

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación.

DISEÑO, IMPLEMENTACIÓN Y COMPARACIÓN DEL RENDIMIENTO DE ALGORITMOS DE PREDICCIÓN DE ESPECTRO, UTILIZANDO MÉTODOS ESTADÍSTICOS Y DE MACHINE LEARNING.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

María Carolina Morales Aguilar.

Erick Andres Díaz Loayza.

GUAYAQUIL-ECUADOR

Año: 2021

DEDICATORIA

Dedico con todo mi corazón mi tesis a mis padres Dayse y Ruffo, pues sin ellos no lo habría logrado, gracias a su sacrificio, esfuerzo y sobre todo por creer en mí.

A mi novio Maickol por estar ahí para mí incondicionalmente, por brindarme apoyo siempre y nunca dejarme sola.

A mis hermanos Jonathan y Gisela por apoyarme e inspirarme a lograr lo que ya lograron ustedes.

A mis sobrinos Keyla y Mateo por ser luz en mi vida y el mejor regalo de la vida.

Por último, dedico mi tesis a todos mis compañeros que me apoyaron durante toda la carrera, gracias por ayudarme a lograrlo.

María Carolina Morales Aguilar

DEDICATORIA

Esta tesis se la dedico a toda mi familia que siempre estuvo a mi lado apoyándome en todo lo que necesitaba, especialmente a mis padres ya que sin ellos no lo hubiera conseguido.

A mis tíos y abuelitos que fueron mis segundos padres cuando algo me faltaba, a mis hermanos por estar siempre presente en todo.

En si a todas las personas que me brindaron su apoyo cuando más lo necesitaba, ya que sin ellos no habría logrado esta meta tan importante para mi vida.

Erick Andres Díaz Loayza

AGRADECIMIENTOS

A Dios por estar siempre para mí y ayudarme a guiar por el buen camino.

A mis padres Dayse y Ruffo, por estar siempre pendientes de mí, por todo el sacrificio y amor durante toda la carrera

A mis hermanos por apoyarme siempre en todas mis decisiones.

A mis amigos por apoyarme siempre en todas las materias y hacer de mi paso por la universidad algo más ameno.

A Andres por trabajar juntos en este logro, por su paciencia y comprensión.

María Carolina Morales Aguilar

AGRADECIMIENTOS

Principalmente a Dios por guiarme en todo el camino recorrido hasta ahora, por apoyarme y darme fortaleza en los momentos más difíciles que pase

A mis padres por haberme enseñado a confiar en mí mismo y nunca darme por vencido, por los consejos y valores que me impartieron desde pequeño.

A mi compañera de tesis María Morales que sin ella no habiéramos podido completar este logro tan importante en nuestras vidas.

A todos mis profesores que me impartieron sus conocimientos para cada día avanzar un poco más.

Erick Andres Díaz Loayza

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; MARIA CAROLINA MORALES AGUILAR y ERICK ANDRÉS DÍAZ LOAYZA, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



María Carolina Morales Aguilar

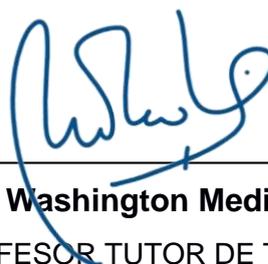


Erick Andrés Díaz Loayza

EVALUADORES



Msc. Washington Medina Moreira
PROFESOR DE LA MATERIA



Msc. Washington Medina Moreira
PROFESOR TUTOR DE TESIS

RESUMEN

En el trabajo presentado a continuación se diseñó un algoritmo capaz de predecir los canales disponibles en el espectro a través del método de machine learning el cual se basa en redes neuronales recurrentes. En la implementación de este algoritmo se utilizó el lenguaje de programación Python. Para evaluar su efectividad se lo comparó con un método tradicional estadístico como es ARIMA, algoritmo que fue diseñado en dos diferentes lenguajes de programación que son RStudio y Python.

En el primer capítulo se detalla la problemática que llevó a realizar la investigación del trabajo, para esto se investigó varias soluciones ya desarrolladas a este problema. Por último, se elaboró una solución para la predicción de los canales disponibles en el espectro. En el segundo capítulo se detallaron todos los conceptos posibles para poder entender el funcionamiento del espectro y de la radio cognitiva.

En el tercer capítulo se presenta el desarrollo de la solución tanto para el algoritmo de machine learning (LSTM) como para el algoritmo estadístico (ARIMA), donde se muestra el código desarrollado para cada método con una respectiva descripción de cómo funciona cada línea de código. Para el cuarto capítulo se obtuvieron los resultados de la predicción de los canales disponibles, en donde se encontró tres parámetros importantes como el accuracy, el tiempo de ejecución y el RMSE de cada algoritmo. Al final se realizó la comparación de cada método de predicción para determinar que método es más eficiente.

ABSTRACT

In the work presented below, an algorithm capable of predicting the available channels in the spectrum is improved through the machine learning method which is based on recurrent neural networks. In the implementation of this algorithm, the Python programming language was changed. To evaluate its effectiveness, it was compared with a traditional statistical method such as ARIMA, an algorithm that was designed in two different programming languages: RStudio and Python.

The first chapter details the problems that led to the investigation of the work, for this several solutions already developed to this problem were investigated. Finally, a solution for the prediction of the available channels in the spectrum was elaborated. In the second chapter, all possible concepts were detailed to understand the functioning of the spectrum and cognitive radio.

In the third chapter, the development of the solution for both the machine learning algorithm (LSTM) and the statistical algorithm (ARIMA) is presented, where the code developed for each method is shown with a respective description of how each line of code works. For the fourth chapter, the results of the prediction of the available channels will be obtained, where three important parameters were found, such as the precision, the execution time and the RMSE of each algorithm. At the end, the comparison of each prediction method was made to determine which method is more efficient.

INDICE GENERAL

DEDICATORIA	2
DEDICATORIA	3
AGRADECIMIENTOS	4
DECLARACIÓN EXPRESA	6
EVALUADORES	7
CAPITULO 1	13
1.1 DEFINICIÓN DEL PROBLEMA	13
1.2 JUSTIFICACIÓN DEL PROBLEMA	14
1.3 OBJETIVOS	15
1.3.1 OBJETIVO GENERAL	15
1.3.2 OBJETIVOS ESPECÍFICOS	15
1.4 ESTADO DEL ARTE	16
1.5 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	20
CAPITULO 2	22
2.1 MARCO TEÓRICO	22
2.2 Espectro radioeléctrico	22
2.2.1 Conceptos básicos utilizados en el espectro radioeléctrico	23
2.2.2 Uso eficiente del espectro	24
2.3 Parámetros técnicos para la utilización del espectro radioeléctrico	25
2.3.1 Frecuencia	25
2.3.2 Propagación	26
2.3.3 Ancho de banda	27
2.3.4 Potencia	27
2.3.5 Interferencia	27
2.4 Radio cognitiva	28
2.4.1 Regulación y estandarización en radio cognitiva	29
2.4.2 Ahorro de energía en radio cognitiva	30
2.5 Acceso dinámico al espectro	31
2.6 Acceso oportunista al espectro (OSA)	31
2.7 Handoff espectral	32
2.7.1 Causas para realizar un Handoff espectral	32

2.7.2	Requerimientos para realizar un Handoff espectral	33
2.7.3	Procedimiento para poder realizar un Handoff espectral exitoso	33
2.8	Importancia del análisis predictivo	34
2.8.1	El desarrollo de la big data.....	34
2.8.2	Aumento de la competencia	34
2.8.3	Tecnologías innovadoras para el uso del big data	35
2.8.4	Ejemplos de áreas donde se utiliza el análisis predictivo.....	35
2.8.5	Análisis predictivo con MATLAB	35
2.9	Python para predicción de datos	36
2.10	Huecos espectrales.....	36
2.11	Método estadístico ARIMA	38
2.12	Método Maching learning LSTM.....	38
2.13	Series de tiempos.....	39
2.14	Redes Neuronales	41
CAPITULO 3	42
3.1	Diseño del Proyecto.....	42
3.1.1	Diagrama de bloques básico	42
3.1.2	Diagrama de bloques detallado.....	45
3.2	Método estadístico (ARIMA)	46
3.2.1	Creación del algoritmo ARIMA	47
3.3	Método machine learning (LSTM)	51
3.3.1	Creación del algoritmo LSTM.....	52
CAPITULO 4	65
4.1	Resultados.....	65
4.2	Método LSTM.....	65
4.3	Método ARIMA.....	67
4.4	Comparación de los métodos LSTM y ARIMA	68
CONCLUSIONES	73
RECOMENDACIONES	75
BIBLIOGRAFÍA	76
ANEXOS	80

ABREVIATURAS

LSTM
ARIMA
PU
SU
OSA
EWMA
HNN
RNN
SVM
BS
MCDM
SAW
GRA
UIT-R
WRAN
IEEE
RC
DSA
QoS
TI
RMSE

CAPITULO 1

1.1 DEFINICIÓN DEL PROBLEMA

Con el paso del tiempo los usuarios que utilizan el espectro han ido creciendo exponencialmente, por ejemplo, el tráfico de la telefonía móvil se incrementó 4 veces desde el 2010, esto se debe a que cada año mayor cantidad de personas tiene la tecnología para utilizar el espectro [1]. El espectro tiene una cantidad limitada y por tanto entre más usuarios se tenga este no será suficiente para abastecer a todos y aquí se producirá una congestión haciendo que los usuarios no puedan utilizar los dispositivos para conectarse al espectro y por tanto no podrán transmitir ni recibir información [2].

Por otra parte la red en la actualidad asciende aproximadamente un 7,3% y esto logra unos 4600 millones de usuarios conectados a internet en el 2021, que corresponde al 59,5% de la población [3], los usuarios que tienen dispositivos móviles alcanzan un porcentaje de 66,6 % de la población mundial que comparado con el 2020 se registra un aumento de 1,8% que corresponde a un incremento de 93 millones de usuarios, este crecimiento se debe por la pandemia de Covid-19 ya que la educación virtual y el trabajo online se hicieron un diario vivir para muchas personas [4].

Debido al crecimiento de usuarios, siendo el espectro un recurso limitado cada vez se congestiona más, provocando escases de espectro, para solucionar este problema hay diversas investigaciones, una de las investigaciones más conocida es la radio cognitiva que utiliza el concepto del acceso oportunista al espectro (OSA), que establece la posibilidad de que los usuarios cognitivos sin licencia accedan a las frecuencias de radio a través de un agujero del espectro [5], OSA provee canales a los usuarios por medio de arquitecturas inalámbricas heterogéneas [6].

La detección en OSA introduce retrasos de tiempo que reducen la precisión de los resultados de la detección, es decir, si un usuario va a hacer uso de este acceso dinámico

en tiempo real pueden existir colisiones, por ende, hay una pérdida de tiempo. Según estudios realizados el modelo de movilidad de espectro tradicional siempre da como resultado colisiones de transmisión ya que un usuario de radio cognitiva no evacúa su canal actualmente ocupado antes de la detección de los usuarios principales [7]. Por ello para aprovechar el espectro evitando retardos y colisiones entre los usuarios se ha propuesto la predicción de espectro como solución.

La predicción del espectro también minimiza el consumo energético utilizado en la radio cognitiva, esto se debe a que los algoritmos utilizados predicen los canales disponibles y por tanto este trabajo ya no lo realiza la radio cognitiva [8]. Comparaciones entre una radio convencional y una radio cognitiva demuestra que el método de predicción del espectro es capaz de ahorrar entre 20% a 40% de energía comparado con la radio convencional [9].

1.2 JUSTIFICACIÓN DEL PROBLEMA

Los motivos que llevaron a realizar el diseño y comparación del rendimiento de predicción de espectro del método estadístico con el método de machine learning, se basan en que la detección de espectro en OSA produce atrasos de tiempo lo que provoca una disminución en la exactitud de los resultados, también produce un gran consumo de energía al momento de detectar los canales disponibles y los usuarios primarios [10].

Por lo que a través de este proyecto al implementar la predicción de espectro se tendrá varias ventajas entre las que se pueden destacar son un significativo ahorro de energía, reducción de costos en la construcción de los dispositivos, ahorro de tiempo y cuidado del medio ambiente. Esto es de suma importancia para el planeta debido a que el crecimiento del consumo energético crece cada vez más [11]. Por ello con este proyecto se busca hacer un uso racional y eficiente de energía, poniendo como prioridad el cuidado del ecosistema.

Una de las ventajas mencionadas anteriormente son el ahorro de tiempo lo cual proporciona un gran beneficio ya que a través de la predicción se puede evitar la mayor cantidad de colisiones entre los usuarios primarios y secundarios, dando así una solución inmediata de descongestionamiento en el espectro radioeléctrico al reutilizar los canales desocupados sin pérdida de tiempo ya que no se estará realizando la detección en tiempo real.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

- Diseñar, implementar y comparar los algoritmos para la predicción de espectro, utilizando el método estadístico (ARIMA) con el método de machine learning (LSTM), mediante el uso del programa Python.

1.3.2 OBJETIVOS ESPECÍFICOS

- Descubrir información sobre el comportamiento del espectro en diferentes canales, la cual servirá como base de datos de los algoritmos LSTM y ARIMA.
- Investigar cómo funciona el método estadístico de ARIMA para la predicción.
- Diseñar el código del algoritmo estadístico (ARIMA) a través de la herramienta de Python.
- Determinar los resultados de predicción de espectro del algoritmo ARIMA.
- Investigar el método de machine learning LSTM para la predicción.
- Diseñar el código del algoritmo machine learning (LSTM) a través de la herramienta de Python.
- Determinar los resultados de predicción de espectro del algoritmo LSTM.
- Determinar el modelo más eficiente con respecto al tamaño de datos.

- Determinar el tiempo que se demora cada método en realizar el proceso de predicción del espectro.
- Evaluar la probabilidad de predicción que posee cada algoritmo.
- Demostrar que método es más eficiente al momento de predecir los huecos espectrales, tomando en cuenta los parámetros antes evaluados.

1.4 ESTADO DEL ARTE

En India, la Universidad de Ciencia y Tecnología de Kochi ha realizado un enfoque simple basado en el teorema bayesiano para predecir el estado de ocupación del espectro del usuario primario (PU) [12]. Su rendimiento se compara con el método basado en la media móvil ponderado exponencial (EWMA), para predecir la información de ocupación del espectro. Se encontró que el enfoque bayesiano tiene un desempeño de predicción de probabilidad que necesita mejorar.

En Singapur la Escuela de Ingeniería Informática y la Universidad Tecnológica de Nanyang diseñó un predictor de estado del canal utilizando un esquema adaptativo, es decir un modelo de Márkov oculto (HMM). Se realizó un análisis de rendimiento del esquema de predicción del estado de los canales y se investigó la precisión del esquema de predicción. Se pudo observar que la predicción del estado del canal es conveniente para el acceso oportunista al espectro [13].

La Universidad Distrital implementó un método de predicción de espectro basado en árboles de decisión donde se predice el valor de las distintas clases que se tienen como objetivo, este funciona a través de varias variables existentes de entrada [6]. Se puede observar que el método es bastante útil, debido a que aporta con la predicción del estado del canal, evitando así el retraso en la búsqueda de canales disponibles, la utilidad de este método estadístico va a depender de las distintas necesidades que se desean lograr.

En Cuba, la Universidad Tecnológica “José Antonio Echeverría” realiza un modelo de cooperación, para un censado cooperativo en el acceso oportunista al espectro basado en Teoría de Juegos, específicamente en juegos de formación de colisiones. El algoritmo fue implementado en Matlab y se comprobó su funcionamiento con 10, 15 y 50 usuarios secundarios (SU). Además, se implementó en C++ para incluirlos en la plataforma del software HEATS-RT, software para la vigilancia del espectro radioeléctrico diseñado para un grupo de investigación [14].

Una tesis presentada a la Facultad de Old Dominion University de Norfolk realizó una investigación sobre las herramientas de aprendizaje automático, incluidas las redes neuronales recurrentes (RNN), para problemas de radio cognitivos [15]. Se demostró que las redes neuronales recurrentes son útiles para la predicción del espectro espacial con datos faltantes, es decir un RNN funciona bien para problemas de datos faltantes en el dominio de tiempo-frecuencia del espectro de radio cognitiva.

Un estudiante de doctorado en Ingeniería en la ciudad de Bogotá implementó Support vector machine (SVM), para predecir el comportamiento de los PU, donde se plantea una forma de ingresar con anticipación los pedidos de los SU en la Base Station (BS). Así se reduce el tiempo que le toma para poder asignar el espectro y crear diferentes posibilidades para la selección de canales eficientes, los cuales toman como referencia la clasificación que tiene el espectro con ayuda de los métodos de aprendizaje SVM [16]. De acuerdo con los resultados obtenidos se puede decir que este sistema no es óptimo para predecir en radio cognitiva, es decir este modelo es capaz de predecir como se comportará un usuario primario, pero de una forma errónea.

En Bogotá se ha realizado un proyecto curricular en la especialidad de ingeniería sobre la predicción de espectro utilizando métodos de aprendizaje por refuerzo. Este método al ser no supervisado permite trabajar modificando el comportamiento cuando este se encuentra en un ambiente no conocido, esta forma de trabajar está basada en el principio de Bellman. Como resultado se obtuvo que la implementación de este método mejoró el rendimiento de los SU en entornos dinámicos [17].

Un artículo científico publicado en la ciudad de Bogotá realiza un algoritmo que permite elegir los canales que tengan una frecuencia libre en una red para que se pueda acceder oportunamente al espectro, basado en el sistema de Colonias de Abejas Artificiales. Al aplicar este método se obtuvo una disminución del 62% en los respectivos tiempos de operación y una reducción del 36% con respecto al número de handoffs fallidos, al ser estimada en una colonia de abejas con una cantidad de 50 a 450 [18].

La universidad Distrital ubicada en el país de Colombia propone algunos algoritmos para la asignación del espectro entre ellos está la lógica difusa que es una herramienta utilizada para mejorar el manejo del espectro solo cuando se conoce la ubicación exacta de los canales y el uso de estos [19], así el usuario secundario puede conectarse sin ningún problema a los huecos del espectro que da el método de la lógica difusa. Este sistema tiene dos controladores difusos que son planteados e implementados en la Figura 1.1.

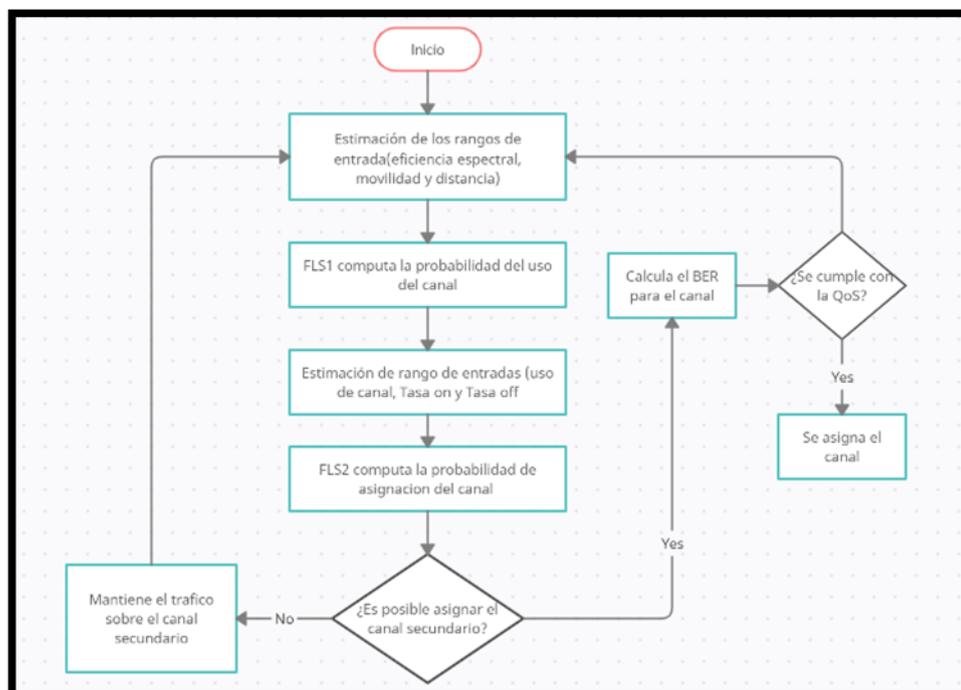


Figura 1.1. Diagrama de flujo del método de lógica difusa utilizado en radio cognitiva [19]

En una universidad de la ciudad de Cuba, se realizó una investigación sobre los algoritmos genéticos los cuales fueron desarrollados e inspirados en el crecimiento de los seres vivos, su función principal es optimizar las soluciones de un problema. Este método trabaja modificando su “población”, en este caso son todos los canales que se encuentran disponibles, luego el algoritmo modifica estas soluciones haciendo que se queden las mejores para luego analizarlas y elegir la más óptima [20].

Por otro lado, la revista TECNURA menciona que la ventaja de usar el algoritmo genético en la radio cognitiva permite operar fácilmente las limitaciones y objetivos de una manera parcial [21], es decir que las soluciones ineficientes que encuentre este algoritmo simplemente serán eliminadas. Este método se basa en la búsqueda de armonía para así poder designar un respectivo canal óptimo, esto lo logra gracias a la construcción de un segmento para poder asignar los canales.

Para mejorar la eficiencia del espectro el instituto Nacional de Tecnología, Hamirpur ubicado en la India propuso la investigación del algoritmo Simple additive weighting (SAW) basado en la toma de decisiones multicriterio (MCDM), ya que este método es el más utilizado debido a su baja complejidad. La toma de decisión de este algoritmo lo hace calculando una suma donde están todos los atributos. Para la asignación del espectro el algoritmo dispondrá de varios canales y cada uno de estos canales tendrá su puntaje. Al final la toma de decisión para elegir el canal solo dependerá de la puntuación de este [22].

En la revista internacional de ingeniería electrónica y de comunicación se analizó dos métodos de predicción del espectro el Multiplicative exponent weighting (MEW) y Grey relational análisis (GRA) [23]. De lo cual se obtuvo que la transferencia fallida del MEW fue de 13.88%, el ancho de banda es de 90.47, tuvo un retraso de 48.07 y por último su rendimiento es de 45.83% y en cambio el algoritmo GRA alcanzó una transferencia fallida de 45.45%, su ancho de banda es de 76.19, el retraso obtenido es de 80 y al final este algoritmo logró un rendimiento de 29.16%.

En base a todas las soluciones posibles, este proyecto se diferencia porque utilizará el algoritmo de machine learning long short-term memory (LSTM) para predecir el espectro, este método se lo comparará con uno estadístico para comprobar cuál es más efectivo. Los algoritmos de machine learning toman decisiones con respecto a los datos utilizados, en este caso LSTM es muy interesante ya que se basa en redes neuronales, además a medida que se pongan más datos este método será más efectivo ya que aprende y optimiza los datos para poder mejorar el rendimiento de la predicción.

1.5 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Debido al incremento de usuarios el espectro cada día está más limitado y como se mencionó anteriormente existen varios métodos para solucionar este problema, cada método tiene ventajas y desventajas, por tal motivo se desea implementar una nueva solución, que es la utilización de los algoritmos de machine learning específicamente LSTM los cuales permiten predecir el espectro. Estas redes neuronales como se las conoce se les evaluará su rendimiento comparándolas con un método tradicional en este caso con el método autoregressive integrated moving average (ARIMA) que trabaja con series temporales.

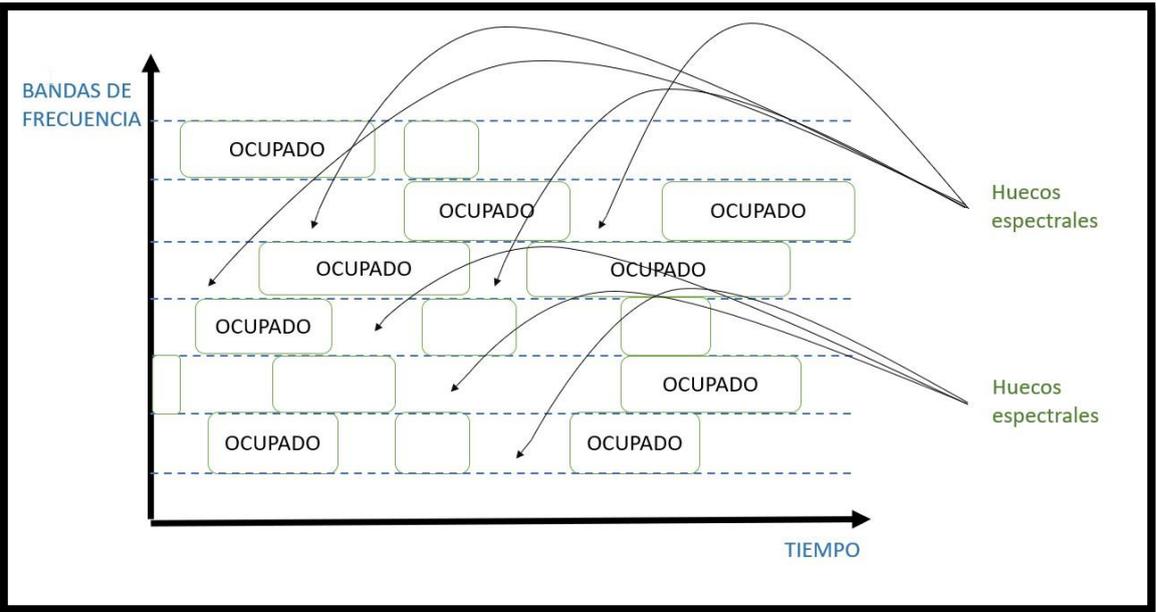


Figura 1.2. Huecos espectrales a lo largo del tiempo

Como se puede observar en la Figura 1.2 existirán huecos en el espectro a los cuales un usuario secundario podrá conectarse, los métodos de ARIMA y LSTM predecirán si los huecos encontrados son adecuados para poder conectarse a ellos, cada método lo hace de una forma específica, pero al final lo que se desea es evaluar que algoritmo es más eficiente.

Para determinar que método es más eficiente se evaluará el tiempo de procesamiento de cada método para encontrar y asignar un canal al usuario secundario, también se estimará en los métodos el tamaño de la muestra, para ver cuál es más preciso con menos datos y por último se obtendrá la probabilidad de predicción que tiene cada método.

CAPITULO 2

2.1 MARCO TEÓRICO

En esta parte del informe se realizan las respectivas investigaciones de los conceptos que serán utilizados en esta tesis, este ayuda a tener bases teóricas sobre los temas que se necesitan conocer para poder desarrollar el respectivo proyecto. Esta parte se enfocará a establecer en que zona de estudio se encuentra el proyecto a desarrollar y permitirá dar un soporte a la propuesta de solución del problema que se va a investigar.

2.2 Espectro radioeléctrico

Conformado por varias ondas electromagnéticas las cuales se encuentran bajo los 3000 GHz, estas tienen la facilidad de propagarse en el espacio de una forma no guiada, es un recurso limitado y todas las personas pueden acceder a este [24]. Es utilizado en la actualidad para diversas tecnologías como para las radiocomunicaciones, lo que se realiza dentro de esta área es permitir el intercambio de información entre el receptor y emisor por medios que no están siendo guiados. El espectro radioeléctrico tiene dos componentes importantes: la propagación que tiene una onda con sus respectivas funcionalidades con distintas frecuencias y los respectivos adelantos científicos de la actualidad.

Para realizar actividades con el espectro radioeléctrico en este momento no se designan con Unión Internacional de Telecomunicaciones (UIT-R) frecuencias que sean menores a 9 kHz, ya que este tipo de espectro no es el adecuado para que las radiotelecomunicaciones trabajen, porque frecuencias menores a 9 kHz cuentan con poca tasa al momento de realizar la transmisión, tampoco UIT-R asigna frecuencias mayores a 275 GHz debido a que la tecnología de la actualidad aún está limitada y por falta de investigación en esta zona del espectro.

Por otro lado, a pesar de que la UIT-R controla este rango de frecuencias para trabajar en el espectro radioeléctrico hay algunas frecuencias que no se encuentran en esta zona regulada como lo son: el infrarrojo y también tenemos la luz visible las cuales son

utilizadas para trabajar en el campo de las telecomunicaciones [25]. En la Figura 2.1 se puede ver las radios frecuencias liberadas que están dentro de la zona regulada.

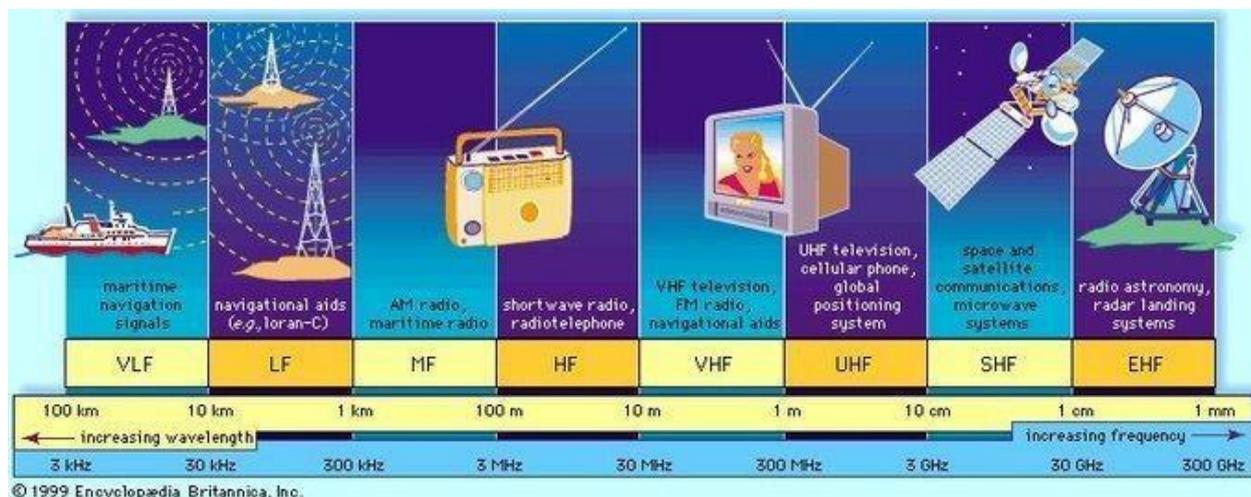


Figura 2.1. Espectro radioeléctrico [50]

2.2.1 Conceptos básicos utilizados en el espectro radioeléctrico.

Establecido en un rango de 3 KHz hasta 3.000 GHz, estas están divididas a su vez en bandas y sub-bandas la cuales se las puede observar en la Tabla 2.1.

BANDAS DE FRECUENCIA				
Número de banda	Símbolos en ingles	Gama de frecuencias	Subdivisión métrica correspondiente	Abreviaturas métricas para las bandas
4	VLF	3 a 30 KHz	Ondas miriámétricas	B. Man
5	LF	30 a 300 KHz	Ondas kilométricas	B.km
6	MF	300 a 3000 KHz	Ondas hectométricas	B.hm
7	HF	3 a 30 MHz	Ondas decamétricas	B.dam
8	VHF	30 a 300 MHz	Ondas métricas	B.m
9	UHF	300 a 3000 MHz	Ondas centimétricas	B.dm
10	SHF	3 a 30 GHz	Ondas milimétricas	B.cm
11	EHF	30 a 300 GHz	Ondas decimilimétricas	B.mm
12	----	300 a 3000 GHz	----	----

Tabla 2.1: Bandas de frecuencia [26]

La tecnología utiliza los diferentes rangos del espectro radioeléctrico. Para transmitir información se requiere de un determinado ancho de banda, el cual se basa en la cantidad de información a transmitirse. Como se puede observar en la Tabla 1 el espectro radioeléctrico está dividido en varias bandas de frecuencia donde cada una de estas tiene características propias como son el ancho de banda [26]. Cuando se desea transmitir información ya sea de audio, video o voz, se necesita una banda de frecuencias específica que se planificó desde el inicio de la división de bandas de frecuencia.

2.2.2 Uso eficiente del espectro

Dependiendo de las necesidades de los usuarios primarios y la disponibilidad del canal, el modelo utilizado para la asignación de espectro procederá a destinar los respectivos canales a los usuarios secundarios, para aprovechar de una forma exitosa el espectro se deben tener en cuenta varios parámetros como lo es su ancho de banda, cuanta interferencia tiene el canal, el error y como está siendo usado este canal específico por el usuario primario. Se puede lograr un uso eficiente del espectro a través de un modelo estadístico el cual nos indique la disponibilidad de los canales que existen en el espectro.

Se puede tomar una decisión en el espectro a través de un método llamado distribuido, el cual consiste en que los usuarios de la red deben analizar la información que se encuentra en la unidad central, lo cual causa un desajuste en el camino de los usuarios [27]. Se debe tener claro que el objetivo es lograr obtener una detección de canales disponibles de una forma eficiente, es decir que aporte a la función de utilidad para que esta trabaje de una forma correcta.

2.3 Parámetros técnicos para la utilización del espectro radioeléctrico

Para poder entender de una mejor manera el espectro radioeléctrico se debe conocer algunos aspectos que son esenciales para analizar las señales, las cuales se las describirá a continuación.

2.3.1 Frecuencia

Cuando se refiere al término de frecuencia, se puede decir que esta mide una cantidad de repeticiones que ocurren en una unidad de tiempo. Según el sistema internacional de unidades la medida de la frecuencia es en hercios (Hz). Al hercio se lo puede interpretar como un suceso que tiene un periodo, el cual se repite una vez por segundo.

Pero el hercio no es el único que mide la frecuencia, también existe otros términos como son los golpes por minuto el cual se lo utiliza en el área de medicina ya que son para medir los latidos del corazón, otra medida son las revoluciones por minuto y también existe los radianes por segundo [28]. En las telecomunicaciones el término frecuencia se utiliza para indicar las bandas donde emite cada emisora.

La frecuencia también se la puede relacionar con su longitud de onda, y es que entre más pequeña sea la longitud de onda más grande va a hacer su frecuencia, por otro lado, entre más grande sea la longitud de onda más pequeña será la frecuencia, esto se debe a que la frecuencia es inversamente proporcional a la longitud de onda [29]. Esto se lo puede observar en la Figura 2.2 que entre más va creciendo la frecuencia más pequeña es la longitud de onda.

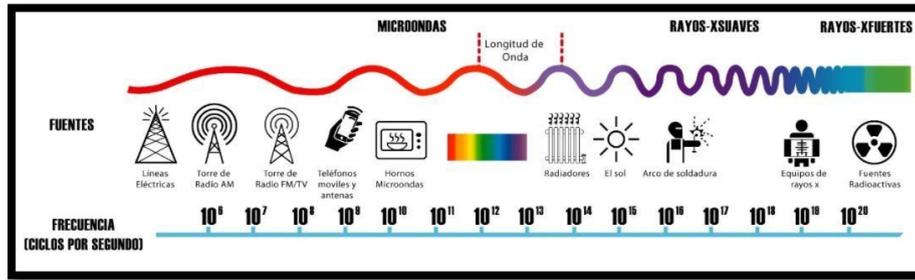


Figura 2.2. Espectro electromagnético [29]

2.3.2 Propagación

La propagación de ondas radioeléctricas puede realizarse de un lugar a otro sin necesidad de algún tipo de cable. Investigaciones confirmaron que dependiendo de la frecuencia estas ondas se transmiten de diferentes maneras. Por tal motivo existen 3 tipos diferentes de propagación de ondas que son las ondas terrestres, las ondas espaciales que a su vez se dividen en reflejadas y directas. Estas se las puede observar en la Figura 2.3.

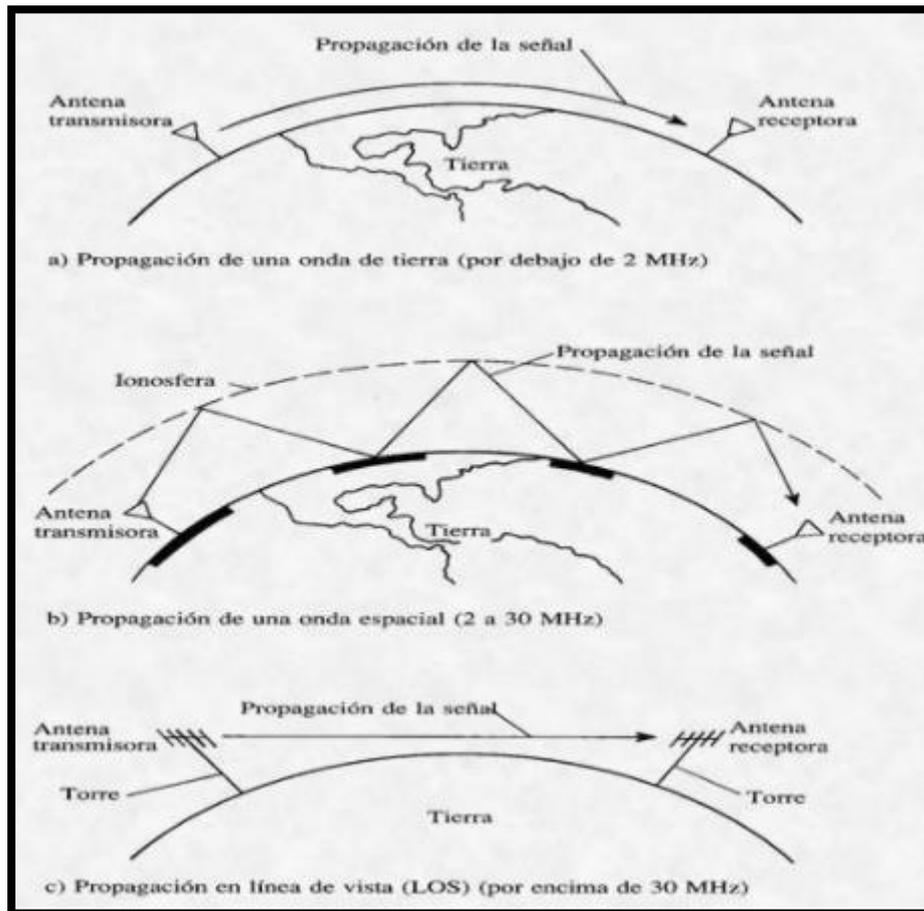


Figura 2.3. a) Ondas terrestres b) Ondas espaciales reflejadas c) Ondas espaciales directas [48]

Cuando se desea transmitir una onda a baja frecuencia existirá una menor atenuación en la onda y por lo tanto esta tendrá una mayor cobertura, en cambio cuando se trabaja a una alta frecuencia la propagación de la onda toma una forma rectilínea, si se desea transmitir de una antena a otra no debe existir ningún obstáculo entre ellos, ósea deben tener línea de vista para poder establecer una comunicación exitosa [30].

2.3.3 Ancho de banda

Se define como la cantidad de información que puede ser transmitida a través de un canal, este término es fundamental para conocer la calidad y velocidad que existe en una conexión [31]. Cada rango del espectro radioeléctrico tiene su propio ancho de banda, en este caso se puede decir que el ancho de banda para el radio es de 220 kHz, en cambio para televisión analógica el ancho de banda es de 6 MHz.

2.3.4 Potencia

La potencia es el indicador de cuanta energía es necesaria para poder transmitir las señales radioeléctricas. Se puede decir que un equipo consumirá menos potencia cuando se trabaje con frecuencias altas, en cambio cuando se utiliza frecuencias bajas la potencia que se utilizará será mayor [26].

2.3.5 Interferencia

Se puede definir a la interferencia radioeléctrica como un efecto de energía no deseada, esto se da por diversas razones como la radiación, emisiones e inducciones. La interferencia puede degradar la calidad de un servicio y esto ocurre por diversos motivos que pueden llegar a ser accidentales o deliberadas. Por tal motivo esto va a tener consecuencias ya sea a largo plazo o a corto plazo. En la figura 2.4 se puede

observar como la interferencia cambia la señal que desea transmitirse, y entre más ruido se tenga más difícil será recuperar la señal original.

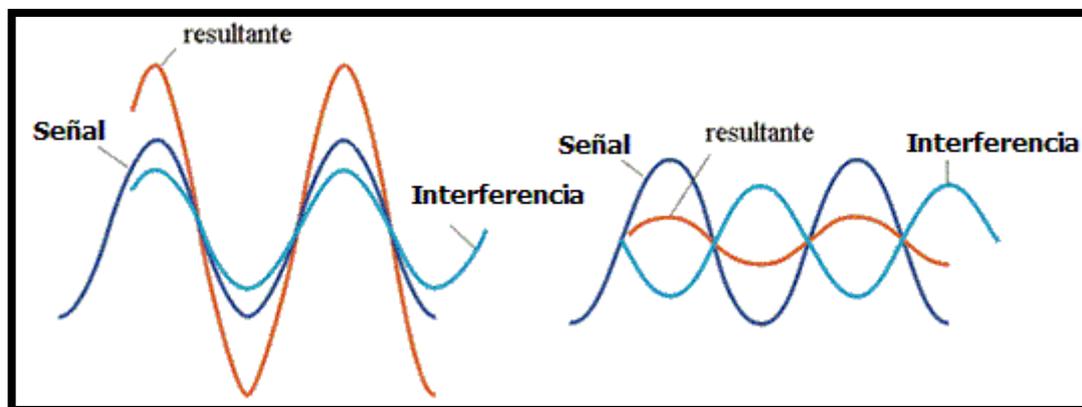


Figura 2.4. Interferencia en una señal [49]

La interferencia también afecta a los sistemas de comunicación de móviles, señales que se utilizan para la radio difusión y también en los sistemas de radionavegación que se los utiliza en buques, aeronaves, vehículos y dispositivos que tengan que ver con la transmisión de señales [32]. Para poder mitigar esta interferencia en la actualidad existen varias soluciones tecnológicas que permiten que el espectro radioeléctrico no se vea tan afectado por la interferencia.

2.4 Radio cognitiva

Se define la radio cognitiva como un sistema diseñado para poder satisfacer los requerimientos de cada uno de los usuarios en el mundo de las telecomunicaciones, a través de analizar cómo se comporta su alrededor para así poder asignar las características necesarias a los usuarios. Su principal función es poder agrupar una respectiva cobertura mundial, la cual tenga la característica de que en cualquier parte del mundo donde se utilice no tenga irregularidades y establecer distintas interfaces a los respectivos sistemas.

Esta cuenta con una arquitectura que se comporta de forma independiente la cual está conformada por diferentes algoritmos que están manejados a través de software, también se lo conoce como un motor el cual está formado por una respectiva interfaz de forma general de radio cognitiva. Tenemos que en la arquitectura dentro de lo que es el

motor este tiene distintos módulos que se comportan de forma funcional, las cuales están constituidas para poder tener habilidades como conciencia, soluciones eficientes a los problemas, lógica y manejo de adaptabilidad del radio.

Como una solución para el sistema de radio cognitivo se debe configurar el algoritmo para que este tenga la capacidad de poder ser cambiado según los requerimientos convenientes para una eficiencia cognitiva [33]. Se tiene un paquete comprendido por el software, el cual tiene un grupo de los respectivos componentes generales correspondientes al aprendizaje y también consta de algoritmos con una interfaz detallada, los cuales tienen la capacidad de poder trabajar con radio implementado por hardware.

Se realiza una implementación en la máquina del núcleo el cual tenga la capacidad de aprender para así poder trabajar con las habilidades cognitivas en el campo de redes inalámbricas. Para que el núcleo pueda funcionar de una manera eficiente este debe tener varios refuerzos de aprendizaje.

2.4.1 Regulación y estandarización en radio cognitiva.

El primer estándar que se implementó en la radio cognitiva fue el IEEE 802.22, se desea que este estándar se lo use para las Wireless Regional Access Network (WRAN), en donde se construyan las redes WRAN punto-multipunto las cuales utilizan bandas UHF y VHF que son de televisión y su frecuencia se encuentra entre los 54 MHz y 862 MHz.

El estándar IEEE 802.22 trata de aprovechar los canales disponibles que hay en la televisión como también aprovecha las respectivas bandas de guarda. El protocolo IEEE 802.22 se estableció en 2004 y su principal diferencia comparado con los demás estándares IEEE 802 es la cobertura de radio que corresponde a las estaciones base [34]. Este estándar aún sigue trabajando en la estandarización, pero por el momento se

especificaron los umbrales para poder conocer si un canal se encuentra desocupado y son los siguientes:

- Para micrófonos inalámbricos se posee un umbral de -107 dBm.
- En la televisión digital se estableció un umbral de -116 dBm.
- En la televisión analógica (NTSC) se tiene un umbral de -94 dBm.

2.4.2 Ahorro de energía en radio cognitiva

Con el tiempo la radio cognitiva se ha hecho popular en todo el mundo, ya que esta busca minimizar la energía que utilizan los equipos en buscar canales que se encuentran disponibles. Las ventajas de la radio cognitiva es que la energía que consume el equipo se reduce, esta es muy importante debido a que se puede lograr una comunicación con menor interferencia, en cambio cuando se posee una mayor interferencia va a haber demasiadas colisiones lo que es un problema al momento de querer asignar un canal a un usuario secundario.

Para comprobar la eficiencia energética de la radio cognitiva se la comparo con una radio convencional, de lo cual obtuvo que la RC puede ahorrar de un 20% a 40% de energía en comparación con la radio convencional. Pero la radio cognitiva también tiene una desventaja y es que al querer detectar huecos espectrales y asignarlos a un usuario secundario conlleva un gran consumo de energía lo que es perjudicial en la batería de cualquier dispositivo (laptops, celulares, tabletas y sensores inalámbricos).

La RC es la encargada de la señalización y búsqueda de los huecos espectrales, por tal motivo es el componente que más energía utiliza. La radio cognitiva consta de 3 operaciones que son la recepción, la transmisión y cuando está en modo inactivo [35]. Cuando la RC está ya sea en el modo de transmisión o recepción consume una potencia muy grande en comparación que cuando esta inactiva, ya que aquí su potencia puede ser despreciable. Se puede decir que cuando la RC está escuchando un canal o esperando algún tipo de información se puede clasificar como modo inactivo.

2.5 Acceso dinámico al espectro

Este método realiza un uso inteligente del espectro, en los últimos tiempos la demanda del espectro ha ido creciendo por ello buscar formas de aprovecharlo de una manera eficiente es primordial. A través del acceso dinámico al espectro se pretende que tanto los usuarios primarios como los secundarios puedan utilizar el mismo espectro, siempre que se logre que los usuarios secundarios no ocasionen interferencia en la red a los usuarios primarios.

Para llevar a cabo el modelo DSA se debe implementar un modelo en el cual los usuarios secundarios sean capaces de detectar cuando el espectro está disponible, es decir, poder utilizar estos huecos espectrales a su beneficio, también debe poder predecir cuándo un usuario principal retorne a su canal para que así este no le cause interferencia. Cuando la radio cognitiva cuenta con un modelo DSA complejo es decir que realiza diversas funciones este es más difícil y caro de realizar [36].

En estos tiempos la radio cognitiva quiere decir siempre que es un método en el cual se puede acceder al ambiente radioeléctrico a través de observar cómo se comportan los usuarios en este y así hacerlo de una manera dinámica. Para saber si el espectro está disponible se realiza un proceso en fuentes fuera de la radio cognitiva y otra forma es que los usuarios secundarios a través de la predicción de espectro saben cuándo el canal está disponible.

2.6 Acceso oportunista al espectro (OSA)

OSA trata de buscar alternativas para reducir la escasez del espectro radioeléctrico, esto ocurre por la administración y también por la asignación de este recurso que es limitado. Por lo tanto, OSA desea que los usuarios secundarios puedan acceder al espectro que le pertenecen a un usuario primario, siempre y cuando este usuario no este ocupando el canal.

Los canales que no están ocupados se los conoce como huecos espectrales a donde un usuario secundario puede conectarse para aprovechar este canal desocupado [37]. Para poder asignar un usuario secundario a un hueco espectral se estableció un valor mínimo de throughput que es -75 dBm, si un valor de potencia es mayor al valor de throughput quiere decir que el canal se encuentra ocupado, pero en otro caso si es menor quiere indicar que el canal se encuentra disponible.

2.7 Handoff espectral

También conocido como movilidad de espectro, este término se refiere a cuando un usuario secundario desea cambiarse de canal de operación, pero para esto se debe tener en cuenta si el usuario primario está ocupando el canal licenciado. El Handoff espectral no tiene los mecanismos tradicionales que se dan en las redes inalámbricas, ya que aquí existen 2 tipos de usuarios. Los usuarios primarios son los que tienen más prioridad en comparación con los usuarios secundarios.

Cuando se desea cambiar de canal siempre ocurrirá que la comunicación se rompa, ya que se está buscando un nuevo canal donde el usuario secundario pueda conectarse, por lo tanto, el modelo de Handoff trata de que un US pueda cambiarse de banda con una mínima degradación de calidad [38]. Cada vez que ocurre este proceso de movilidad de espectro existen varios parámetros que deben satisfacer la calidad de servicios (Qos), para esto se tiene los siguientes requerimientos y causas para realizar un Handoff espectral.

2.7.1 Causas para realizar un Handoff espectral

- Si el canal está siendo utilizado por un usuario primario y un usuario secundario en una misma banda espectral, esto ocurre porque no se verifica el estado de ocupación de una canal y puede que se conecte aun estando un usuario primario ocupando ese canal.

- Cuando llega un usuario primario a la banda que está siendo ocupada por un usuario secundario.
- El SU está usando un canal, pero este se degrada y tenga que cambiarse a otra banda donde la calidad sea mucho mejor.
- Si el usuario secundario interfiere con el usuario primario, aquí se debe realizar un Handoff espectral debido a que el PU tiene la prioridad de estar en el canal.
- Cuando el tráfico en el canal aumenta el usuario secundario se ve en la necesidad de cambiarse de banda donde se tenga unos mejores niveles de desempeño.
- En el momento que un SU se desplaza se debe realizar un Handoff espectral, debido a que se pierde conectividad con el canal.

2.7.2 Requerimientos para realizar un Handoff espectral

- La velocidad es un parámetro muy importante ya que el retardo de la movilidad del espectro debe ser lo más pequeño posible para que no ocurran interrupciones.
- Se debe minimizar la tasa de Handoff espectral, debido a que cuando se realiza varios cambios de canal, esto afecta el desempeño para transmitir los datos.
- La señalización debe ser lo más mínima posible, puesto que, si se tiene un alto volumen de señalización, esto afectará el desempeño en la transmisión de datos.
- Siempre debe haber canales disponibles para que se pueda realizar un Handoff espectral exitoso.

2.7.3 Procedimiento para poder realizar un Handoff espectral exitoso

Para poder realizar un Handoff espectral con el mínimo error, este se basa en 3 fases que son la medición, la decisión y la ejecución.

Medición: Esta fase es una de las más importantes ya que aquí se descubren los huecos espectrales a donde un usuario secundario podría conectarse.

Decisión: Aquí se elige cuando y en que canal se realizará la movilidad de espectro, pero para tomar esta decisión se basa en los parámetros que ya se mencionó anteriormente.

Ejecución: Esta es la última fase y aquí se realiza el cambio de un canal a otro.

2.8 Importancia del análisis predictivo

Con el pasar de los años el análisis predictivo se ha mucho muy popular en todas las áreas. Para esto existen muchas herramientas capaces de realizar una predicción para los eventos futuros, regularmente existen muchos métodos para poder predecir datos. En la actualidad estos sistemas se utilizan prácticamente casi en todo ya que sugieren tomar acciones lo que permite mejorar los resultados. La predicción en las áreas que más se la utiliza son en big data y también en el aprendizaje automático.

2.8.1 El desarrollo de la big data

La big data juega un papel muy importante en la predicción de los datos, esto se debe a que con esta información se realiza la predicción. Por tal motivo las mediciones que se realizan con sensores o algún otro instrumento deben estar bien realizadas ya que con esta valiosa información las empresas toman decisiones.

2.8.2 Aumento de la competencia

Entre más popular se vuelva la predicción en todas las áreas, más competencia va a ver, esto es muy importante ya que día a día los modelos predictivos van a mejorar lo que implica que van a ayudar a las empresas en sus problemas. Un ejemplo para poder entender de mejor manera el aumento de competencia es que algunas empresas utilizan la predicción para la energía eléctrica, pero puede venir otra persona presentado una predicción más eficiente para ahorrar la energía.

2.8.3 Tecnologías innovadoras para el uso del big data

Cuando se menciona el big data es la parte más importante de la predicción ya que si no se tuviera esta base de datos no sería posible predecir. Para poder trabajar con una base de datos grandes las empresas utilizan programas como Spark y Hadoop, aunque cada vez se innovan nuevos programas que hace más fácil trabajar con los datos. Para poder predecir los algoritmos utilizan aprendizaje automático que se basan en construir modelos que permitan encontrar patrones o tendencias de los datos y así predecir un resultado futuro.

2.8.4 Ejemplos de áreas donde se utiliza el análisis predictivo

- Aeroespacial
- Industria automotriz
- Servicios financieros
- Ingeniería mecánica
- Producción y ahorro de energía
- Medicina

2.8.5 Análisis predictivo con MATLAB

Entre más evolucionan las técnicas de predicción, más personas utilizan Matlab ya que esta herramienta permite aprovechar la base de datos técnicos y comerciales para poder predecir decisiones en base a esta información. Matlab ofrece muchas ventajas al momento de predecir puesto que realiza muchos análisis predictivos en distintas áreas como son en datos científicos, técnicos y comerciales, aparte puede trabajar con una base de datos muy grande [39]. Los beneficios de trabajar con Matlab son los siguientes:

- En Matlab se puede trabajar con datos técnicos y comerciales, esto es debido a que este programa tiene un soporte nativo para tomar datos en tiempo real, estos datos pueden ser tomados de imágenes, sensores, binarios, telemetría y video, aparte permite trabajar con herramientas muy útiles como Spark y Hadoop.

- Gracias a Matlab las personas pueden configurar la ciencia de datos de acuerdo con su conveniencia, esto se da ya que Matlab posee herramientas muy útiles como procesamiento de imágenes, aprendizaje automático de los datos, funciones estadísticas y análisis de señales.
- Se puede analizar los sistemas de tecnología de información (TI) gracias a Matlab, esto es debido a que se puede integrar en la nube, clústeres y sistemas corporativos.

2.9 Python para predicción de datos

Es el campo más actual de las telecomunicaciones, es muy interesante ya que existen infinidad de datos en el mundo los cuales se busca convertirlos en conocimiento a través de la observación de los datos para aprender cómo funcionan estos. Debido a las variadas librerías que se tiene hoy en la internet es mucho más fácil para las personas implementar algoritmos de aprendizaje a través de la plataforma Python, logrando así poder predecir eventos futuros.

Se utiliza el programa Python para realizar la predicción de datos, lo cual en los últimos tiempos ha sido de gran interés para desarrollar algoritmos de predicción, a estos métodos se le denomina aprendizaje automático lo cual pertenece a la rama de investigación de la inteligencia artificial [40]. Para evitar largas horas de trabajo por el ser humano se crea un algoritmo que sea capaz de aprender de los datos para así crear un modelo eficiente para un objetivo específico, un ejemplo muy conocido sobre la predicción de datos a través del programa Python es el juego de ajedrez el cual analiza los datos y aprende a jugar este juego.

2.10 Huecos espectrales

Estos son aquellos espacios que existen en la red radioeléctrica, es decir si un usuario primario no está usando el canal en ese momento se crean huecos espectrales que pueden ser usados por los usuarios secundarios. En la actualidad se busca

aprovechar estos huecos espectrales para cubrir la demanda que existen en el internet inalámbrico, desarrollando así métodos para utilizar de forma eficiente el espectro, los cuales permiten utilizar la frecuencia requerida si esta está disponible. Se puede observar en la Figura 2.5 cuales son los huecos espectrales

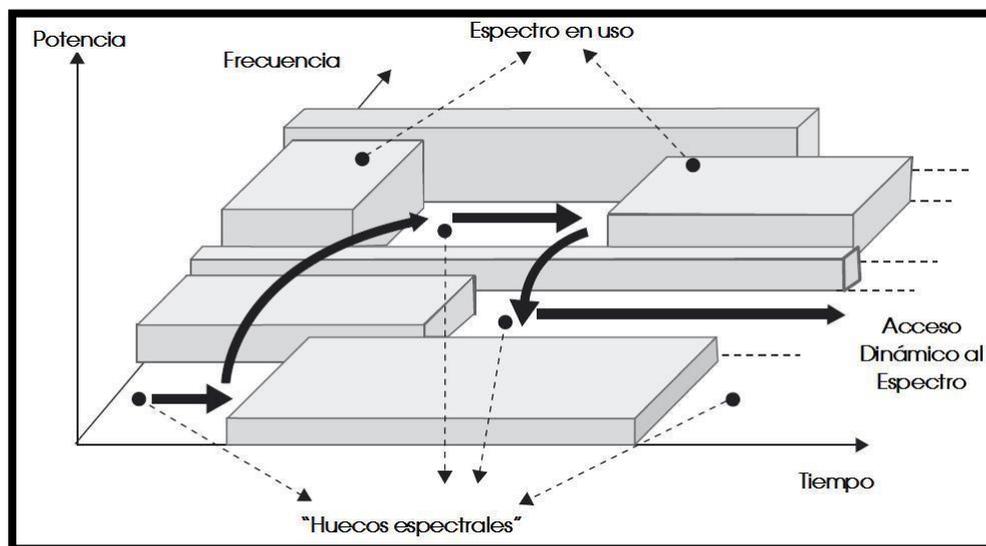


Figura 2.5 Huecos espectrales [51]

Este método desarrollado se basa en la observación de datos, analizando los espacios en blanco y de acuerdo con las regulaciones que se tenga en cada zona específica se podrá establecer un método eficiente. Estos buscan siempre comprender bien las características de este canal para así poder evaluar y evitar que los SU interfieran con los PU, por ello aprovechar estos huecos espectrales es conveniente para que la tecnología de las redes inalámbricas siga creciendo.

Existen diversas investigaciones sobre estos huecos espectrales las cuales mencionan que asignar un espacio en blanco a un usuario tiene varias dificultades como lo son que no se garantiza que el canal esté siempre disponible y que el canal cambiará constantemente su estado por un usuario principal, por lo que se pueden ocasionar interferencias [41]. Por ello se propone la predicción de espectro para así poder determinar con anticipación si un canal estará disponible y en qué momento un usuario principal ocupará este canal.

2.11 Método estadístico ARIMA

Este modelo es uno de los más conocidos para realizar predicciones debido a su eficiente trabajo en las investigaciones realizadas. Este modelo tiene varias fases para su procedimiento como lo son: verificar con que modelo se va a trabajar, analizar los parámetros del modelo, realizar el proceso a las características que tiene el modelo y por último se realiza el pronóstico. A partir de este proceso se requiere conocer el proceso estocástico que generan los datos.

Para poner en práctica el modelo ARIMA se requiere determinar ciertos parámetros necesarios para que este modelo sea el más óptimo, la serie correcta para poder realizar los pronósticos debe tener una media y una varianza estacionaria [42]. Debido a que este modelo depende de tres variables que son (p, d, q) donde p indica el número de términos autorregresivos que tiene el sistema, d representa las veces que los datos necesitan ser diferenciados para volverlos estacionarios y por último la variable q indica las tendencias en la media móvil de carácter invertible.

El modelo ARIMA ha sido desarrollado desde el año 70, la cual se basa en retomar datos del pasado a través de la observación de la serie de tiempo con respecto a su autocorrelación, por ello este modelo cuenta con la variable p en su ecuación principal la cual hace referencia a las variables autorregresivas, donde también se analizan los errores del pasado con ayuda de la media móvil. Es considerado un método eficiente para predicciones futuras.

2.12 Método Machine learning LSTM

Este método tiene la capacidad de trabajar con redes neuronales, estas trabajan basándose en cómo funcionan las neuronas del cerebro humano. Este método LSTM permite realizar predicciones futuras, el cual tiene la ventaja de ser estable y con mayor efectividad comparada con un algún otro método de predicción, debido a que este

método cuenta con una memoria larga puede trabajar de una mejor manera las series de tiempo.

Es importante recalcar que se sugiere utilizar redes neuronales cuando los métodos tradicionales utilizados para la predicción no son capaces de resolver el problema, la idea de utilizar este método es contribuir a través de la observación de las series de tiempo poder realizar predicciones futuras, utilizando así la información existente en la red sobre el tema que se quiera realizar el trabajo y a través de ello poder realizar una predicción futura.

Este modelo es utilizado para trabajos relacionados con detección de voz e imagen, gracias a que tiene la característica de selectividad, es decir cuenta con células de memoria que permiten identificar una imagen o voz. Este método da resultados muy buenos, debido a sus altos porcentajes de precisión, también a que trabaja con una aproximación no lineal y utiliza un autoaprendizaje de una forma adaptiva [43].

2.13 Series de tiempos

Se llama serie de tiempo cuando se tiene varios datos que son observados en un tiempo determinado, los cuales se encuentran de forma ordenada según los datos y se los caracteriza de forma uniforme, esto quiere decir que los datos que se tomen para analizar van a depender todos de sí, desde el primer dato hasta el último. Las series de tiempo casi siempre son analizadas para poder realizar predicciones futuras aprovechando los datos obtenidos de un área específica.

Estas se desarrollan de una forma en que predice un dato de acuerdo con los datos observados, esto lo realiza a través de las componentes de tendencia, estacionalidad y aleatoriedad que se pueden observar en la Figura 2.6. Las dos variables primeras se utilizan para determinar datos específicos de las series de tiempo y la última variable definida se utiliza para determinar datos aleatoriamente de acuerdo con las observaciones realizadas. Se puede decir que son una serie de datos las cuales van evolucionando con respecto al tiempo.

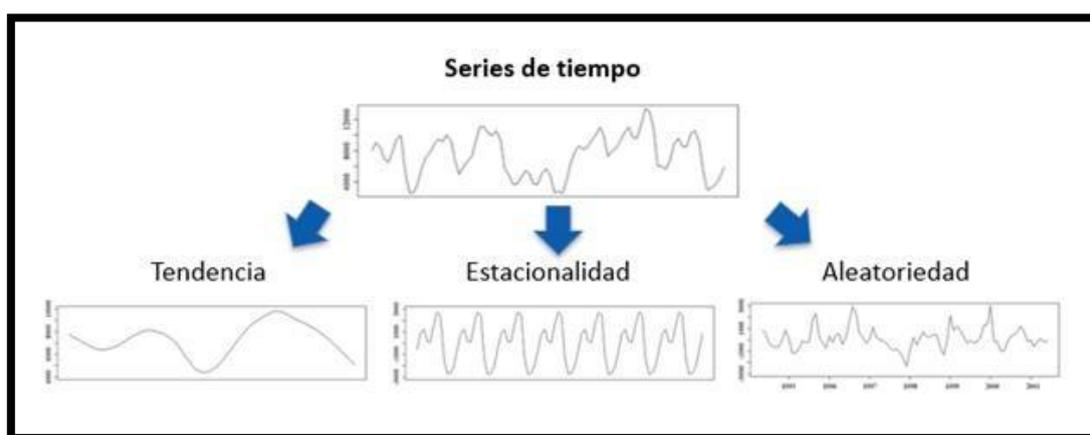


Figura 2.6 Componentes de las series de tiempo [52]

Se pueden organizar en series estacionarias y no estacionarias. Donde las estacionarias nos dicen que la serie se encuentra estable durante todo el tiempo, esto quiere decir que la varianza y la media de la serie analizada van a hacer constantes a lo largo de todo el tiempo, para definir si una serie de tiempo es estacionaria se debe analizar que la varianza sea constante con referencia a la media que se tiene respectivamente de la serie analizada [44].

Mientras que las series no estacionarias se definen porque la tendencia y la variabilidad de los datos a través del tiempo van cambiando, es decir siempre están de forma variable por ejemplo tienden a decrecer y a crecer durante toda la serie, esto implica que la serie no tiende a oscilar en un ambiente constante. Antes de trabajar con una serie de tiempo se debe determinar qué clase de serie es esta y luego de esto determinar la media, varianza y covarianza.

2.14 Redes Neuronales

Esta área de estudio pertenece a la inteligencia artificial, es una de las ramas más importantes de esta debido a su enorme beneficio. Trabajan teniendo neuronas de conocimiento es decir analizan la naturaleza de los datos de acuerdo con las características de estos a través de la observación de su comportamiento y almacena esta información para predecir como se comportarán en un tiempo futuro. Este método está basado en cómo se comportan las neuronas del ser humano.

Existen varias diferencias entre cómo trabajan las neuronas humanas y las neuronas de las redes neuronales una de ellas es la arquitectura que presentan las computadoras siguen el modelo Von Neuman, el cual ha sido implementado para trabajar con series complejas que varían rápidamente en el tiempo haciéndolo de una forma eficiente y segura [45]. Las neuronas del cerebro humano no necesitan ser programadas estas funcionan de forma natural analizando los estímulos que reciben las personas de lo que tienen a su alrededor.

Las redes neuronales artificiales están arquitectónicamente diseñadas en paralelo de los respectivos elementos simples y están organizadas de una forma jerárquica, estas trabajan interpretando los datos del mundo y se encargan de procesar la distinta información de la red a través de un estado dinámico, debido a la variabilidad que existe en los datos. Son capaces de realizar tareas a través de su mecanismo como por ejemplo aprender de patrones definidos y aprender a clasificar según lo requerido.

CAPITULO 3

3.1 Diseño del Proyecto

En este capítulo se presenta la solución desarrollada para el proyecto de tesis. Se encontrará con diagramas de bloques que explican los pasos a realizar para resolver el problema plantado anteriormente, primero se tiene un diagrama de bloques básico el cual explica de forma general como se realizara la solución al proyecto, luego se presenta un diagrama de flujo de cómo funciona el algoritmo realizado para predecir la disponibilidad del canal en el espectro, después se tiene el diagrama de bloques detallado el cual presenta de una forma más específica los pasos a realizar para la solución del proyecto. Por último, se presenta los dos métodos realizados en lenguaje de programación con su respectiva explicación de cada paso del algoritmo planteado para resolver el problema del proyecto.

3.1.1 Diagrama de bloques básico

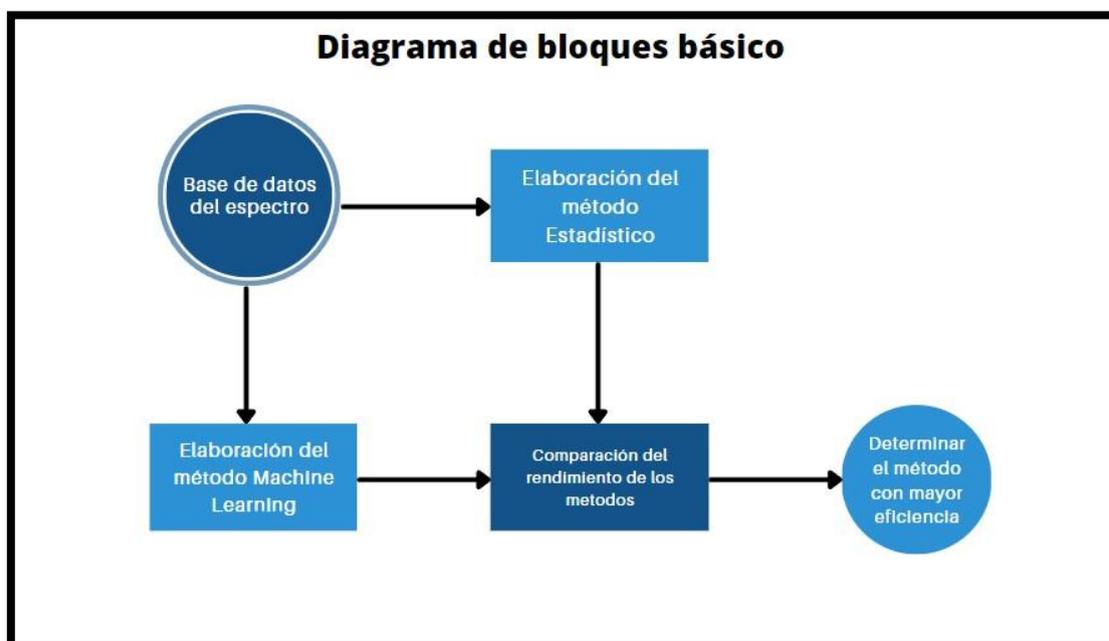


Figura 3.1: Diagrama de bloques

Para realizar este proyecto se basó en la Figura 3.1, primero se necesitará de una recolección de datos para poder determinar la disponibilidad del canal en el espectro, a través de dos métodos uno estadístico específicamente ARIMA y el otro método será con Machine Learning LSTM. Se comparará estos dos métodos con la base de datos obtenida y se comprobará cual de estos dos realiza una mejor predicción de canal disponible en el espectro.

Analizar que método tiene mayor eficiencia es el propósito en el proyecto por lo cual para obtener resultados de la predicción de la disponibilidad de los canales en el espectro se lo realizó a través de un lenguaje de programación en el cual se asignó los valores de 1 y 0 para determinar cuando el canal se encuentra disponible y ocupado.

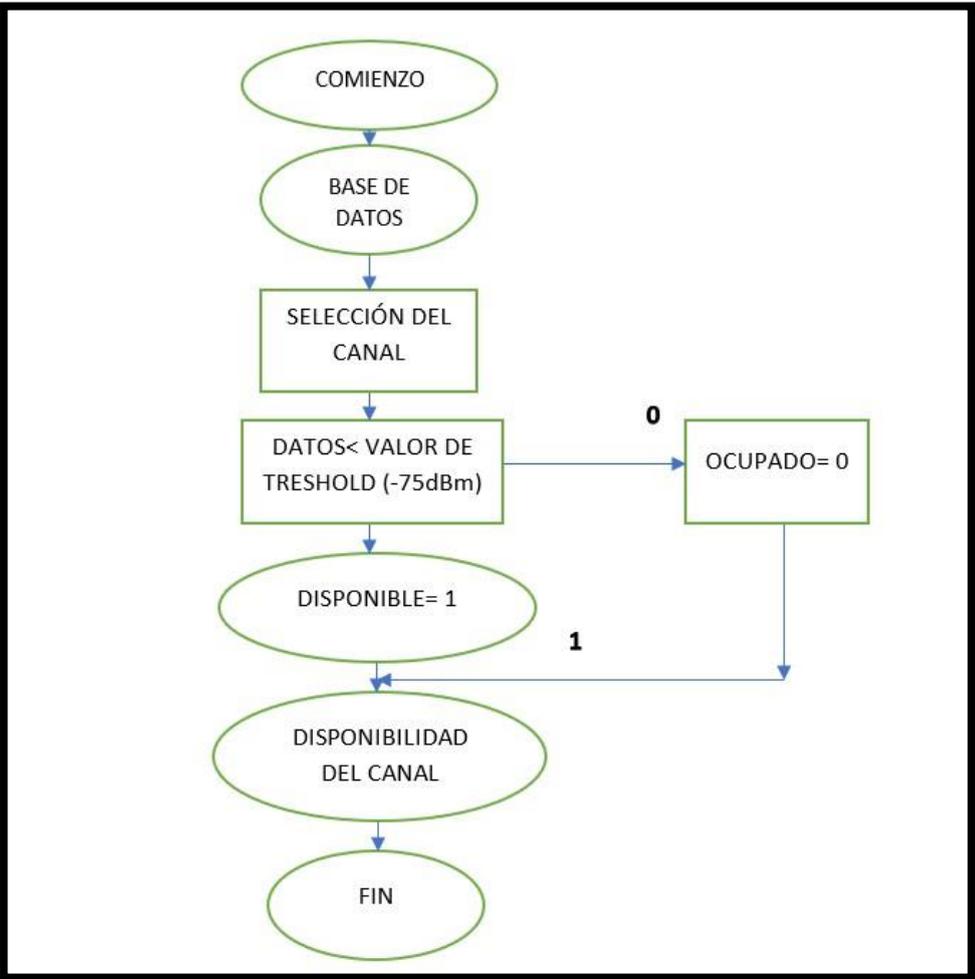


Figura 3.2: Diagrama de flujo para la detección de canales disponibles

Cuando tenemos 1 quiere decir que el canal está disponible y cuando se tiene 0 quiere decir que el canal se encuentra ocupado, en el diagrama de flujo de la Figura 3.2 se puede ver cómo funciona el algoritmo para la detección de los canales disponibles en el espectro.

Los algoritmos de predicción se realizaron en diferentes programas, el método de ARIMA en RStudio y el método de machine learning en Python debido a que estos programas contienen varias funciones útiles para el desarrollo de los algoritmos ya que permiten utilizar los datos obtenidos que se encuentran en Excel, exportándolos a través de los mismos programas.

Se trabajó con el programa RStudio para desarrollar el método estadístico ARIMA, debido a que este facilita la visualización de los respectivos datos para poder realizar de una forma mejor la predicción de los canales disponibles en el espectro. El programa RStudio está diseñado en base a la estadística, por ello al ser ARIMA un método estadístico es más sencillo al momento de realizar el análisis. Para el método de LSTM se trabajó con Python debido a que este es fácil de utilizar, ya que cuenta con una sintaxis más aproximada al inglés, lo cual hace más fácil al momento de ejecutar los correspondientes algoritmos.

Cuando se obtengan todos los datos requeridos de los algoritmos para realizar la respectiva predicción de los canales disponibles en el espectro, se procederá a comparar estos dos métodos para entender cuál de ellos está realizando la predicción de una forma más eficiente. Los valores para comparar no siempre serán los mismo debido a que estos cambiarán cada vez que se realice la predicción de espectro.

3.1.2 Diagrama de bloques detallado

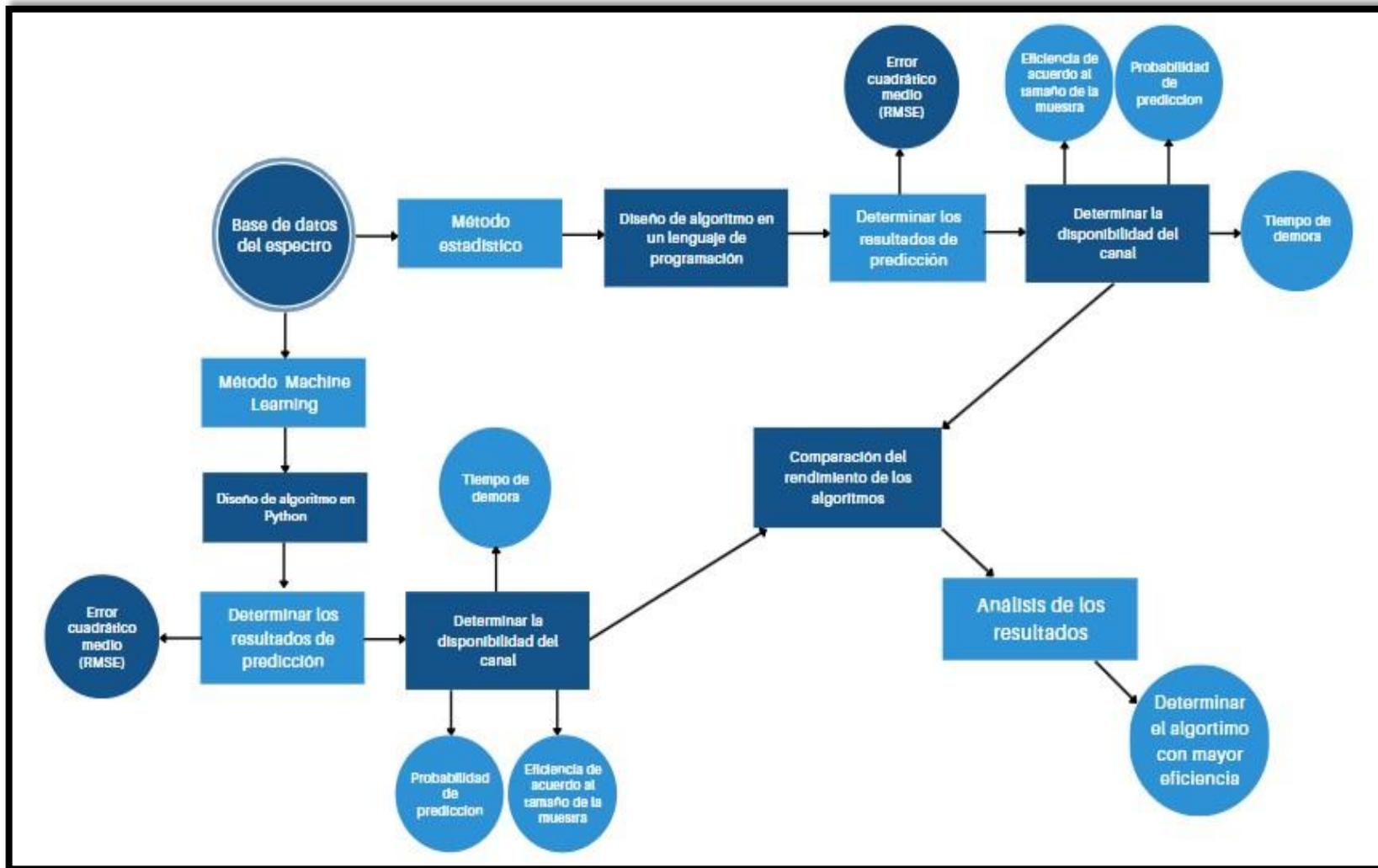


Figura 3.3: Diagrama de bloques completo

Para el diagrama mostrado en la Figura 3.3, primeramente, se dispondrá de una base de datos del espectro el cual contiene información de 3 canales, luego gracias a esta información se podrá predecir la disponibilidad del canal usando dos algoritmos, uno es el método estadístico y otro es el método de machine learning.

3.2 Método estadístico (ARIMA)

En este caso se trabajó con el método de ARIMA para poder predecir la disponibilidad del canal, se usó el lenguaje de programación de RStudio el cual permitió crear un algoritmo en base a las características que se desea. El modelo de ARIMA tiene 3 parámetros importantes para poder trabajar adecuadamente, estos parámetros son p, d y q. La variable p simboliza el orden de auto regresión del modelo, q indica el orden de media móvil que significa cuantas observaciones pasadas va a tomar el método y d es el número de diferenciaciones que se debe hacer a los datos para que este sea estacionario.

Al disponer de 3 canales se tendrá 3 modelos ARIMA diferentes ya que van a variar los parámetros de p, d y q, esto se debe a que los 3 canales van a tener diferentes datos. Cuando ya se haya implementado estos modelos lo siguiente que se hará es predecir el espectro y que error tiene esta predicción, para esto se calcula el error cuadrático medio (RMSE) que indicará que tan bueno es el modelo. Lo que se busca en esta parte es que el RMSE sea lo más pequeño posible ya que esto significará que la predicción del espectro va a tener muy poco error.

Una vez ya realizada la predicción del espectro se procede a calcular la disponibilidad, para esto si los valores son menores a -75 dBm quiere indicar que el canal está disponible y se lo representa con un 1, pero si los valores son mayores a -75 dBm quiere decir que el canal se encuentra ocupado y por lo tanto se lo representa con un 0. En la disponibilidad del canal se obtuvo 3 resultados, el primero es la eficiencia de acuerdo con el tamaño de la muestra, el segundo es la probabilidad de predicción de la disponibilidad y por último el tiempo de demora del algoritmo.

3.2.1 Creación del algoritmo ARIMA

Para poder crear el algoritmo para la predicción de disponibilidad del canal, primero se procedió a trabajar con RStudio ya que este programa es muy bueno cuando se trata de métodos estadísticos. Cuenta con librerías muy buenas para la predicción de datos. En este método se usaron las librerías que se las puede observar en la Figura 3.4, donde cada de una de estas es importante para poder realizar la predicción correctamente.

```
#Importar librerías  
library(forecast)  
library(tseries)  
library(astsa)  
library(readr)
```

Figura 3.4: Librerías a utilizar

La librería forecast contiene varias funciones muy útiles para la predicción de datos, en este caso aquí se encuentra la función ARIMA que será de utilidad para poder predecir la disponibilidad del canal. La librería tseries permite manipular los datos relacionados a las series temporales, como aquí ARIMA trabaja con series temporales es muy importante esta librería ya que se puede jugar con los datos a conveniencia de lo que se desee realizar.

La librería astsa es un paquete útil para trabajar con series de tiempo y también para poder realizar predicciones, además contiene herramientas estadísticas básicas que son necesarias para poder obtener los resultados de predicción del algoritmo. El paquete readr permite exportar los datos del computador al programa, en este caso se tienen los datos en formato .csv donde esta librería facilitará el trabajo y el manejo de los datos.

Una vez importada las librerías se procede a llamar los datos que se encuentra en la computadora, para esto se utilizó el código que se puede observar en la Figura 3.5. Aquí se utiliza la función read_delim la cual permite exportar los datos al programa, además estos datos esta separados por punto y coma, y en este código se eliminó esa

variable ya que se desea que los datos no posean caracteres de este tipo porque luego no se los puede trabajar.

```
#extraer los datos .csv
datos<- read_delim("DATA.csv",delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

Figura 3.5: Extracción de datos

Luego para comprobar que se extrajo de manera exitosa los datos, se procede a graficarlos, este código se encuentra en la Figura 3.6, pero no se puede graficarlos directamente ya que este archivo contiene información de 3 canales diferentes, por tal motivo se debe llamar al canal específico con el que se desea trabajar. Cuando ya se realizó esto se procede a crear una señal en el tiempo que comienza en 1 y termina en la longitud del canal utilizado, luego gracias a la función plot se puede graficar todos los datos a utilizar.

```
#Graficar los datos del canal a usar
CH1<-datos$CH3
CH1
x1=ts(CH1,start=1,end=length(CH1),frequency=1)
plot(x1, main="Datos del canal",xlab="tiempo", ylab="potencia")
```

Figura 3.6: Código del gráfico del canal

Una de las partes más importantes para poder crear el código ARIMA es que sus datos sean estacionarios, ya que si no lo son los resultados serán erróneos. En la figura 3.7 se realizó una prueba de estacionalidad para los datos, en la cual se utilizó la función adf.test donde como resultado indica si el canal es estacionario o no. Si el canal no fuera estacionario se debería hacer diferenciaciones para que los datos se vuelvan estacionarios y poder trabajar con ellos.

```
#Prueba de estacionalidad
adf.test(x1, alternative = "stationary")
#H0: No estacionaria
#H1: Estacionaria
#Con un p-value menor a 0.05, se acepta que la serie es estacionaria
```

Figura 3.7: Código para prueba de estacionalidad

Debido a que ARIMA predice los datos de acuerdo con sus datos anteriores y de acuerdo con la tendencia de los datos, resulta difícil predecir el espectro esto se debe a que el espectro no tiene una tendencia fija y sus datos en el tiempo cambian demasiado. Por tal motivo se dividió los datos en 250 cortes para que así el modelo de ARIMA sea más eficiente al momento de predecir los datos futuros, el código para la separación de los datos está indicado en la Figura 3.8.

```
#partir la data en 250 parte iguales
n<-length(CH1) #tamaño de las observaciones
k<-250 #numero de cortes
corte<-seq(n/k, n, length.out = k)
corte<-c(0 ,corte)
abline(v=corte, col="blue",lwd=0.5)
```

Figura 3.8: Código para partir los datos en partes iguales

El método de ARIMA se basa en 3 parámetros para poder predecir los datos, estos parámetros son p, d y q como se muestra en la Figura 3.9. Las variables p y q pueden ser cambiadas a conveniencia, pero la d debe ser fija. Antes se analizó el código de prueba de estacionalidad donde resultó que los datos eran estacionarios, cuando los datos ya son estacionarios el parámetro d debe ser cero y si no es estacionario el parámetro d debe ser aumentado hasta que se los datos se vuelvan estacionarios.

```
#Crear arima
p<-4 #el número de términos autoregresivos
d<-0 #el número de diferenciaciones para hacerla estacionaria
q<-5 #el número de términos de la media móvil invertible
```

Figura 3.9: Creación del modelo ARIMA

Una vez creado el modelo se procede a hacer la predicción de los datos, como se menciono anteriormente se separaron los datos en 250 partes por tal motivo se debe realizar la predicción 250 veces para esto se creó un for que esta descrito en la Figura 3.10 y 3.11, este for va desde el corte 1 hasta el corte 250, para cada corte se realiza

una predcción donde se toma el 70% de datos para entrenar al modelo y 30% para la predcción.

Ya que se haya realizado la predcción de espectro se procede a convertir los valores en 0 y 1 dependiendo de su valor, si son menores a -75 dBm representan un 1 y por otro lado si son mayores a -75 dBm representan un 0. Los valores 1 y 0 indican si el canal esta disponible o ocupado respectivamente. Una vez realizado esto se procede a comparar estos valores de predcción de disponibilidad con los valores reales, de lo cual se obtiene el porcentaje de que tan eficiente esta predciendo el modelo ARIMA la disponibilidad.

Otra cosa muy importante que se obtuvo es el RMSE el cual indica cuanto error hay entre los datos predcidos con los datos reales. También se realizó el porcentaje de datos ocupados y disponibles que hay en el canal.

```
#muestras independientes
for (j in 1:k) {
  #hacer arima con la primera muestra
  ii<-j
  muestra<-CH1[corte[ii]:corte[ii+1]]

  #Prediccion y comparar del 70% y 30%, de los 50824 datos
  N<-length(muestra)
  i<-0.70*N
  CH1.train<-muestra[1:i] #Datos para entrenar
  CH1.test<-muestra[(i+1):N] #Datos para testear

  #crea modelo arima
  modelo=arima(CH1.train,order=c(p,d,q),method="ML")

  #prediccion
  s<-N-i #cantidad de prediccion de test
  predi<-forecast(modelo,s)

  #Diagnostico de prediccion
  Diagnostico4<-rep(0,s)
  Diagnostico4[predi$mean <= -75] <- 1
}
```

Figura 3.10: Predicción de datos

```

#Diagnostico reales
Diagnostico<-rep(0,s)
Diagnostico[CH1.test <= -75] <- 1

Diagnostico5<-rep(0,s)
for(m in 1:length(Diagnostico)){
  if(Diagnostico[m]==Diagnostico4[m]){Diagnostico5[m]<-1} else{Diagnostico5[m]<-0}
}
Porcentaje[j]<-sum(Diagnostico5)/s

#calcular residuos entre observado y esperado
residuos<-predi$mean - CH1.test
RMSE[j]<-sum(residuos^2)/s

#calcular diagnostico señal ocupado
Porcentajesenalocupado[j]<- sum( predi$mean >= -75 )/s
#calcular diagnostico disponible
PorcentajesenalDisponible[j]<- sum( predi$mean < -75 )/s

```

Figura 3.11: Predicción de datos

Por último, se obtienen los resultados de la predicción, al a ver 250 cortes, se tendrá 250 valores, por tal motivo se debe sacar el promedio de estos datos para conocer exactamente el porcentaje. El promedio que se observa en la Figura 3.12 representa que tan eficiente es el modelo ARIMA para predecir todos los canales disponibles, asimismo se puede observar que se sacó el promedio de todos de los datos disponibles y ocupados, además del promedio del RMSE.

```

promedio=mean(Porcentaje)
mean(Porcentajesenalocupado)
mean(PorcentajesenalDisponible)|
mean(RMSE)

```

Figura 3.12: Código de los resultados

3.3 Método machine learning (LSTM)

Para la elaboración del algoritmo LSTM que predice la disponibilidad del canal se usó el lenguaje de programación de Python. LSTM es un método que se le da muy bien aprender toda clase de secuencias ya que trabaja con redes recurrentes las que permiten recordar la información pasada de todos los datos. En este caso a comparación de ARIMA solo se necesita un modelo para poder predecir la disponibilidad de los 3 canales,

esto se debe a que LSTM no trabaja con parámetros si no la función principal de este modelo es aprender de forma iterativa todos los datos del canal y así poder predecir de una forma más adecuada.

Ya realizado el algoritmo lo primero que se realiza es calcular la predicción del espectro y ver que tan eficiente es esta predicción, para esto igual que ARIMA se obtuvo el RMSE el cual permite calcular el error que hay en la predicción. Una vez ya realizada la predicción del espectro se procede a calcular la disponibilidad del canal de la misma forma que se lo hizo para el modelo de ARIMA. Aquí también se obtuvo 3 resultados, la eficiencia de acuerdo con el tamaño de la muestra, la probabilidad de predicción de la disponibilidad y por último el tiempo de demora del algoritmo.

El principal objetivo de haber realizado estos algoritmos es determinar cuál es más eficiente, para esto se procedió a comparar los parámetros que se obtuvieron en la disponibilidad que eran la eficiencia de acuerdo con el tamaño de la muestra, la probabilidad de predicción de la disponibilidad y el tiempo de demora del algoritmo. Lo que se hizo luego es analizar todos estos parámetros para ARIMA y LSTM y así poder determinar el algoritmo con mayor eficiencia.

3.3.1 Creación del algoritmo LSTM

Para la creación del método de LSTM se trabajó con el programa Python, el cual su función principal es predecir la disponibilidad del canal. Se usó la herramienta Python debido a que esta cuenta con muchas librerías importantes, para la predicción de la disponibilidad de los canales en el espectro. En la Figura 3.13 se puede observar las librerías principales que se utilizarán para la creación del algoritmo LSTM.

```

▶ import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
pd.set_option('display.float_format', lambda x: '%.4f' % x)
import seaborn as sns
sns.set_context("paper", font_scale=1.3)
sns.set_style('white')
import warnings
warnings.filterwarnings('ignore')
from time import time
import matplotlib.ticker as tkr
from numpy.random import randn
from numpy.random import seed
from scipy import stats
from scipy.stats import pearsonr
from statsmodels.tsa.stattools import adfuller
from sklearn import preprocessing
from statsmodels.tsa.stattools import pacf
%matplotlib inline
import math
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import *
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import accuracy_score
from keras.callbacks import EarlyStopping
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import signal

```

Figura 3.13: Librerías a utilizar

A continuación, se describirá lo que realiza cada librería:

- La librería numpy es muy importante ya que permite trabajar con arreglos, lo cual es esencial para poder ordenar los datos correctamente y poder predecir de una forma más eficiente.
- Para poder entender de una mejor manera los datos se utilizó la librería matplotlib.pyplot, la cual permite graficar en dos dimensiones.
- Al momento de tener la base de datos, es necesario tener una opción que permita exportar datos desde el computador hacia el programa, esto se logra gracias a la librería pandas.
- La librería seaborn trabaja casi igual que matplotlib.pyplot, pero la diferencia es que esta proporciona una interfaz más avanzada, para que se puedan apreciar de una mejor manera los datos.

- Cada vez que se ejecuta una línea de código puede aparecer una advertencia, para que no aparezcan estos avisos se utiliza la librería warnings.
- Para que los datos se los ordene de una mejor manera es necesario que se guíen por un tiempo, esto se logra a través de la librería time.
- La librería scipy contiene muchas funciones importantes para el manejo de los datos, entre las principales esta la FFT, la resolución de EDOs, la interpolación y la integración.
- Statsmodels es un paquete que permite realizar modelos estadísticos.
- Con la librería sklearn se puede predecir datos, este paquete está basado en las librerías SciPy, NumPy y matplotlib.
- Esta librería math permite realizar operaciones matemáticas.
- Keras es una librería que permite la predicción a base de redes neuronales, las cuales se basan en la inteligencia artificial.

Cuando ya se exportan todas las librerías a utilizar, lo siguiente que se debe hacer es extraer los datos que están en el computador para esto se utiliza la librería de pandas. Aquí se usó la función pd.read para poder obtener los datos del archivo con formato .csv, en la Figura 3.14 se puede observar el código donde se está exportando los datos.

```
#Lectura de datos de potencia de senial
pd.set_option('display.float_format', lambda x: '%.4f' % x)
df=pd.read_csv('DATA.csv', delimiter=';')
```

Figura 3.14: Lectura de datos

Una vez que se exportaron los datos lo siguiente que se debe hacer es verificar la información. Se tienen 3 canales para realizar la predicción en la Figura 3.15 se elige un canal para poder graficarlo. En este caso se utilizó la función plt.plot para obtener la gráfica del canal 2.

```

#Se escoje canal para prediccion
rssi=df.CH2.values
#se grafican datos de entrada
plt.figure(figsize=(18,5))
plt.plot(rssi[:len(rssi)])
plt.show()
print('Number of rows and columns:', df.shape)
print (df.head(5))
len(df)

```

Figura 3.15: Canal para predicción

Cuando se realice las mediciones de potencia de los canales siempre se obtendrá un error, a estos se los conoce como datos aberrantes, estos datos causan que la predicción no sea tan eficiente. En la Figura 3.16 se utiliza la función hist para poder realizar un histograma de los datos del canal y así saber de mejor manera como está distribuida la potencia de todos los datos almacenados en el canal 2.

```

#eSTADISTICA para Limitar datos aberrantes.
CANAL='CH2'
plt.figure(figsize=(14,6))
df[CANAL].hist(bins=50)
plt.title('Potencia Distribution');

```

Figura 3.16: Limitar datos aberrantes

Para poder eliminar las aberraciones se tuvo que limitar los datos de toda la información del canal 2, se tuvo que sacar el promedio de todos los datos para conocer por donde estaba la media y así tomar un rango de valores para los nuevos datos. En la Figura 3.17 se utilizó un for para establecer un límite superior e inferior en los datos y así poder obtener la nueva gráfica a través de la función plot.

```

#Se limitan valores a media+-3.5*varianza
media_data=np.mean(rssi, axis = None);
#print(media_data);
std_data=np.std(rssi, axis = None);
#print(std_data)
max_data=media_data+2.5*std_data; #OJO se cambio a 2.5
min_data=media_data-2.5*std_data; # OJO se cambio a -2.5
rssi1=rssi
for i in range (len(rssi)):
    if rssi[i]>max_data:
        rssi1[i]=max_data
    if rssi[i]<min_data:
        rssi1[i]=min_data

plt.figure(figsize=(18,5))
plt.plot(rssi1[:])
plt.show()

```

Figura 3.17: Limitación de datos

LSTM es un método que aprende continuamente, entre más datos se tenga mejor será la predicción. En este caso se le dio el 80% de los datos para que pueda entrenar y el 20% restante son para la predicción. Como se puede ver en la Figura 3.18 se realizó lo mencionado anteriormente para que el algoritmo diseñado pueda trabajar de una forma eficiente.

```

#dividiendo la data para entrenamiento, pruebas y predicciones
spec_size1= int(len(rssi1) * 0.20)# 0.20 tamaño datos para predicción
dataset_model =len(rssi1)-spec_size1 #tamaño datos para entrenamiento y prueba
print(dataset_model, spec_size1)

```

Figura 3.18: Datos para predicción y entrenamiento

En cualquier método de predicción siempre es bueno tener los datos ordenados en base al tiempo, por esto en la Figura 3.19 se realiza la asignación del tiempo a los respectivos datos. En t se asigna un intervalo de 7.2 seg y en t1 se asigna un intervalo de 0.01 seg para cada uno de los datos.

```

# definiendo el tiempo

t=np.arange(0,len(rssi1)*8,7.2);#realizando intervalos de 7.2 seg
t1=np.arange(0,len(t)*8,0.01);#realizando intervalos de 0.01 seg

print('#t:',len(t),'#t1:',len(t1))

```

Figura 3.19: Tiempo para los datos

Todo método para la predicción de espectro es difícil de tratar debido a que el espectro cambia constantemente de una manera abrupta, por tal motivo antes de ingresar los datos directamente al programa se debe tratarlos. En la Figura 3.20 se realiza una modulación para los datos, ya que esto ayuda a que el algoritmo LSTM pueda predecir de una manera más eficiente los canales disponibles en el espectro.

Primeramente, para la modulación de los datos se realizó la fase modulante, lo cual consiste en multiplicar todos los datos de entrenamiento con una fase que en este caso es 0.285. Luego debido a que los valores del espectro están muy dispersos en un rango muy grande, se procede a establecer otro nuevo rango que va desde -1 a 1, al final obtendremos dos señales que son la señal portadora y la señal modulada.

```

## modulando
import math
from decimal import *
#senial modulada=cos(angulo); angulo=argum+fidt
fidt=[];
fidt=(inp*mdt);#fase modulante
argum=[]
argum=np.pi*2*fcent*t1[spec_size1*800:];#fase de portadora
argum=argum.reshape(-1,1);#formando arreglo
argum_m=[]
argum_m=np.pi*2*fcent*t[spec_size1:spec_size1+dataset_model];
argum_m=argum_m.reshape(-1,1);#formando arreglo

fase=[];#angulo=argum+fidt
fase=fidt+argum_m;
y_2=[]
y_1=[]
for i in range(len(t1)-1-spec_size1*800):
    y_2.append(np.cos(argum[i]))# senial portadora

for i in range(dataset_model):#len(t)-spec_size1-1
    y_1.append(np.cos(fase[i]))# senial modulada

from array import array
import numpy as np
y_1=np.asarray(y_1)
y_2=np.asarray(y_2)

```

Figura 3.20: Modulación de datos

Ya realizada la modulación de datos se procede a graficar las señales, para esto se utiliza el código que se puede observar en la Figura 3.21 para la señal portadora y modulada, y en la Figura 3.22 en cambio se observa la señal modulada vs cambio de fase.

```
#Portadora y señal "cambio de fase" de señal modulada
plt.figure(figsize=(18,5))
plt.plot(t1[spec_size1*720:spec_size1*720+29520], y_2[:29520], color = 'b')
plt.stem(t[spec_size1:spec_size1+41], y_1[:41], 'k', "o",)
plt.title("Señal Portadora y Cambio de Fase señal Modulada")
plt.show()
```

Figura 3.21: Código de la señal portadora y modulada

```
#señal modulante vs. cambio de fase
plt.figure(figsize=(18,5))
plt.plot(t[spec_size1:spec_size1+400], mdt[:400], color = 'b', marker = "*",)
#plt.stem(t[spec_size1:spec_size1+400], y_1[:400], 'k', "o",)
#plt.stem(t[spec_size1:spec_size1+400], mdt[:400], 'b', "*",)
plt.title("Señal Modulante")
#plt.xlabel("X")
#plt.ylabel("Y")
plt.show()
```

Figura 3.22: Código de la señal modulada vs cambio de fase

Ya realizado la modulación de los datos se procede a tomar la señal modulada para poder convertirla a un vector 1D, para luego normalizar los datos y realizar la predicción. En la Figura 3.18 se sacó el 80% de toda la DATA, y en cambio en la Figura 3.23 se sacó el 80% de los nuevos datos para entrenamiento y el 20 % para predecir.

```
#Define tamaño de archivo para entrenamiento y pruebas
#y_1=np.asarray(y_1)
dataset = y_1#valores de los cambios de fase de la señal modulada
dataset = dataset.astype('float32')
dataset = np.reshape(dataset, (-1, 1)) #Lo convierte en un vector 1D

scaler = MinMaxScaler(feature_range=(-1, 1)) # funcion para normalizar
dataset = scaler.fit_transform(dataset) # normaliza los datos

train_size = int(dataset_model* 0.80) #toma el 80% de los datos para entrenamiento
test_size = dataset_model - train_size #deja el 20% de datos para pruebas
#spectrum=dataset[0:spec_size,:];#archivo para prediccion
train, test = dataset[:train_size:], dataset[train_size:train_size+test_size:] # asigna datos
```

Figura 3.23: Entrenamiento y pruebas

Para que el algoritmo sea más eficiente se dispuso a crear una función la cual permite obtener los datos separados de entrenamiento y prueba. Esta función tiene tres entradas la primera es la base de datos con la que se está trabajando, la segunda es la característica para entrenar los datos, que en este caso tiene el valor de 90 y por último está el ancho de ventana de predicción, en la Figura 3.24 se puede observar lo descrito anteriormente.

```
#Función para crear archivos de entrenamiento y prueba
#ft : features to train
#wd : size of window

def create_dataset(dataset, ft,wd):
    X, Y = [], []
    for i in range(len(dataset)-ft-1):
        a = dataset[i:(i+ft),0]
        X.append(a)
        b = dataset[(i+ft):(i+ft+wd),0]
        Y.append(b)
    return np.array(X), np.array(Y)
```

Figura 3.24: Creación de una función para entrenamiento y prueba

Cuando se creó la función para el entrenamiento y las pruebas, luego se la utiliza para crear todas las entradas y salidas del entrenamiento. A todos estas variables se las debe convertir en arreglos como se puede ver en la Figura 3.25.

```
#Creando entradas y salidas para entrenamiento y pruebas

X_train, Y_train = create_dataset(train, feature,wd);#entradas y salidas para entrenamiento
X_test, Y_test = create_dataset(test, feature,wd);#entradas y salidas para pruebas
X_train= np.array(X_train);#formando arreglo
X_test=np.array(X_test);#formando arreglo
X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]));#formando archivo 3D
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]));#formando archivo 3D
print(len(X_train))
```

Figura 3.25: Entradas y salidas para el entrenamiento de datos

Para la predicción de los datos se utilizó las funciones que se encuentran en la Figura 3.26, en este método de LSTM se entrenó 500 veces los datos para obtener una predicción más precisa. Después de realizar el entrenamiento se procede a hacer 20 épocas del modelo, esto quiere decir que se están realizando 20 predicciones, para luego

sacar un promedio de estos datos ya que con una sola muestra no se puede concluir de una forma confiable.

Los resultados de la ejecución de este código van a hacer distintos, pero no deben variar mucho, ya que todas las predicciones realizadas tienen los mismos datos de entrenamiento.

```
#Modelo LSTM, prediccion
b_siz=1024 ###
model = Sequential()
model.add(LSTM(500, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='sgd', metrics=['accuracy'])
epocas_model=20
history = model.fit(X_train, Y_train, epochs=epocas_model, batch_size=b_siz, validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)
model.summary()
```

Figura 3.26: Modelo LSTM para predicción de datos

Para comprobar que el modelo está trabajando correctamente en la Figura 3.27 se procedió a sacar el error cuadrático medio de la predicción, en este caso mientras menor sea el valor del error cuadrático medio, mejor será el modelo al momento de predecir los datos, pero por otra parte mientras mayor sea el error quiere decir que el modelo no es muy eficiente.

```
train_error=history.history['loss']
test_error=history.history['val_loss']
from sklearn.metrics import mean_squared_error
error_train_test=mean_squared_error(train_error,test_error);
print('error cuadratico medio del modelo LSTM :',error_train_test)
```

Figura 3.27: Error cuadrático del modelo LSTM

En la Figura 3.28 se puede observar el código para la corrección de fase de los datos, esto se realiza después de la modulación para corregir los valores y así se pueda realizar la predicción de una mejor manera en el algoritmo.

```

#Calcula correccion de fase
fase_equiv=[]
corrige_fase=[]
for i in range(len (spectrum)):
    fase_equiv.append(np.arccos(y_1p[i]))
    corrige_fase.append(fase_p[i]-fase_equiv[i]) #####SACAR EL 0.8
fase_equiv=np.asarray(fase_equiv)
corrige_fase=np.asarray(corrige_fase)
print(corrige_fase[20],fase_p[20],fase_equiv[20])

```

Figura 3.28: Corrección de fase

Para observar de una mejor manera la señal portadora, moduladora y modulada se realizó el código que se puede encontrar en la Figura 3.29, la cual muestra las tres señales en un solo grafico para así poder tener una referencia entre las señales y verificar que estén trabajando correctamente.

```

#Portadora y senial modulada
spectrum2=spectrum
spectrum2 = scaler.fit_transform(spectrum2)
plt.figure(figsize=(18,10))
plt.plot(t1[:72000], y_2p[:72000], color = 'c')
plt.stem(t[:100], y_1p[:100], 'r','o')
plt.plot(t[:100], spectrum2[:100],'b')
plt.ylabel('Normalized signals', size=30)
plt.xlabel('time (s)', size=30)

plt.legend(('Carrier signal','Modulating signal', 'Modulated signal'),loc='upper right', fontsize=18)

plt.grid()
plt.show()

```

Figura 3.29: Señal portadora y modulada

Ya que se tiene la señal modulada y se mejoró los errores que tenían los datos, después se realiza la predicción. Para esto se utilizó una función que ya se la estableció anteriormente que se llama `create_dataset` la cual permite separar datos para el entramiento y prueba. La función `model.predict` ayudo a la predicción de los valores del espectro, luego de realizar esto se obtuvo el error cuadrático medio como se puede ver en la Figura 3.30.

```

#Hacer las predicciones DE LA SENIAL MODULADA para nuevos datos
y_1p = scaler.fit_transform(y_1p) # normaliza los datos
X_new, Y_new = create_dataset(y_1p,feature,wd) # create dataset

X_new = np.reshape(X_new, (X_new.shape[0], 1, X_new.shape[1])) #Features to model

#Y_new=Y_new.reshape(len(Y_new),1) # Ajust the output
#Y_new=repair_set(Y_new,wd)
Y_predict = model.predict(X_new) #Generate Prediction
#Y_predict=Y_predict.reshape(len(Y_predict),1) #Ajust prediction
from sklearn.metrics import mean_squared_error
error_pred=mean_squared_error(Y_new,Y_predict);
print('error cuadratico medio de la prediccion',error_pred)

```

Figura 3.30: Predicción para señal modulada

Todos los datos se encuentran entre 1 y -1 y al momento de hacer la predicción algunos valores no estarán dentro de este rango, para esto en la Figura 3.31 se implementó un código que permita corregir estos valores. Cuando los valores son mayores que 1 el código los transforma en 1 y en el otro caso cuando son menores a 1 los transforma a -1.

```
#Filtra predicciones mayores que 1 y menores que -1
for i in range (len(Y_predict)):
    if Y_predict[i]>1.0:
        Y_predict[i]=1.
    if Y_predict[i]<-1.0:
        Y_predict[i]=-1.0
```

Figura 3.31: Filtro para predicción mayores que 1 y menores que -1

Cuando se trataron los datos se les realizó una modulación para que así al momento de realizar la predicción sea más eficiente, pero cuando se hizo esto a los datos originales se los llevó a otro rango por tal motivo en la Figura 3.32 se elaboró un código para que los datos vuelvan a su rango original

```
#Demodulacion de la prediccion
fidt_1=[]
mdt_1=[]
for i in range (len(Y_predict)):
    fidt_1.append((fase_corregida[i]-argum2[i]))#####
    mdt_1.append(fidt_1[i]*indim) #senal demodulada
```

Figura 3.32: Demodulación de datos

Después de realizar la predicción del espectro, se procedió a calcular la disponibilidad del canal en el espectro, para esto se creó una función que está definida en la Figura 3.33. Esta función posee dos entradas una es un conjunto de datos y la otra entrada es el valor threshold que en este caso es -75 dBm. Al final se tendrá una salida que será una lista de datos conformada por 0 y 1, esto representa si el canal se encuentra disponible u ocupado.

```

#Función para calcular disponibilidad
def disponibilidad(dataset,th):
    predict=[]
    predict_val=0
    for i in range(len(dataset)):
        if dataset[i]<th:
            predict.append(0);
            predict_val+=1
        else:
            predict.append(1);
    return predict, predict_val

```

Figura 3.33: Función de disponibilidad

Para saber el porcentaje de disponibilidad del canal a utilizar se realizó el código de la Figura 3.34, donde se obtuvo la disponibilidad real del canal, la disponibilidad de predicción con el método LSTM y por último el error relativo de la disponibilidad del canal en el espectro.

```

#calculando disponibilidad
th=-75 #-75
mdt_orig_disp=mdt_orig[:] #limitando muestras
mdt_1_disp=mdt_1[:] #limitando muestras
X_orig,Xd_orig= disponibilidad(mdt_orig_disp, th)
Y_pr,Yd_pr= disponibilidad(mdt_1_disp, th)

#Disponibilidad porcentual
d1=(Xd_orig/len(X_orig))*100
d2=(Yd_pr/len(Y_pr))*100
Dre=(d2-d1)/d1

print('Disponibilidad real:',d1)
print('Disponibilidad predict:',d2)
print('Error relativo de Disponibilidad:',Dre)

```

Figura 3.34: Porcentaje de disponibilidad real y de predicción

Para evaluar el rendimiento del método LSTM se procedió a crear una matriz de confusión, a través del código que está en la Figura 3.35. En esta matriz se obtuvo un parámetro muy importante que es la exactitud del método implementado LSTM.

```

#Cuantificando la precision
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
print('Availability')
print(confusion_matrix(X_orig,Y_pr))
print("Accuracy")
print(accuracy_score(X_orig,Y_pr))

```

Figura 3.35: Matriz de confusión de la Disponibilidad

Al final de ya haber realizado la predicción del espectro y la disponibilidad del canal se obtuvo parámetros muy importante para ver que tan bueno es el modelo. Se obtuvo el error cuadrático del modelo LSTM, la señal modulada y de la demodulación, además se sacó el valor del factor de correlación Spearman para la señal modula vs la señal predecida y de la señal original vs la señal demodulada. Otros parametros que se encontraron fue la disponibilidad real y predicha del canal, a parte gracias a estos se pudo obtener el error relativo de disponibilidad. Por último se obtuvo la matriz de confusión de la disponibilidad con la cual se encontró la exactitud del modelo.

```

print('error cuadratico medio del modelo LSTM :',error_train_test)
print('error cuadratico medio de la prediccion de senial modulada',error_pred)
print('Factor de Correlacion Spearman, senial modulada vs senial predecida: %.5f' % corr1)
print('error cuadratico medio en demodulacion :',error_demod)
print('Factor de Correlacion Spearman,senial original vs senial demodulada: %.5f' % corr)
print('Error del modelo',test_error[finish-1])
print('Disponibilidad real:',d1)
print('Disponibilidad predict:',d2)
print('Error relativo de Disponibilidad:',Dre)
print('Matriz de Confusion de la Disponibilidad')
print(confusion_matrix(X_orig,Y_pr))
print("Availability-Prediction Accuracy")
print(accuracy_score(X_orig,Y_pr))

```

Figura 3.36: Parámetros obtenidos con el método LSTM

CAPITULO 4

4.1 Resultados

En esta sección se hablará acerca de los gráficos y datos que se obtuvieron al momento de usar los métodos de predicción de LSTM y ARIMA, luego se analizará los resultados obtenidos para ambos métodos los cuales fueron la probabilidad de predicción de los canales disponibles, el RMSE, la eficiencia de acuerdo con el tamaño de la muestra y el tiempo que se demoran en ejecutarse los algoritmos. Realizando una comparación de los resultados de ambos métodos se puede determinar cuál es más eficiente al momento de predecir la disponibilidad de un canal. Para poder visualizar de una mejor manera los datos se elaborará una tabla acerca de estos dos métodos.

4.2 Método LSTM

Al haber 3 canales se dispondrá de 3 predicciones para este método, en este caso primero se realizó una predicción del espectro que se muestra en la Figura 4.1, 4.2 y 4.3 las cuales corresponden al canal 1, canal 2 y canal 3 respectivamente. Después se elaboró la predicción de la disponibilidad que tendrá el canal lo cual se lo podrá observar más adelante.

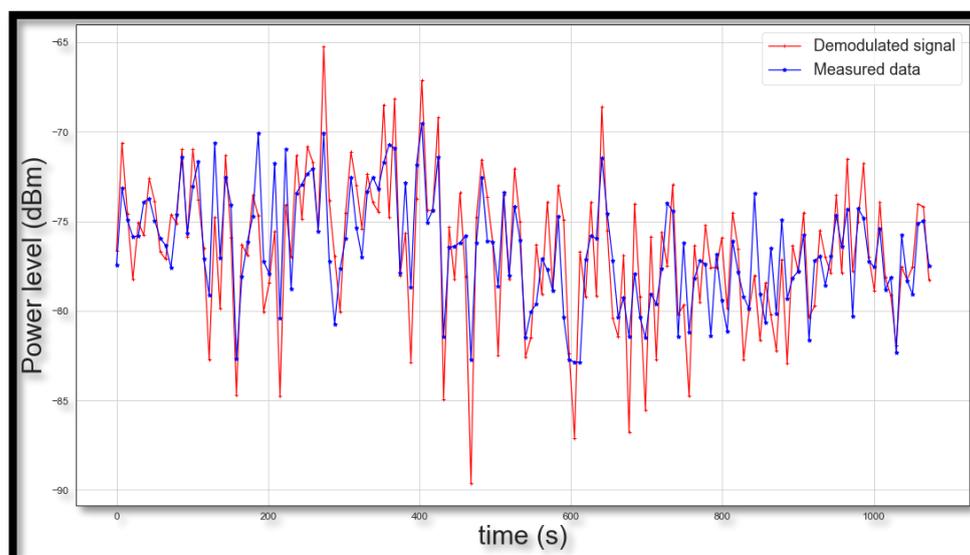


Figura 4.1: Predicción del espectro para el canal 1

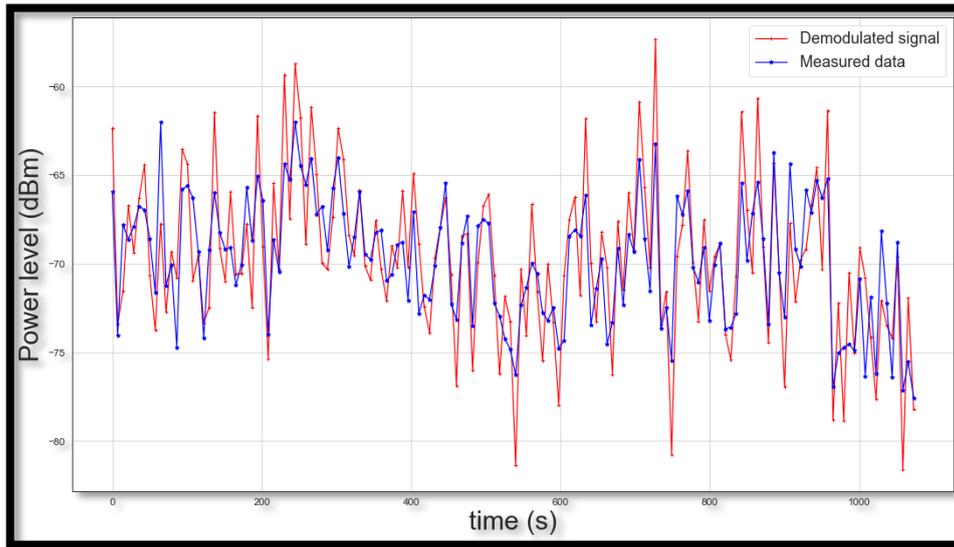


Figura 4.2: Prediccion del espectro para el canal 2

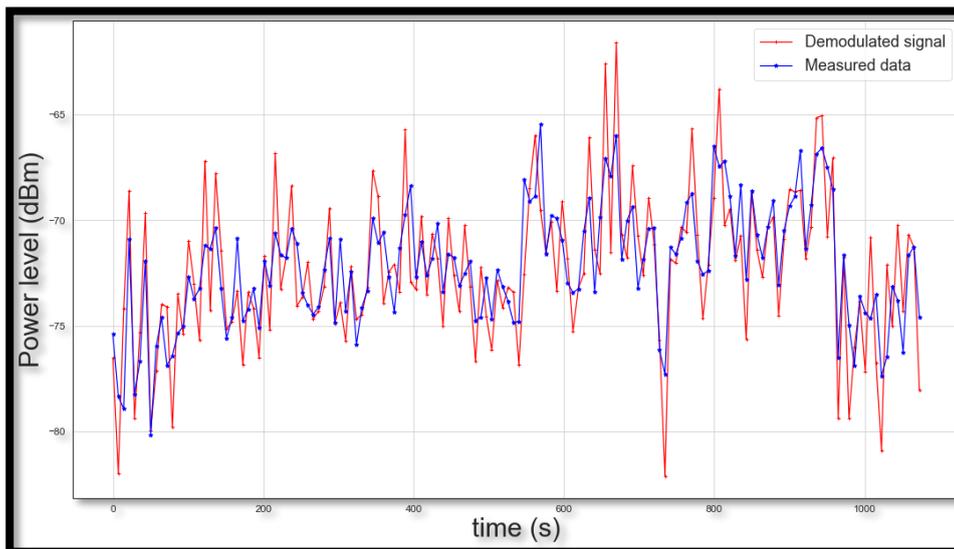


Figura 4.3: Prediccion del espectro para el canal 3

Para todos los canales se tomó el 80% de los datos para el entrenamiento y el 20% para la predicción, en la Figura 4.1, 4.2 y 4.3 la señal de color rojo es el espectro original y la señal de color azul es la predicción. En todas las Figuras mencionadas anteriormente se puede decir que el método LSTM hace una muy buena predicción del espectro ya que la gráfica original con la que se predijo no es tan diferente.

4.3 Método ARIMA

Para este método se elaboró dos algoritmos uno en Python y uno en RStudio, al momento de su ejecución se decidió trabajar con el algoritmo realizado en Python debido a que este mostraba mejores resultados al momento de la predicción del espectro, aunque para la predicción de disponibilidad ambos métodos mostraban un resultado similar.

En el método de ARIMA también se realizó la predicción de los 3 canales que se tiene, así mismo en este caso se utilizó el 80% de los datos para el entrenamiento y el 20% para la predicción. En las Figuras desde la 4.4 hasta la 4.6 se puede observar la predicción del espectro.

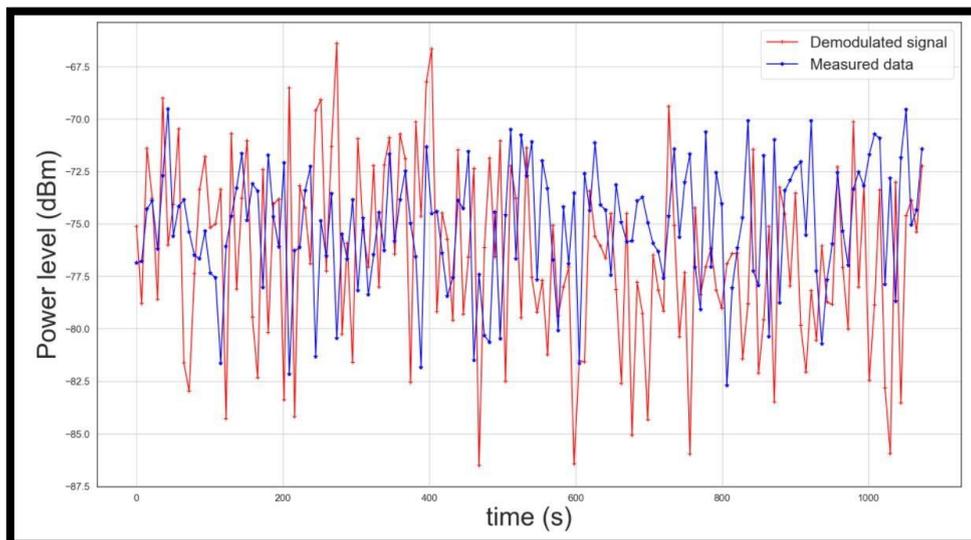


Figura 4.4: Predicción del espectro para el canal 1

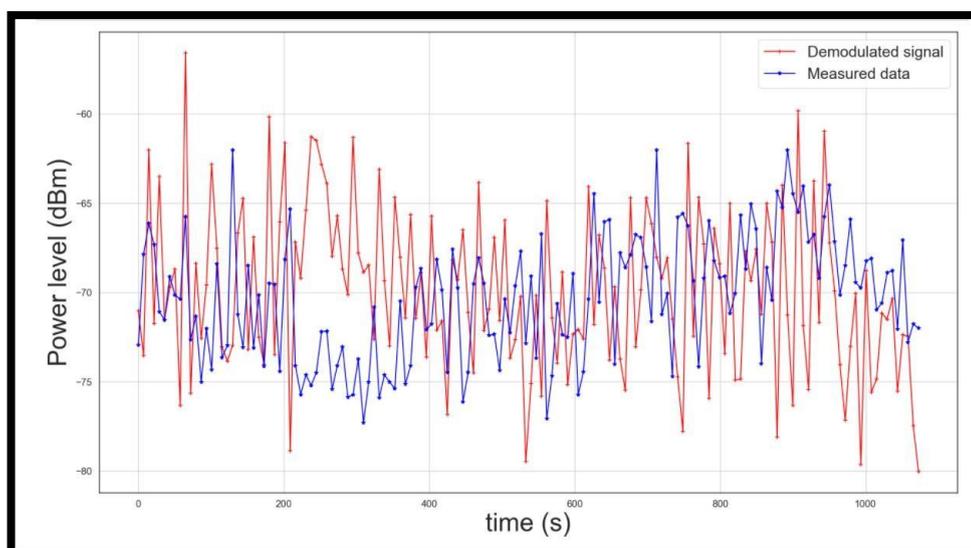


Figura 4.5: Predicción del espectro para el canal 2

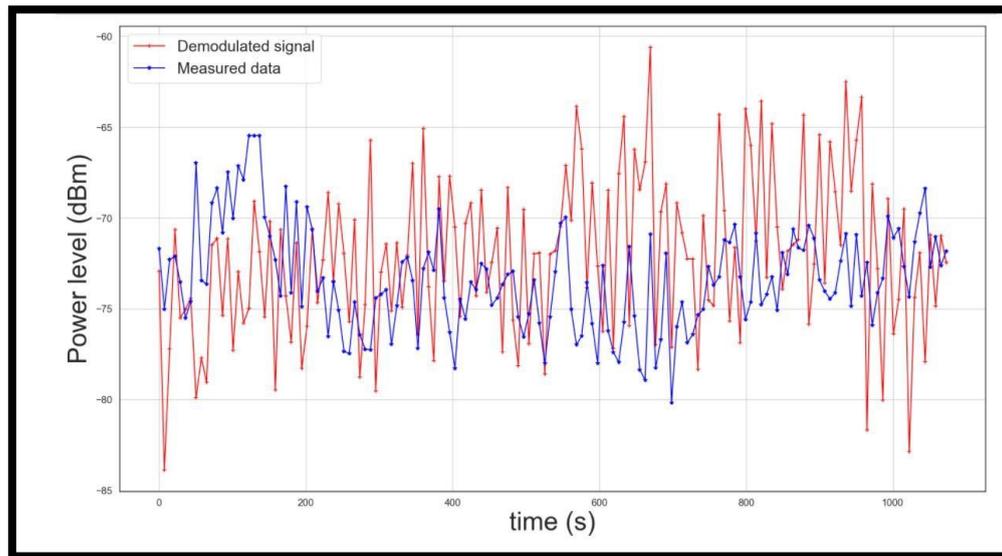


Figura 4.6: Predicción del espectro para el canal 3

Al usar el método de ARIMA se puede decir que no es un buen método para la predicción de espectro ya como se observa en la Figura 4.4, Figura 4.5 y Figura 4.6 la señal original con la señal que se predijo no es igual. Para poder concluir si este método es bueno o malo prediciendo la disponibilidad se elaboró una tabla que se la mostrará más adelante, donde indicará el RMSE de este método, además de la efectividad de ARIMA.

4.4 Comparación de los métodos LSTM y ARIMA

Para poder comparar estos métodos se elaboró tablas donde mostrará el RMSE y la efectividad de los algoritmos que se realizó. En la tabla 4.1 se puede observar la exactitud del modelo de ARIMA y LSTM, cabe recalcar que estos resultados son usando el 80% de los datos para el entrenamiento y 20% para la predicción. Además, se tomó 10 resultados debido a que LSTM al ser una red neuronal sus predicciones cambiarán cada vez que se ejecute el algoritmo.

EXACTITUD DEL MODELO PARA LA DISPONIBILIDAD DEL CANAL						
	MODELO LSTM			MODELO ARIMA		
	CANAL 1	CANAL 2	CANAL 3	CANAL 1	CANAL 2	CANAL 3
RESULTADO 1	0.8080015	0.8691551	0.8238856	0.6934	0.7971	0.6872
RESULTADO 2	0.8085972	0.8692544	0.8212052	0.6934	0.7971	0.6872
RESULTADO 3	0.8068102	0.8704457	0.8219994	0.6934	0.7971	0.6872
RESULTADO 4	0.8086965	0.8690558	0.8221979	0.6934	0.7971	0.6872
RESULTADO 5	0.8109798	0.8696515	0.8206095	0.6934	0.7971	0.6872
RESULTADO 6	0.8077037	0.8690558	0.8203117	0.6934	0.7971	0.6872
RESULTADO 7	0.8084979	0.8701479	0.8220986	0.6934	0.7971	0.6872
RESULTADO 8	0.8092921	0.8696515	0.8214037	0.6934	0.7971	0.6872
RESULTADO 9	0.8093914	0.8692544	0.8229921	0.6934	0.7971	0.6872
RESULTADO 10	0.8090936	0.8700486	0.8237863	0.6934	0.7971	0.6872
PROMEDIO	0.8087063	0.8695720	0.8220490	0.6934	0.7971	0.6872

Tabla 4.1: Exactitud del modelo LSTM y ARIMA

Los datos del método LSTM cambian, pero los de ARIMA no, esto se debe a que ARIMA al momento de predecir los datos trabaja con series temporales, en el cual crea un modelo que se basa en los parámetros p , d y q . Por tal motivo por más que se ejecute el algoritmo los resultados serán iguales ya que como los datos no cambian siempre creará el mismo modelo y los datos predichos siempre serán los mismos.

Al observar la tabla 4.1 se puede concluir que para este caso el modelo de LSTM tiene una mejor predicción de disponibilidad ya que para el canal 1 tiene una exactitud de 0.8087063, el canal 2 de 0.8695720 y el canal 3 de 0.8220490, en cambio ARIMA tiene una exactitud menor a esto ya que el canal 1 es de 0.6934, el canal 2 es de 0.7971 y el canal 3 de 0.6872.

Otro de los parámetros que se evaluó es el RMSE ya que a través de este se puede saber el error que tendrá el modelo. En la tabla 4.2 se encuentran los resultados del RMSE de los modelos LSTM y ARIMA. Aquí se puede concluir que el método LSTM tiene menor error en comparación a ARIMA ya que LSTM tiene un error para el canal 1 de 3.9002554, el canal 2 de 4.9326732 y el canal 3 de 3.0402982, en cambio ARIMA tiene un error mayor debido a que el canal 1 tiene un valor de 6.539086, el canal 2 de 8.972167 y el canal 3 de 4.687055.

RMSE DEL MODELO EN LA DEMODULACION						
	MODELO LSTM			MODELO ARIMA		
	CANAL 1	CANAL 2	CANAL 3	CANAL 1	CANAL 2	CANAL 3
RESULTADO 1	3.9017958	4.9468330	3.0322180	6.539086	8.972167	4.687055
RESULTADO 2	3.8935772	4.9294647	3.0433167	6.539086	8.972167	4.687055
RESULTADO 3	3.8940886	4.9201097	3.0404045	6.539086	8.972167	4.687055
RESULTADO 4	3.8997936	4.9293965	3.0529900	6.539086	8.972167	4.687055
RESULTADO 5	3.8924469	4.9274041	3.0358720	6.539086	8.972167	4.687055
RESULTADO 6	3.9110961	4.9222820	3.0427334	6.539086	8.972167	4.687055
RESULTADO 7	3.8983697	4.9164849	3.0247370	6.539086	8.972167	4.687055
RESULTADO 8	3.8983435	4.9379613	3.0528195	6.539086	8.972167	4.687055
RESULTADO 9	3.9113824	4.9564951	3.0375437	6.539086	8.972167	4.687055
RESULTADO 10	3.9016603	4.9403007	3.0402343	6.539086	8.972167	4.687055
PROMEDIO	3.9002554	4.9326732	3.0402982	6.539086	8.972167	4.687055

Tabla 4.2: RMSE del modelo LSTM y ARIMA

Cabe recalcar que los resultados que se muestran en la tabla 4.2 corresponden al error que existe en la demodulación de los datos, este no es el error de la disponibilidad. El método LSTM es muy bueno prediciendo el espectro, pero también tiene sus fallas, en cambio ARIMA no es muy bueno prediciendo el espectro. Ambos métodos al momento de predecir la disponibilidad mejoran debido a que se asigna un valor de threshold de -75 dBm, donde los valores que sean mayores representan que un canal está ocupado y los valores menores indican que un canal está disponible.

Para concluir que algoritmo es más eficiente de acuerdo con el tamaño de la muestra se elaboró dos tablas más, donde se utilizó el 70% de los datos para el entrenamiento y el 30 % para la predicción. En la tabla 4.3 se muestra el nuevo valor de exactitud para los modelos cambiándole estos parámetros, para LSTM se obtuvo una exactitud en el canal 1 de 0.8091976, el canal 2 de 0.86679197 y el canal 3 de 0.8349691, y para ARIMA los resultados en el canal 1 fue de 0.7116666, el canal 2 de 0.8098 y el canal 3 de 0.7066666.

EXACTITUD DEL MODELO PARA LA DISPONIBILIDAD DEL CANAL						
	MODELO LSTM			MODELO ARIMA		
	CANAL 1	CANAL 2	CANAL 3	CANAL 1	CANAL 2	CANAL 3
RESULTADO 1	0.8099102	0.8665874	0.8342570	0.7116666	0.8098	0.7066666
RESULTADO 2	0.8091844	0.8671153	0.8347189	0.7116666	0.8098	0.7066666
RESULTADO 3	0.8101741	0.8667854	0.8344550	0.7116666	0.8098	0.7066666
RESULTADO 4	0.8082607	0.8653998	0.8345209	0.7116666	0.8098	0.7066666
RESULTADO 5	0.8081947	0.8666534	0.8350488	0.7116666	0.8098	0.7066666
RESULTADO 6	0.8088545	0.8674452	0.8337292	0.7116666	0.8098	0.7066666
RESULTADO 7	0.8092504	0.8675771	0.8360385	0.7116666	0.8098	0.7066666
RESULTADO 8	0.8088545	0.8667194	0.8355766	0.7116666	0.8098	0.7066666
RESULTADO 9	0.8105040	0.8666534	0.8366983	0.7116666	0.8098	0.7066666
RESULTADO 10	0.8087885	0.8669833	0.8346480	0.7116666	0.8098	0.7066666
PROMEDIO	0.8091976	0.86679197	0.8349691	0.7116666	0.8098	0.7066666

Tabla 4.3: Exactitud del modelo LSTM y ARIMA

La tabla 4.3 en comparación con la tabla 4.1 indica de acuerdo con los resultados obtenidos que el método ARIMA mejora sus resultados cambiando el tamaño de la muestra, esto se debe a que ARIMA crea un modelo para poder realizar la predicción y en este caso está creando un mejor modelo con el 70 % de los datos para el entrenamiento. Para LSTM los resultados no varían mucho, aunque entre más datos se tenga en el método LSTM este realizará una mejor predicción debido a que este trabaja con redes neuronales recurrentes que aprenden constantemente.

Cambiando el tamaño de la muestra el RSME aumenta para ambos algoritmos, esto se debe a que ahora se tendrán más datos para predecir y por lo tanto esto generará más error en los modelos. Como se puede ver en la tabla 4.4 se tiene que el RMSE del método LSTM para el canal 1 es de 4.1137999, el canal 2 de 5.3553889 y el canal 3 de 3.0880093, en cambio para ARIMA se obtiene en el canal 1 un valor de 6.624848, el canal 2 de 9.687838 y el canal 3 de 4.711437.

RMSE DEL MODELO EN LA DEMODULACION						
	MODELO LSTM			MODELO ARIMA		
	CANAL 1	CANAL 2	CANAL 3	CANAL 1	CANAL 2	CANAL 3
RESULTADO 1	4.1092874	5.3465861	3.0939154	6.624848	9.687838	4.711437
RESULTADO 2	4.1173170	5.3731562	3.0771476	6.624848	9.687838	4.711437
RESULTADO 3	4.1069931	5.3652652	3.0812017	6.624848	9.687838	4.711437
RESULTADO 4	4.1144946	5.3562374	3.1002742	6.624848	9.687838	4.711437
RESULTADO 5	4.1055252	5.3398973	3.0997197	6.624848	9.687838	4.711437
RESULTADO 6	4.1149113	5.3571617	3.1023925	6.624848	9.687838	4.711437
RESULTADO 7	4.1187984	5.3448384	3.0778034	6.624848	9.687838	4.711437
RESULTADO 8	4.1219245	5.3480102	3.0867880	6.624848	9.687838	4.711437
RESULTADO 9	4.1143170	5.3622459	3.0851147	6.624848	9.687838	4.711437
RESULTADO 10	4.1144314	5.3604907	3.0757363	6.624848	9.687838	4.711437
PROMEDIO	4.1137999	5.3553889	3.0880093	6.624848	9.687838	4.711437

Tabla 4.4: RMSE del modelo LSTM y ARIMA

Por último, para obtener el tiempo de ejecución de los algoritmos se tomó el lapso que demora cada algoritmo en predecir los datos, aquí está incluido el tiempo que tarda en formar el modelo. Para esto en la tabla 4.5 se puede observar el tiempo de ejecución para cada método de acuerdo con el tamaño de la muestra empleada.

TIEMPO DE EJECUCIÓN DE LOS ALGORITMOS						
	Modelo LSTM			MODELO ARIMA		
	CANAL 1	CANAL 2	CANAL 3	CANAL 1	CANAL 2	CANAL 3
70% entrenamiento, 30% predicción	60,34 s	60,03 s	58,89 s	18,71 s	123 s	74,73 s
80% entrenamiento, 20% predicción	70,66 s	74,42 s	72,7 s	70,91 s	210,9 s	177,2 s

Tabla 4.5: Tiempo de ejecución del modelo LSTM y ARIMA

Al aumentar los datos del entrenamiento los algoritmos se demoran más, esto se debe a que estos métodos se tardan más creando su modelo que prediciendo los datos. ARIMA por su parte se observa que aumenta demasiado su tiempo en la predicción de datos cuando se añade más datos en la parte de entrenamiento.

CONCLUSIONES

- Se descubrió que cada canal tiene diferente comportamiento, por tal motivo para el algoritmo de LSTM al ser una red neuronal no necesitará cambios al momento de realizar la predicción de todos los canales, en cambio para ARIMA se debe realizar un cambio en los parámetros de acuerdo al canal que se utilice.
- Se investigó el funcionamiento del método de ARIMA donde su principal característica para que el algoritmo trabaje correctamente es que sus datos sean estacionarios, además este modelo se rige por tres parámetros muy importantes que se los denota como p , d y q . En el caso de los tres canales que se disponían todos eran estacionarios, por tal motivo no se tuvo que hacer un trabajo adicional para volverlos estacionarios.
- Se diseñó el algoritmo para el método de ARIMA en dos lenguajes de programación que son Python y RStudio, al momento de crear ambos códigos se observó que los resultados obtenidos en el programa Python eran más convenientes para la predicción del espectro.
- Se determinó los resultados con el método ARIMA donde se observó que este no es un buen método para la predicción de espectro debido a que este predice los datos de acuerdo con su comportamiento y tendencia, pero el espectro no posee un comportamiento ni tendencia fija ya que los datos cambian muy bruscamente de un rato a otro.
- Se investigó el método LSTM el cual trabaja con redes recurrentes lo que quiere decir que este recuerda estados previos, por tal motivo entre más datos tenga este modelo para el entrenamiento mejores serán sus resultados, aunque el tiempo de predicción será mayor.
- Se diseñó el algoritmo de LSTM en el programa Python, ya que este permite trabajar de una forma más sencilla las redes recurrentes, en este caso primero se trató los datos para luego poder realizar una mejor predicción de la disponibilidad.

- Se determinó los resultados para el método de LSTM, donde se pudo concluir que este método es muy bueno para la predicción de la disponibilidad, ya que se obtuvo un porcentaje de exactitud alto para la predicción de la disponibilidad en los tres canales.
- Se comprobó que el método LSTM no tiene cambios significativos al momento de variar el tamaño de la muestra de 80% a un 70% de datos para el entrenamiento, mientras que el método ARIMA si presenta mayor cambio al momento de variar el tamaño de la muestra, aquí ARIMA crea un mejor modelo con el 70% de los datos que con un 80%.
- Se pudo observar que el método LSTM tiene un menor tiempo de predicción de datos comparado con el método de ARIMA, aunque para mostrar todos los resultados finales ARIMA posee un tiempo menor que LSTM.
- Se evaluó la probabilidad de predicción de los canales disponibles de cada método, donde se concluyó que el modelo de LSTM tiene una mayor exactitud al momento de predecir la disponibilidad de los canales
- Se demostró que para los métodos LSTM y ARIMA el mejor para la predicción de canales disponibles en el espectro es el método de LSTM ya que es más eficiente al momento de predecir la disponibilidad, debido a que tiene una mayor exactitud, un menor tiempo de predicción de datos y un menor RMSE comparado con el método ARIMA.
- Se pudo comprobar que para el método LSTM una sola ejecución del código no basta para afirmar la exactitud y el RMSE del método, por ello se realizó 10 ejecuciones al algoritmo donde se obtuvieron resultados diferentes para poder sacar un promedio entre estos y así obtener un resultado final más preciso.

RECOMENDACIONES

- Se recomienda al momento de usar el método ARIMA para otros canales verificar que los datos sean estacionarios, esto se lo puede realizar a través de la gráfica o con un código en el programa utilizado, si no son se debe volverlos estacionarios para que el algoritmo funcione correctamente.
- Instalar todas las librerías a utilizar en el programa que se realice el algoritmo a través del cmd del computador y comprobar que estas funcionen correctamente antes de realizar el diseño del código, para así evitar problemas al momento de la ejecución del algoritmo.
- Al momento de utilizar el código de ARIMA para la predicción de la disponibilidad de un canal diferente a los que se trabajó, se recomienda cambiar los parámetros p , d y q , ya que para cada canal estos parámetros siempre van a hacer diferentes.
- Estos algoritmos de ARIMA y LSTM implementados funcionan únicamente para predecir el espectro, por tal motivo si se desea predecir alguna otra clase de datos se debe cambiar el código a conveniencia.

BIBLIOGRAFÍA

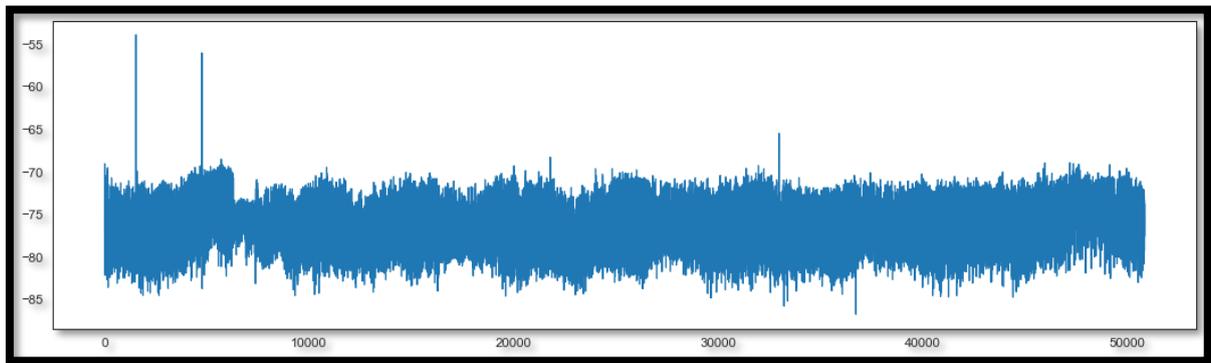
- [1] «Espectro en America latina,» 2016. [En línea]. Available: <data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAVQAAACUCAMAAAD70yGHAAAAGVB MVEX///8AAAD8/Pzr6+vd3d2pqanv7+/4+PjBwcG8vLxhYWG3t7fx8fH19fWTk5MsLCxpaWknJyFT 09MiliLHx8fj4+OZmZmFhYUaGhqfn59YWFhGRkbY2NilpaVmZmZSUIIzMzMNDQ1xcXE+Pj6BgYFK Skp5eXIAQECDg4OMjIwCHBw36QBMA>. [Último acceso: 10 Octubre 2021].
- [2] L. G. A. Lopez, «Banco Interamericano de Desarrollo,» Julio 2017. [En línea]. Available: <https://publications.iadb.org/publications/spanish/document/El-impacto-de-las-obligaciones-en-el-valor-del-espectro-radioelectrico.pdf>. [Último acceso: 10 Octubre 2021].
- [3] S. Galeano, «Marketing ecommerce,» 2021 Enero 28. [En línea]. Available: <https://marketing4ecommerce.net/usuarios-de-internet-mundo/>. [Último acceso: 10 Octubre 2021].
- [4] Expansion, «Expansion Economia digital,» Unidad Editorial Información General, S.L.U., 10 Febrero 2021. [En línea]. Available: <https://www.expansion.com/economia-digital/innovacion/2021/02/10/6022c89de5fdea59448b459b.html>. [Último acceso: 10 Octubre 2021].
- [5] P. Supraja y R. Pitchai, «Spectrum prediction in cognitive radio with hybrid optimized neural network,» Tamil Nadu, 2017.
- [6] C. P. M. L. F. & M. S. F. H. Hernández, «Algoritmos para asignación de espectro en redes de radio cognitiva,» 15 Febrero 2016. [En línea]. Available: <https://www.redalyc.org/pdf/2570/257046835006.pdf>. [Último acceso: 10 Octubre 2021].
- [7] X. X. A. T. JING, «SPECTRUM PREDICTION IN COGNITIVE RADIO NETWORKS,» Beijing, 2013.
- [8] B. F. Tavio, E. O. Guerra, J. C. P. García y W. P. Rodríguez, «Múltiples radios vs. consumo de energía en algoritmos de rendezvous para redes radio cognitivas,» *Scielo*, vol. 39, nº 3, 2018.
- [9] «Radio Cognitiva,» [En línea]. Available: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/894/A5.pdf?sequence=5>. [Último acceso: 2021 10 19].
- [10] F. F. I. P. P. Luis Fernando Pedraza, «Detección de espectro para radio cognitiva,» *Ingeniare*, vol. 20, nº 2, pp. 197-210, 2012.
- [11] C. Ruiz, «Especialización en Uso Racional y Eficiente de la Energía,» Veracruz, 2019.
- [12] B. R. J. J. M. Jaison jacob, «Spectrum Prediction In Cognitive Radio Networks: A Bayesian Approach,» Kochi, 2014.
- [13] P. W. y. D. N. VK Tumuluru, «Channel state prediction for radio networks,» Wiley Online Library, Nanyang, 2010.

- [14] I. M. O. B. ,. D. J. T. G. Ing. Luis M. Gato Díaz, «signals detection by coalitional formation in cognitive radio networks,» 2017.
- [15] A. M. Glandon, «Recurrent neural networks and matrix methods for the prediction and security of the radio spectrum,» Norfolk, 2017.
- [16] D. A. L. SARMIENTO, «IMPLEMENTACIÓN DE UN MODELO PREDICTOR PARA LA TOMA DE DECISIONES EN REDES INALAMBRICAS DE RADIO COGNITIVA,» Bogota, 2017.
- [17] J. C. Ordoñez, «Caracterización de usuarios primarios para la implementación de un modelo predictor para la toma de decisiones en redes inalámbricas de radio cognitiva,» repository.udistrital, Bogota, 2016.
- [18] D. J. M.-P. C. A. Jorge Alexis Rodríguez-García, «Evaluation of an algorithm for the selection of spectral opportunities in cognitive wireless network,» Bogota, 2019.
- [19] D. A. L. Sarmiento, E. R. Trujillo y O. G. Guerrero, «PREDICCIÓN ESPECTRAL EN REDES INALÁMBRICAS DE RADIO COGNITIVA,» E-ISSN: 2248 – 762X, Bogota, 2014.
- [20] R. B. López, «Análisis y diseño de algoritmos para la,» Santa Clara, 2012.
- [21] C. Hernández, L. F. P. Martínez y F. H. M. Sarmiento, «Algoritmos para asignación de espectro en redes de radio cognitiva,» *TECNURA*, vol. 20, nº 48, p. 88, 2016.
- [22] M. Bhatia y K. Kumar, «Digital Communications and Networks,» de *Network selection in cognitive radio enabled Wireless Body Area Networks*, Beijing, Chinese Academy of Engineering, Beijing, China, 2020, p. 144.
- [23] D. G. I. P. Cesar Hernández, «Hybrid Algorithm for Frequency Channel Selection in Wi-Fi Networks,» *International Journal of Electronics and Communication Engineering*, vol. 9, nº 12, 2015.
- [24] C. C. Calvo, «ENACOM,» 23 04 2017. [En línea]. Available: https://www.enacom.gob.ar/-que-es-el-espectro-radioelectrico-_p117.
- [25] J. L. Ordóñez, «Espectro electromagnético y espectro radioeléctrico,» *Manual formativo de ACTA*, nº 62, pp. 17-31, 2012.
- [26] A. L. Yáñez, «Gestión del espectro radioeléctrico en Ecuador,» Corporación Editora Nacional, Quito, 2013.
- [27] S. M. S. Raikel Bordón López, «La Radio Cognitiva y su Impacto en el Uso Eficiente del Espectro de Radio,» *Revista Científica de Ingeniería Electrónica, Automática y Comunicaciones*, vol. 36, nº 1, pp. 42-55, 2015.
- [28] E. C. Leskow, «Concepto,» 15 Julio 2021. [En línea]. Available: <https://concepto.de/frecuencia/>. [Último acceso: 30 Noviembre 2021].

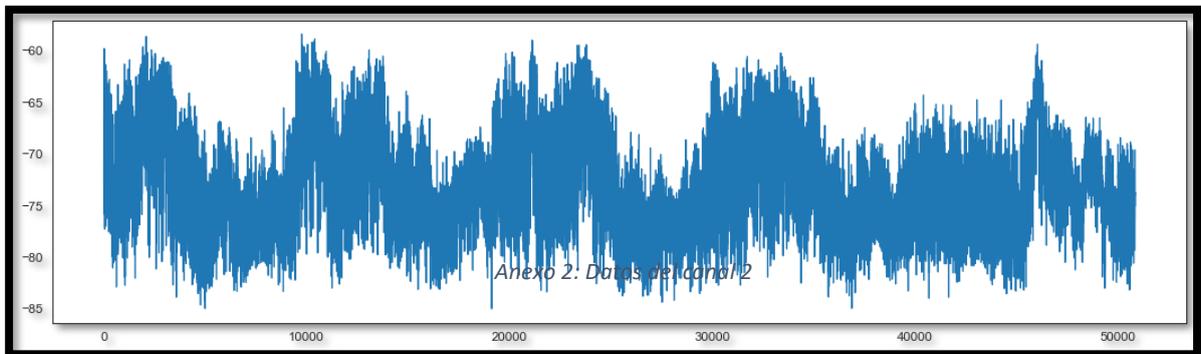
- [29] ANE, «Agencia Nacional del espectro,» [En línea]. Available: https://portalespectro.ane.gov.co/Style%20Library/ane_master/que-es-el-espectro-radioelectrico.aspx. [Último acceso: 07 Diciembre 2021].
- [30] S. G. Gago, «Radios Libres,» 30 Agosto 2014. [En línea]. Available: <https://radioslibres.net/14-que-es-el-espectro-radioelectrico/>. [Último acceso: 30 Noviembre 2021].
- [31] Axess Networks, «Axess Networks,» 2019. [En línea]. Available: <https://axessnet.com/que-es-ancho-de-banda-velocidad-y-latencia-en-una-conexion-de-internet/>. [Último acceso: 30 Noviembre 2021].
- [32] ITU, «Comprometida para conectar el mundo,» Febrero 2021. [En línea]. Available: <https://www.itu.int/es/mediacentre/backgrounders/Pages/radio-interference.aspx>. [Último acceso: 30 Novimebre 2021].
- [33] A. N. C. Julio Hector Aguilar Rentería, «Radio cognitiva,» *Sistemas y Telemática*, vol. 9, nº 18, pp. 31-53, 2011.
- [34] J. H. A. Rentería y A. N. Cadavid, «Radio cognitiva,» *Revista Sistemas y Telemática*, vol. 9, nº 16, pp. 31-53, 2011.
- [35] B. F. Tavio, E. O. Guerra, J. C. P. García y W. P. Rodríguez, «Múltiples radios vs. consumo de energía en algoritmos de rendezvous para redes radio cognitivas,» *Scielo*, vol. 39, nº 3, 2018.
- [36] R. C. C. J. R. C. Á. Gerardo Martínez Cruz, «Tecnologías de acceso dinámico y uso compartido del espectro,» Ricardo Flores Castañeda, Mexico, 2017.
- [37] D. Ospina, A. F. Ocampo y R. Hincapié, «Aplicación de esquemas de Acceso Oportunista al Espectro en ambientes inalámbricos heterogéneos,» *Revista en Telecomunicaciones e Informática*, vol. 1, nº 1, pp. 49-71, 2011.
- [38] L. P. I. P. Cesar Hernández y E. Rodriguez-Colina, «Análisis de la Movilidad Espectral en Redes de Radio Cognitiva,» *Scielo*, vol. 26, nº 6, 2015.
- [39] MathWorks, «MathWorks,» The MathWorks, Inc., [En línea]. Available: <https://de.mathworks.com/discovery/predictive-analytics.html>. [Último acceso: 26 Noviembre 2021].
- [40] S. R. Vahid Mirjalili, Python Machine Learning, Marcombo: Packt Publishing, 2019.
- [41] C. L. V. HERREÑO, «ESTUDIO DE DISPONIBILIDAD DE HUECOS ESPECTRALES EN,» BOGOTA, 2020.
- [42] J. P. C. Fredy Ocaris Pérez Ramírez, «compelect,» 21 04 2005. [En línea]. Available: https://compelect.com.co/archivos/diamatlab/2005/pdf/ARIMA,%20%20ARCH,%20%20GARCH%20y%20Redes%20Neuronales_%20Modelos%20para%20Pronosticar%20Series%20de%20Tiempos%20Financieras.pdf. [Último acceso: 01 02 2001].
- [43] D. F. H. Cofre, «hdl.handle,» expeditiorepositorio, Bogota, 2020.

- [44] J. Villavicencio, «Introducción a Series de Tiempo,» Fondo de Cultura Económica, Puerto Rico, 2010.
- [45] R. Albeteri, «Las redes neuronales artificiales,» NETBIBLO, S. L., La Coruña, 2008.
- [46] M. Lindberg, «QoS Provisioning in Channel Allocation Strategy in Cognitive Radio Using Fuzzy Logic,» *American Journal of Electrical Systems*, vol. 2, nº 1, pp. 11-22, 2012.
- [47] H. J. E. B. J. A. F. C. Ernesto Cadena Muñoz, «Gestión del espectro radioeléctrico en Colombia,» *Tecnura*, vol. 19, nº 45, pp. 159-173, 2015.
- [48] CENTRO ANDALUZ, «CENTRO ANDALUZ,» 19 Julio 2018. [En línea]. Available: <https://centroandaluz.net/blog-academia/clasificacion-ondas-segun-propagacion-alcances-propagacion-bandas/>. [Último acceso: 07 Diciembre 2021].
- [49] L. Leon, «Ruido e interferencia,» 17 Mayo 2015. [En línea]. Available: <http://ruidoeinterferencia.blogspot.com/2015/05/naturaleza-y-causas-de-las.html>. [Último acceso: 07 Diciembre 2021].
- [50] V. Marsh, «La Comisión Europea regula el espectro radioeléctrico resultante del dividendo digital,» TENDENCIAS, Barcelona, 2010.
- [51] R. C. Álvarez, «TECNOLOGÍAS DE ACCESO DINÁMICO Y USO COMPARTIDO DEL ESPECTRO,» Instituto Federal de Telecomunicaciones, México, 2018.
- [52] D. Mezzeti, «Series de Tiempo,» Madrid, 2019.

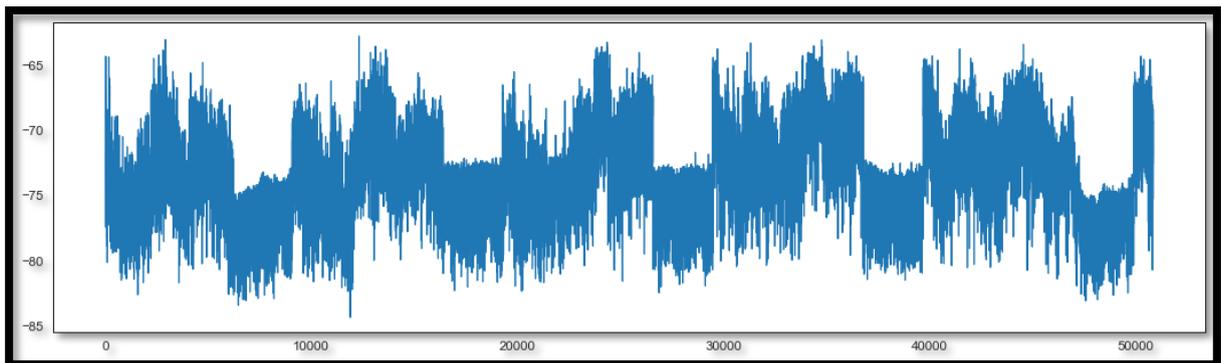
ANEXOS



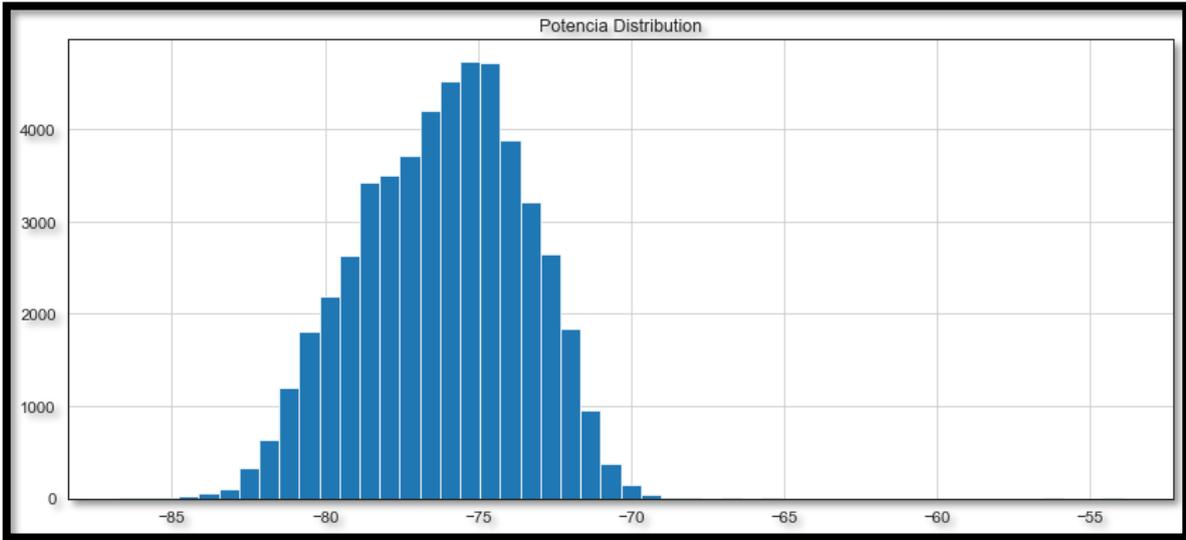
Anexo 1: Datos del canal 1



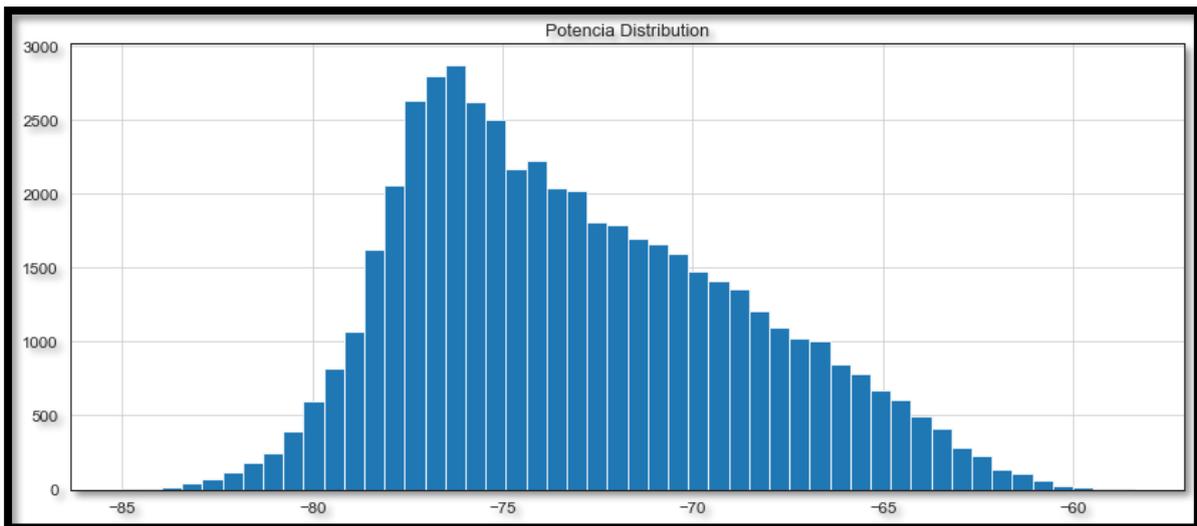
Anexo 2: Datos del canal 2



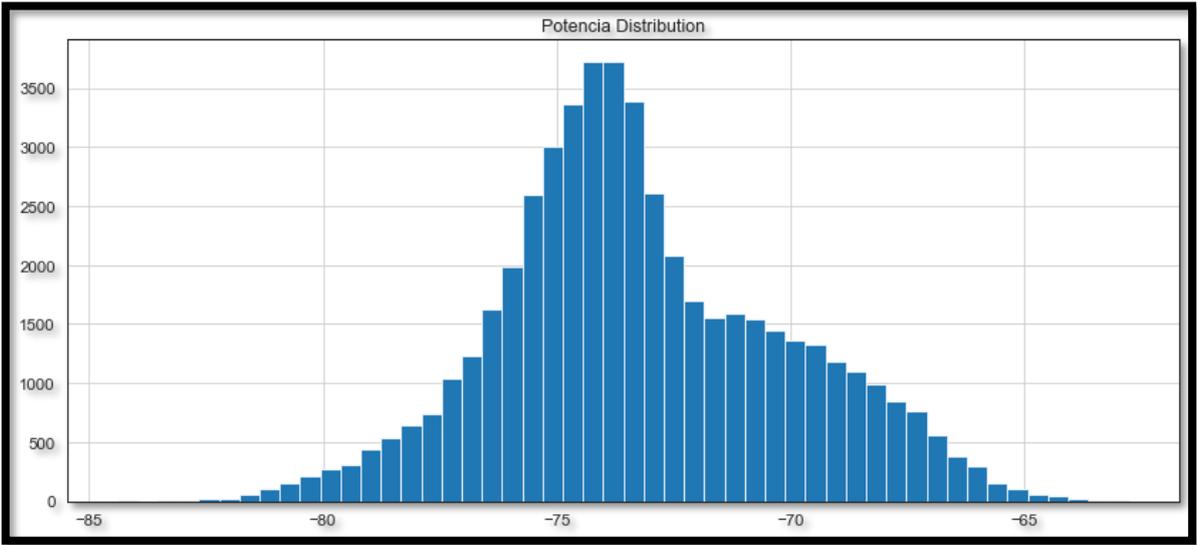
Anexo 3: Datos del canal 3



Anexo 4: Potencia distribuida del canal 1



Anexo 5: Potencia distribuida del canal 2



Anexo 6: Potencia distribuida del canal 3