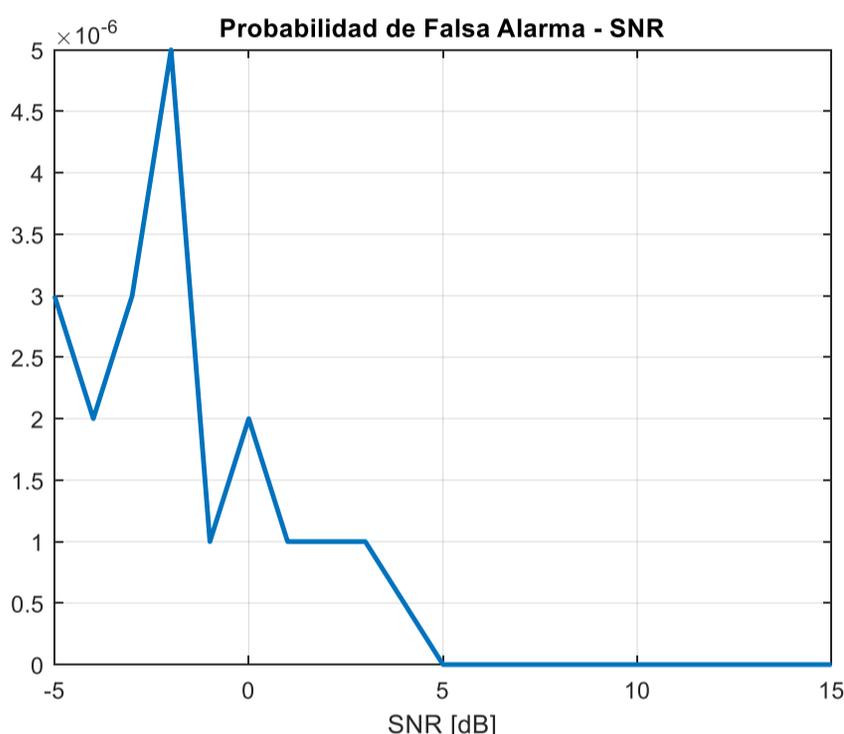


Figura 17. Probabilidad de falsa alarma vs SNR obtenidos en el bloque de pruebas 2.

Esto se justifica debido a que la cantidad de ventanas que predijo el sistema de forma incorrecta son pequeños, y su variación en algunos casos es mínima. Cabe recalcar que para estas predicciones el valor a considerar es el de los falsos positivos (FP).

Para el caso en -5 dB , el sistema predijo que en tres ventanas existía objetivo de un millón en las que no existía, y para -4 dB el sistema predijo que en dos ventanas existía objetivo de un millón en las que no existía. Los casos completos en el rango de -5 dB a 3 dB se muestra en la **Tabla 4**.

Al utilizar el método de verificación cruzada k-fold, los conjuntos de prueba son escogidos de manera aleatoria y el resultado es obtenido mediante el promedio de este subconjunto de pruebas. Con lo que, en un nuevo entrenamiento, las predicciones incorrectas mostradas en la

Tabla 5 pueden variar y ajustarse de diferente manera.

Tabla 5

Falsos positivos existentes en el rango de -5 dB a 3 dB.

<i>SNR (dB)</i>	<i>Falsos Positivos (FP)</i>
-5	3
-4	2
-3	3
-2	5
-1	1
0	2
1	1
3	1

Otro factor que se puede modificar para esperar otros resultados es el aumento de ventanas utilizadas en las bases de datos. Como ya se mencionó anteriormente, la utilización de bases de datos de dos millones de muestras se debe a las limitaciones del hardware utilizado.

Con respecto al rango de 5 dB hasta 15 dB, se ha obtenido una probabilidad de falsa alarma del 0%. Es decir, la cantidad de falsos positivos (FP) existentes es nulo.

Para los entrenamientos realizados en este bloque de pruebas el tiempo de procesamiento fue de aproximadamente 7 horas por cada uno.

Para una última prueba se utiliza una base de datos de dos millones de muestras con SNR de 0 dB, 26 características y una tasa de validación *k-fold* con un valor de *k* igual a 10. La diferencia con respecto a las pruebas anteriores es que esta base de datos consta de los valores del módulo o amplitud, calculada a partir de la parte real y la parte imaginaria, los parámetros se los resume en la **Tabla 6**.

De este último entrenamiento se ha obtenido una probabilidad de detección del 99.9804% y una probabilidad de falsa alarma de 0.000042. Lo que quiere decir que, de cada dos millones de elementos de datos, el sistema predijo en 238 de manera incorrecta. Esto se ve reflejado como 42 falsos positivos (FP) y 196 falsos negativos (FN).

Tabla 6

Parámetros de entrenamiento con dos millones de muestras.

Parámetro	Valor
SNR (dB)	0
Número de datos	2000000
Número de características	26
Tasa de validación <i>k-fold</i>	10
Datos de la base de datos	Amplitud de las componentes

De esto se puede mencionar que al trabajar con bases de datos que tienen las partes reales e imaginarias tienen una mayor probabilidad de detección y menor tasa de error con respecto a los datos que presentan la amplitud de las componentes mencionadas.

Para este caso de prueba, el tiempo del proceso fue de aproximadamente 5 horas.

4.2. Resultados con Deep Learning

El entrenamiento en Deep Learning se realizó con un clasificador *Adam* que refleja los mejores resultados para las señales de radar utilizadas en el presente trabajo, siendo implementado mediante un script en el software Matlab utilizando las funciones dentro del mismo para el clasificador de redes neuronales en Deep Learning. Al igual que con Machine Learning se utilizará una base de datos con la fase y cuadratura de la señal y con un método de validación *k-fold* con *k* igual a 10. Según la teoría de las redes neuronales los parámetros fundamentales para un desempeño óptimo son tanto la tasa de aprendizaje como el número de Hidden Units utilizados en el sistema, los parámetros utilizados se definen en la **Tabla 7**

La tasa de aprendizaje inicial utilizada es de 0.1 con un multiplicador de 0.1 por cada dos épocas donde la información pasa por las capas, este proceso se repitió por 10 épocas para poder determinar el comportamiento del entrenamiento en cada una de ellas variando la tasa de adaptación desde 0.1 hasta 1e-6 obteniendo los siguientes resultados. Al igual que con Machine Learning se utilizarán los mismos bloques de datos con las características que se muestran en la **Tabla 7**

Tabla 7

Parámetros de entrada utilizados para cinco millones de muestras

Parámetro	Valor
SNR (dB)	0
Número de datos	5000000
Tasa de validación <i>k-fold</i>	10
Número de clases	2
Épocas de entrenamiento	10
Tasa inicial de aprendizaje	0.1
Multiplicador de tasa de aprendizaje	0.1
Frecuencia de multiplicación por épocas	2
Tamaño del lote	500
Cantidad de Hidden Units	150

Los resultados obtenidos en el entrenamiento y prueba del sistema adaptativo reflejaron una probabilidad de detección del 100% y una probabilidad de Falsa Alarma de 0.0000004, es decir que de los cinco millones de muestras analizadas de la señal existe

un Falso positivo (FP) y ningún Falso Negativo (FN), teniendo un desempeño que cumple con las exigencias del sistema de radar y equiparable al método utilizado en Machine Learning, la compilación de este entrenamiento tuvo una duración de 36 horas aproximadamente siendo mayor que el entrenamiento de Machine Learning .

Al igual que para Machine Learning la limitación de tiempos de compilación y de Hardware de los equipos utilizados para las pruebas, la base de datos utilizada se reduce de 5 millones a 2 millones de muestras teniendo una variación de -5dbS hasta 15dBs utilizando los parámetros de la **Tabla 8**

Tabla 8

Parámetros de entrada utilizados para dos millones de muestras

Parámetro	Valor						
	-5	-4	-3	-2	-1	0	1
SNR (dB)	3	5	8	10	12	14	15
Número de datos	2000000						
Tasa de validación <i>k-fold</i>	10						
Número de clases	2						
Épocas de entrenamiento	10						
Tasa inicial de aprendizaje	0.1						
Multiplicador de tasa de aprendizaje	0.1						
Frecuencia de multiplicación por épocas	2						
Tamaño del lote	500						
Cantidad de Hidden Units	150						

Las catorce bases de datos utilizadas en este nuevo entrenamiento reflejan los resultados dados por la **Figura 18** y **Figura 19** respectivamente.

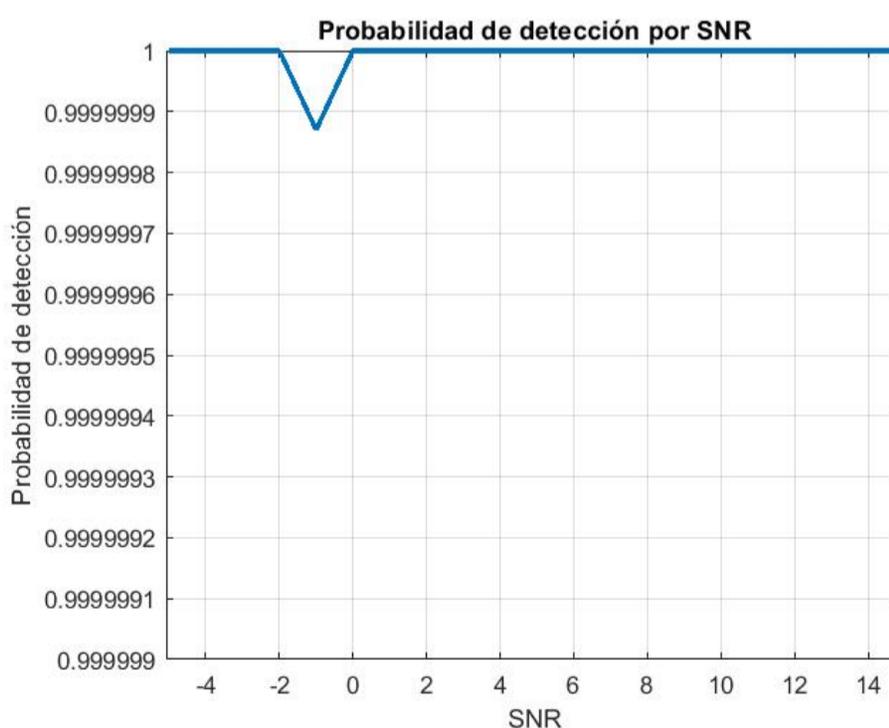


Figura 18. Probabilidad de detección vs SNR bloque de pruebas 2 (Deep Learning)

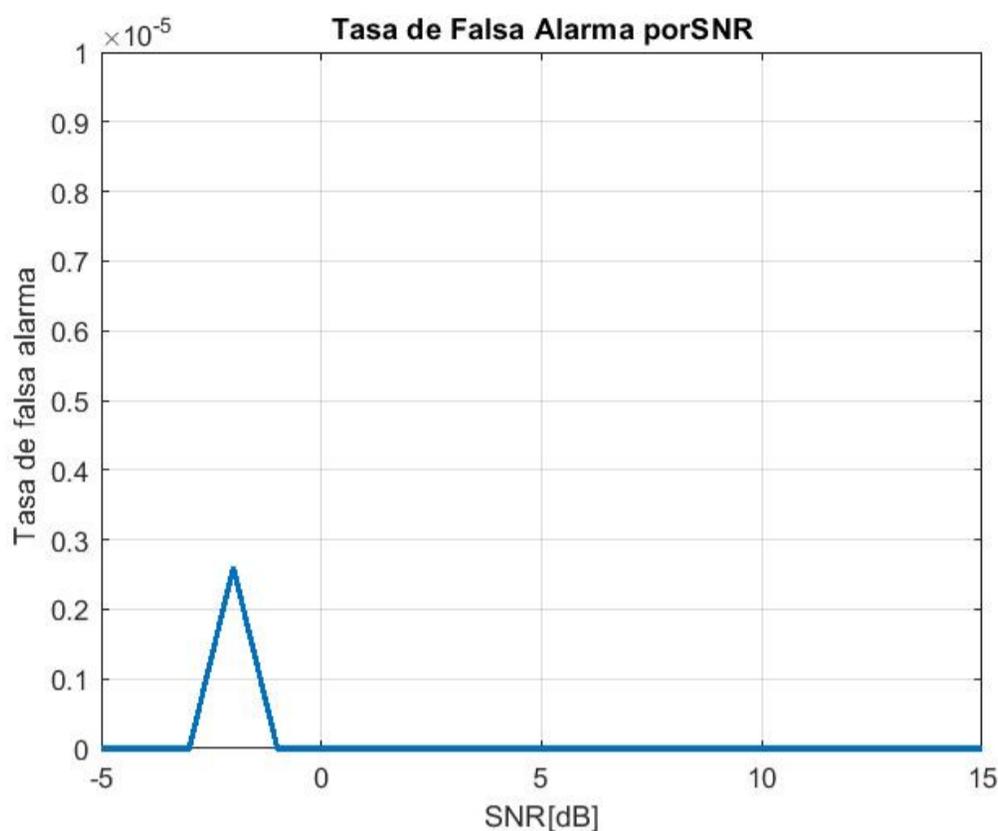


Figura 19. Probabilidad de falsa alarma vs SNR bloque de pruebas 2(Deep Learning)

Se obtiene como resultados promedios dentro de cinco ejecuciones realizadas que en la mayoría de los niveles de SNR se obtiene un total de 100% en la probabilidad de detección y una probabilidad de falsa alarma a 0 obteniendo únicamente en -2dBs una probabilidad de detección del 99.999987% y una probabilidad de falsa alarma de 0.0000026 lo cual se refleja en promedio 1 falso negativo (FN) y 2 Falsos Positivos (FP). La compilación de este proceso fue de aproximadamente 16 horas por cada nivel de ruido. Los dos picos obtenidos en -2dBs resultan como errores que pueden suscitarse estadísticamente ya que solamente se presentaron en una de las cinco compilaciones realizadas.

No se muestra una tendencia creciente como se puede observar en las pruebas realizadas con Deep Learning debido a que en la mayoría de los resultados se genera una probabilidad de falsa alarma perfecta, siendo esta una predicción incorrecta como se señaló en el caso de Machine Learning debido al uso de una base de datos de 2 millones de muestras en lugar de una ventana mayor de datos por las limitaciones de Hardware y tiempos de compilación. Se puede observar sin embargo en los resultados que el sistema cumple con un alto rendimiento y puede mejorar tanto en su probabilidad de detección como en la probabilidad de falsa alarma como se refleja al utilizar la ventana de 5 millones en el entrenamiento mostrado en la **Tabla 7**.

Para poder observar la tendencia de las curvas de rendimiento obtenidas y de igual manera comprobar la eficiencia superior de una tasa de aprendizaje adaptativo se tiene el resultado de los experimentos realizados con una tasa de aprendizaje fijo que reflejó los mejores resultados. Los parámetros iniciales de prueba se resumen en la **Tabla 9**

Tabla 9

Parámetros de entrada utilizados para las pruebas de Hidden Layers

Parámetros	Valor
Número de datos	2000000
Número de clases	2
Épocas de entrenamiento	10
Tasa inicial de aprendizaje	0.00003
Multiplicador de tasa de	Ninguno
Tamaño del lote	500
Cantidad de Hidden Units	10/20/70/150

El parámetro por variar dentro de los diferentes entrenamientos para mejorar el rendimiento de la red neuronal son las unidades ocultas o Hidden Units y se observó los resultados con varios niveles de señal a ruido obteniendo por el método de validación como se observa en la **Figura 20** y **Figura 21**

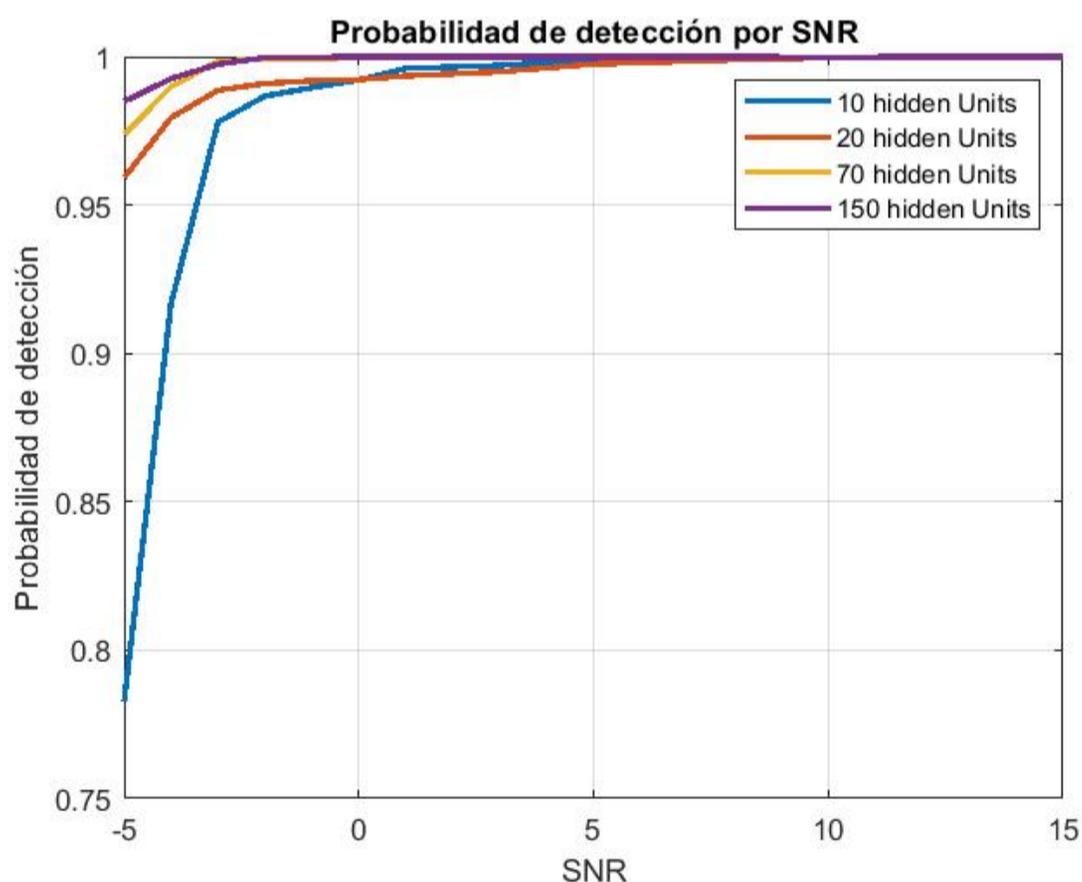


Figura 20. Probabilidad de detección con Hidden Layers y tasa de aprendizaje fija

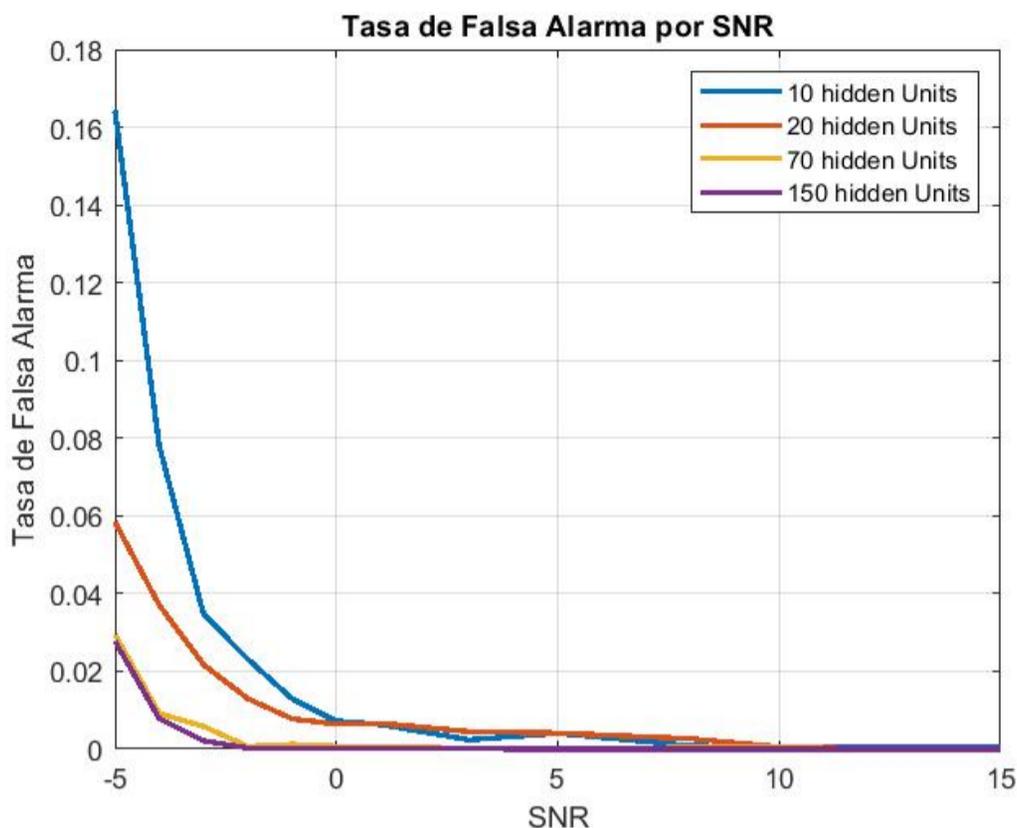


Figura 21. Probabilidad de falsa alarma con Hidden Layers y tasa de aprendizaje fija

Donde se puede apreciar una mejora en los parámetros de rendimiento no solamente en función de las Hidden Units utilizadas en cada entrenamiento, sino que también se puede apreciar como el sistema obtiene mejores resultados al tener una señal de objetivo de radar de mayor relación señal a ruido facilitando de esta manera la extracción automática de las características y la asignación del reconocimiento de estas al clasificador.

Tanto como en la probabilidad de falsa alarma como en la probabilidad de detección se puede observar que el número de Hidden Units determina un factor importante en el rendimiento del sistema, manteniéndose en valores altos al utilizar 150 Hidden Units, la afectación se puede observar en los niveles bajos de SNR como en el caso de 10 Hidden Units siendo probabilidad de detección del 78,2674% y una probabilidad de falsa alarma del 0.16427.

Se observa claramente que es necesario un número elevado de Hidden Units para un mejor rendimiento, en este caso 150 Hidden Units reflejan en sus resultados una probabilidad de detección del 98,4909% y una probabilidad de falsa alarma de 0.027757 en el nivel de -5dBs, a partir de -4dBs los resultados son mayores al 99% y menores al 0.01 en la probabilidad de detección y de falsa alarma respectivamente, llegando a tener a partir de 0dBs valores superiores a 99.9981% y 0.000125 hasta llegar 100% y 0%.

En ambos parámetros en los niveles más altos de SNR, los resultados obtenidos a partir de 0dBs cumplen con las exigencias de los sistemas de radar siendo evidente la necesidad de usar el aprendizaje adaptativo para la mejora del rendimiento en los niveles de SNR menores a 0dBs.

De igual manera que en Machine Learning se realizó la prueba con una base de datos con 2 millones de muestras y un SNR de 0dBs, pero en este caso se utilizó el módulo calculado a partir de la parte real e imaginaria lo cual se resumen en la **Tabla 10**.

Tabla 10

Parámetros de entrada utilizados para dos millones de muestras (Amplitud)

Parámetro	Valor
SNR (dB)	0
Número de datos	2000000
Tasa de validación <i>k-fold</i>	10
Número de clases	2
Épocas de entrenamiento	10
Tasa inicial de aprendizaje	0.1
Multiplicador de tasa de	0.1
Frecuencia de multiplicación por	2
Tamaño del lote	500
Cantidad de Hidden Units	150

De este experimento se determinó una probabilidad de detección del 99,9951% y una tasa de error del 0.00002 lo que quiere decir que, de un total de 2 millones de muestras, el sistema detectó 20 casos de falsos positivos (FP) y 49 casos de falsos negativos (FN). Con estos resultados se corroboró lo obtenido con los experimentos realizados en Machine Learning observando que tanto la probabilidad de detección como la probabilidad de falsa alarma se obtienen mejores resultados cuando se utiliza la parte real e imaginaria en comparación que con el uso de la amplitud de estas componentes.

La compilación de esta prueba fue de 11 horas aproximadamente.

De igual manera los resultados generados en Deep Learning pueden variar debido a que se utiliza la validación *k-fold* aleatoria, aunque se espera que en diferentes ejecuciones se tenga resultados de la misma tendencia.

4.3. Comparación entre Machine Learning y Deep Learning

Dentro de los experimentos realizados y sus resultados mostrados en el presente capítulo se puede apreciar que tanto Machine Learning como Deep Learning tienen un

alto desempeño y reflejan resultados que cumplen con las exigencias de un sistema de detección de radar como se puede reflejar en la **Tabla 11**

Tabla 11

Comparación entre Machine y Deep Learning con 5 millones de muestras

	Machine Learning	Deep
Probabilidad de detección	100	100
Probabilidad de falsa	0.0000008	0.0000004
VP	2500000	2500000
VN	2499999	2499999
FN	0	0
FP	2	1

Como se puede observar Deep Learning y Machine Learning tuvieron resultados similares con una ligera mejora en Deep Learning con un Falso Positivo menos que en su contraparte de Machine Learning.

Con respecto al tiempo los resultados se pueden reflejar en la **Tabla 12**

Tabla 12

Tiempos de compilación de los algoritmos de Machine y Deep Learning.

Tiempos de ejecución (horas)			
	Machine	Deep	Comparación
5 millones	24	52	216.6666667
2 millones (fase y	7	16	228.5714286
2 millones (amplitud)	5	11	220

Como se puede observar el algoritmo de Deep Learning tuvo una duración del 220% aproximadamente más que Machine Learning siendo evidente la ventaja que este último tiene. Sin embargo, en Machine Learning se requiere de un preprocesamiento que consiste en realizar una extracción de características, siendo estas las variables de entrada para el entrenamiento.

Por otra parte, los algoritmos utilizados con Deep Learning si bien tienen un tiempo mayor de entrenamiento, no utilizan ningún tipo de preprocesamiento ni extracción de características de las señales a utilizar, sino que autoajustan los pesos de sus distintas capas de red neuronal con el paso de los datos por estas. Otra ventaja de Deep Learning es su potencial de reflejar mejores resultados mientras se varíen sus parámetros de entrenamiento con un número mayor de Hidden Layers, épocas de paso de los datos, una tasa variable de aprendizaje diferente, etc. Si bien estos cambios pueden aumentar la complejidad de la red y por ende el tiempo de entrenamiento y el uso de la memoria

del sistema, los resultados obtenidos con esta red más compleja tienden a generar resultados de mayor rendimiento en el clasificador, siendo el tiempo y la potencia del Hardware las mayores limitantes.

4.4. Comparación de Machine, Deep Learning y algoritmos bayesianos

Finalmente se puede observar los resultados tanto de Machine Learning como Deep Learning en contraste con los algoritmos de detección anteriores a la investigación (Lara & Ortiz, 2019) teniendo los resultados en la **Figura 22** donde se observa la probabilidad de detección del sistema y cuya comparación se puede resumir en la **Tabla 13**, siendo el algoritmo de AR LMS MEAN el que posee un mejor rendimiento al observar su curva de desempeño.

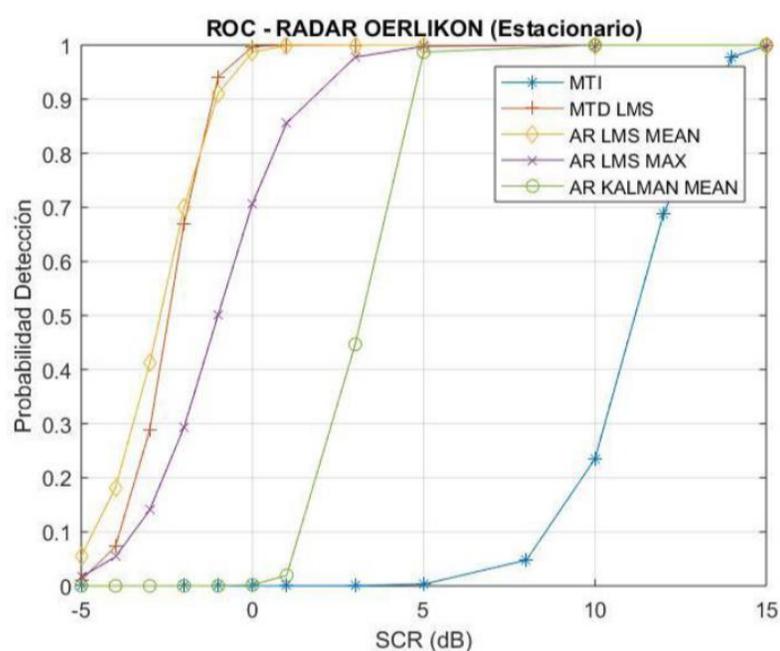


Figura 22. Resultados obtenidos con algoritmos anteriores a la investigación

Tabla 13

Comparación de la probabilidad de detección entre algoritmos de Machine, Deep Learning y AR LMS MEAN en los distintos niveles de SNR

SNR	Machine	Deep	AR LMS
-5	0.9999765	1	0.01656
-4	0.999988	1	0.05383
-3	0.999995	1	0.14027
-2	0.9999955	1	0.29305
-1	0.9999995	0.99999987	0.50146
0	0.999999	1	0.70681
1	0.9999995	1	0.85614
3	0.9999995	1	0.97782
5	1	1	0.99781
8	1	1	0.99987
10	1	1	1
12	1	1	1
14	1	1	1
15	1	1	1

Como se puede observar Machine y Deep Learning con sistemas que presentan una robustez mayor ante niveles de SNR bajos en contraste con la técnica de AR LMS MEAN que presenta un desempeño mucho menor en niveles negativos. Esto se debe a que el procesamiento de los dos sistemas estudiados en el presente trabajo es más complejo y analizan por lo tanto más características de las señales pudiendo de esta manera reflejar resultados mayores en ambientes donde la calidad de la señal es menor.

5. CAPÍTULO V: CONCLUSIONES

Tanto en Machine Learning como en Deep Learning, los resultados obtenidos al utilizar las bases de datos de la parte real y la parte imaginaria de la señal de radar son mejores que al utilizar solamente la amplitud de ambas componentes. Para el caso de prueba de 0 dB en Machine Learning, la probabilidad de detección mejora un 0.0196% hasta llegar a un 100%; mientras que la tasa de error disminuye en 0.00004 hasta llegar a 0.000002. Esta tendencia se repite en el caso de Deep Learning donde el utilizar la parte real e imaginaria genera resultados más elevados que su contraparte con el uso de la amplitud de la señal, en este caso con una mejora en un 0.0049% hasta llegar al 100% en la probabilidad de detección y la tasa de error desde 0.00002 hasta llegar a 0.

Los resultados obtenidos con Deep Learning son equiparables a aquellos de Machine Learning como se puede observar en la **Tabla 11** y la **Tabla 13** donde hay una mejora promedio del 100.000456% en los niveles menores de SNR. Sin embargo, hay que tomar en cuenta que la velocidad de entrenamiento de Machine Learning es mayor a la de Deep Learning, por lo que los algoritmos Machine Learning son más rápidos al momento de obtener un resultado, en cualquiera de los casos.

La principal ventaja de Deep Learning sobre Machine Learning es la autonomía del sistema al poder trabajar directamente con la señal de entrada sin necesidad de la extracción de características y por ende un preprocesamiento inexistente de la señal.

Deep Learning como se demostró en los resultados obtenidos tiene el potencial de obtener mejores resultados conforme se incrementa la complejidad del sistema, siendo el coste de estos resultados el tiempo de entrenamiento necesario para que la red neuronal se entrene y logre un mejor desempeño, sin embargo este problema se suscita solamente en el proceso de entrenamiento del sistema ya que una vez entrenados los sistemas se espera que tanto Machine como Deep Learning funcionen en tiempo real, siendo este último el que refleja mejores resultados.

Los resultados obtenidos tanto en Machine Learning como en Deep Learning superan notablemente a los métodos de detección tradicionales utilizados previos al desarrollo de Machine y Deep Learning, especialmente en niveles de SNR bajos donde ambos métodos de redes neuronales presentan una robustez mayor. La capacidad de ambos sistemas de mantener una probabilidad de detección y probabilidad de falsa alarma constantes es una característica deseada y difícil de lograr, siendo Machine Learning y Deep Learning un avance importante en los sistemas de detección de radar.

6. TRABAJOS FUTUROS

El estudio del presente trabajo demostró que las técnicas de Machine Learning y Deep Learning reflejan resultados que cumplen con las exigencias de un sistema de señales de radar, siendo robustos ante niveles de SNR bajos. Los resultados sin embargo fueron realizados con una base de datos de un simulador de señales de radar.

Una nueva línea de investigación a futuro sería el realizar las pruebas del entrenamiento de ambos sistemas y medir su rendimiento con las señales reales obtenidas, trabajando por lo tanto con diferentes niveles de SNR; el objetivo de esta línea de investigación será el de probar a los sistemas como clasificadores adaptables ante diferentes SNRs, volviendo al sistema más robusto

La siguiente línea de investigación adicional sería la aplicación de ambos sistemas en tiempo real manejando las señales directamente de un radar y determinando la presencia o no de un objetivo al instante.

7. BIBLIOGRAFÍA

- Álvarez Pastor, M. (2016). *Identificación de objetos mediante el análisis de una señal radar. Desarrollo de un modelo de simulación cooperativo*. Madrid - España: Universidad Politécnica de Madrid.
- Bandinera, F., De Maio, A., Greco, A., & Ricci, G. (2007). IEEE Transactions on Signal Processing Vol. 55, NO. 4, April 2007 1223 Adaptive Radar Detection of Distributed Targets in Homogeneous and Partially Homogeneous Noise Plus Subspace Interference. *IEEE Transactions on Signal Processing*, 1223-1237.
- Barton, D. K. (2005). *Radar System Analysis and Modeling, Volumen 1*. Boston - London: Artech House.
- Bolton, W. (1995). *Mediciones y Pruebas Eléctricas y Electrónicas*. Barcelona - España: Marcombo.
- Bonaccorso, G. (2017). *Machine Learnings AlgorithmsÑ A reference guide to popular algorithms for data science and machine learning*. Birmingham - Inglaterra: Packt Publishing.
- Broesen, P. M. (2006). *Automatic Autocorrelation and Spectral Analysis*. Londres: Springer.
- Cuartas, J. A., Madrigal, P. A., & Torres, R. A. (2013). Interfaz Cerebro Computador Controlada por Sincronización y Desincronización Relacionada a Eventos en Sujetos no Entrenados. *IFMBE Proceedings* 33, 611-614.
- Donges, N. (22 de febrero de 2018). *The Random Forest Algorithm*. Obtenido de Towards Data Science: <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>
- Emery, W. J., & Thomson, R. E. (2004). *Dara Analysis Methods in Physical Oceanography*. Amsterdam: Elsevier.
- Gini, F., & Farina, A. (2002). Vector Subspace Detection in Compound-Gaussian Clutter Part 1: Survey and New Results. *IEEE Transactions on Aerospace and Electronic Systems*, 1295-1311.
- Gras, J. A. (2001). *Diseños de series temporales: técnicas de análisis*. Barcelona: EDICIONS UNIVERSITAT DE BARCELONA.
- Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications*. Springer.

- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., . . . Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine* .
- Hurwitz, J., & Kirsch, D. (2018). *Machine Learning for dummies*. Hoboken: John Wiley & Sons, Inc.
- Jokanovic, B., Amin, M., & Ahmad, F. (2016). Radar Fall Motion Detection Using Deep Learning. *IEEE radar conference (RadarConf)*, 1-6.
- Kelly, E., & Forsythe, J. (1989). *Adaptative Detection and Parameter Estimation for Multidimensional Signal Model*. Massachusetts: Massachusetts Institute of Technology Lincoln Laboratory.
- Kingma, D., & Lei Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint* .
- Kurbiel, T., & Khaleghian, S. (2017). Training of Deep Neural Networks based on Distance Measures using RMSProp. *arXiv preprint*.
- Lara, F., & Ortiz, M. (2019). *Procesador Bayesiano de Señales de Radar para el Sistema Oerlikon*. Sangolquí - Ecuador: Universidad de las Fuerzas Armadas - ESPE.
- Le, H.-S., Oparin, I., Allauzen, A., Gauvain, J.-L., & Francois, Y. (2011). Structured Output Layer Neural Network Language Model. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (págs. 5524-5527).
- LeCun, Y., Bengio, Y., & Geoffrey, H. (2015). Deep Learning . *Nature* , 436-444.
- Manson, E., Yonel, B., & Yazici, B. (2017). Deep Learning for radar. *IEEE Radar Conf.*
- Martens, J. (2010). Deep learning via Hessian-free optimization. *ICML*.
- Mataix, C., & Rivadulla, A. (2002). *Física Cuántica y Realidad*. Madrid: Complutense.
- MathWorks. (2015). *Classification Learner*. Obtenido de MathWorks - Documentarion: <https://la.mathworks.com/help/stats/classificationlearner-app.html?lang=en>
- MathWorks. (2018). *Choose Classifier Options*. Obtenido de MathWorks - Documentation: <https://la.mathworks.com/help/stats/choose-a-classifier.html>
- Mathworks. (2019). *Long Short-Term Memory Networks*. Obtenido de <https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html;jsessionid=f504e7a307e098af386700d5ce88>
- Mathworks. (2019). *Mathworks: Long Short-Term Memory (LSTM)*. Obtenido de <https://www.mathworks.com/discovery/lstm.html>

- Mathworks. (2019). *Mathworks: Sequence Classification Using Deep Learning*. Obtenido de <https://www.mathworks.com/help/deeplearning/examples/classify-sequence-data-using-lstm-networks.html>
- Moujahid, A., Inza, I., & Larrañaga, P. (1997). *Tema 6: Evaluación de Modelos de Clasificación Supervisada*. Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco.
- Parikh, R., Mathai, A., Parikh, S., & Thomas, R. (2008). Understanding and using sensitivity, specificity and predictive values. *Indian Journal of ophthalmology*, 45.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint*, 1609.04747.
- Sainath, T., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional Long Short Term Memory Full Connected Deep Neural Network. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (págs. 4580-4584).
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth annual conference of the international speech communication association*.
- Sassi, O., Sellami, L., Slima, M., Chrourou, K., & Hamida, A. (2012). Improved Spatial Gray Level Dependence Matrices for Texture Analysis. *International Journal of Computer Science & Information Technology*, 209-220.
- Sinminyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint*.
- Sundermeyer, M., Schluter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. *Thirteenth annual conference of the international speech communication association*.
- Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *International conference on machine learning*, (págs. 1139-1147).
- Tiros, M., Bueno, A., García, L., & Benítez, C. (2018). A deep neural networks approach to automatic recognition systems for volcano-seismic events. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1533-1544.
- Upton, G., & Cook, I. (1996). General summary statistics. *Understanding Statistics*, 36-74.

Wong, T.-T. (2015). Performace evaluation of classification algorithms by k-fold and leave-one-out cross validation . *Pattern Recognition* , 2839-2846.

Zhang, G., Rong, H., Hu, L., & Jin, W. (2004). Entropy Feature Extraction Approach for Radar Emitter Signals. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 621-625.