

Modelo de decisión espectral para redes de radio cognitiva

Modelo de decisión espectral para redes de radio cognitiva

Cristian Camilo Bernal Ariza
César Augusto Hernández Suárez



Dedicatoria

Este libro está dedicado a Dios por haberme permitido llegar hasta este punto llenándome de fuerza para lograr mis objetivos, a mi familia y mi novia por sus consejos y motivación, gracias a su comprensión se ha logrado alcanzar esta meta.

Cristian Camilo Bernal Ariza

Agradecimientos

A mi familia por llenarme de paciencia y voluntad para afrontar los diferentes problemas presentados en este logro alcanzado.

Cristian Camilo Bernal Ariza

Financiamiento

El presente libro de investigación es producto de los resultados alcanzados en el proyecto de investigación sin financiación: “Modelo inteligente de decisión de espectro para mejorar el desempeño en redes de radio cognitiva”, institucionalizado en el 2017 con el código 3207257617, por la Facultad Tecnológica y el Centro de Investigaciones y Desarrollo Científico de la Universidad Distrital Francisco José de Caldas.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

UD
Editorial

E2
ESPACIOS

© Universidad Distrital Francisco José de Caldas
© Centro de Investigaciones y Desarrollo Científico
© Cristian Camilo Bernal Ariza, César Augusto Hernández Suárez
Primera edición, noviembre de 2019
ISBN: 978-958-787-152-4

Dirección Sección de Publicaciones
Rubén Eliécer Carvajalino C.

Coordinación editorial
Nathalie De la Cuadra N.

Corrección de estilo
Karen Grisales

Diagramación y montaje de cubierta
Fabién Camargo Muñoz.

Editorial UD
Universidad Distrital Francisco José de Caldas
Carrera 24 n.º 34-37
Teléfono: 3239300 ext. 6202
Correo electrónico: publicaciones@udistrital.edu.co

Hernández Suárez, César Augusto
Modelo de decisión espectral para redes de radio cognitiva /
César Augusto Hernández Suárez, Cristian Camilo Bernal
Ariza. -- Bogotá: Universidad Distrital Francisco José de
Caldas, 2019.
122 páginas; 24 cm.
ISBN 978-958-787-152-4
1. Ingeniería de sistemas 2. Redes inalámbricas 3. Sistemas
de transmisión de datos 4. Redes de radio cognitiva I. Bernal
Ariza Cristian Camilo, autor II. Tít.
004.6 cd 22 ed.
A

CEP-Banco de la República-Biblioteca Luis Ángel Arango

Todos los derechos reservados.
Esta obra no puede ser reproducida sin el permiso previo escrito de la
Sección de Publicaciones de la Universidad Distrital.
Hecho en Colombia

Contenido

INTRODUCCIÓN	21
Objetivos de la investigación	22
Financiamiento del proyecto de investigación	22
Organización del libro de investigación	22
RADIO COGNITIVA	23
Ciclo cognitivo	24
Decisión de espectro en redes de radio cognitiva	26
Caracterización del espectro	26
Métodos de acceso oportunista al espectro	29
Técnicas de inteligencia computacional para radio cognitiva	30
Algoritmo GRA (Grey Relational Analysis)	36
Técnicas de predicción en redes de radio cognitiva	36
Redes neuronales	37
Soporte de máquina vectorial (SVM)	38
K-algoritmo de vecinos más cercanos (KNN)	38
Trabajos relacionados	39
METODOLOGÍA	41
Modelo conceptual	41
Modelo inteligente de decisión de espectro propuesto	43
Procesamiento de los datos	44
Caracterización espectral	46

Ocupación de espectral	46
Ciclo de trabajo	46
Índice de potencia	47
SNR	48
Diseño de la etapa de clasificación	48
Construcción del algoritmo GRA	49
Construcción de la matriz de juicios	50
Cálculo de los pesos normalizados	50
Diseño de la etapa de predicción	51
Implementación de redes neuronales	51
Implementación de SVM	52
Implementación de KNN	53
Modelo inteligente de decisión de espectro definitivo	54
Metodología de validación y evaluación	55
Métricas de evaluación	55
Descripción de los experimentos	57
Simulador	61
RESULTADOS	65
Algoritmo GRA	65
Parámetros elegidos como criterios	65
Tabla de criterios	66
Pesos normalizados	67
Análisis relacional gris	68
Métricas de evaluación	69
Evaluación comparativa para la selección de canal	70
Cálculo del error de predicción	72
Ánalysis del costo computacional	72
Evaluación comparativa para la predicción	74
Ánalysis estadístico de los resultados	81
CONCLUSIONES	83
Trabajo futuro	84

REFERENCIAS BIBLIOGRÁFICAS	85
ANEXOS	91
A. Algoritmo general de decisión de espectro	91
B. Cálculo de los parámetros	96
C. Algoritmo GRA	98
D. Algoritmo de predicción	101
E. Simulador	104

Lista de figuras

Figura 1. Arquitectura de red de radio cognitiva	24
Figura 2. Ciclo cognitivo	25
Figura 3. Caracterización del ambiente de radio	28
Figura 4. Composición de una red neuronal	38
Figura 5. Modelo conceptual de decisión de espectro	42
Figura 6. Modelo de decisión de espectro	43
Figura 7. Información de los niveles de potencia obtenidos en la medición	45
Figura 8. Información del tráfico espectral	45
Figura 9. Ciclo de trabajo para la banda GSM	47
Figura 10. Índice de potencia por canal para la banda GSM	48
Figura 11. SNR para la banda 824 MHz - 874 MHz	49
Figura 12. Sistema no lineal auto regresivo en redes neuronales	52
Figura 13. Proceso de predicción SVM	53
Figura 14. Proceso de predicción KNN	54
Figura 15. Modelo de decisión de espectro definitivo	55
Figura 16. Diagrama de flujo del experimento 1	59
Figura 17. Diagrama de flujo del experimento 2	61
Figura 18. Construcción de matrices del simulador	62
Figura 19. Diagrama de flujo del simulador	63
Figura 20. Handoff totales tráfico bajo experimento 1	70
Figura 21. Handoff totales tráfico alto experimento 1	71
Figura 22. Promedio de handoff experimento 1	71
Figura 23. Handoff totales experimento 2	74
Figura 24. Handoff totales	78
Figura 25. Handoff anticipados	78
Figura 26. Handoff fallidos	79
Figura 27. Handoff perfectos	79
Figura 28. Handoff sin interferencia	80
Figura 29. Handoff con interferencia	80

Lista de tablas

Tabla 1. Comparativa de técnicas de inteligencia computacional	31
Tabla 2. Parámetros técnicos de medición del espectro	44
Tabla 3. Normalización de los parámetros para los 10 primeros canales	66
Tabla 4. Criterios para establecer pesos en el método GRA	67
Tabla 5. Pesos normalizados de los criterios	67
Tabla 6. Matriz ponderada normalizada	68
Tabla 7. Distancia correlacionada gris GRC tráfico bajo	69
Tabla 8. Error relativo promedio para tráfico bajo y tráfico alto	72
Tabla 9. Tiempo de ejecución del cálculo de los parámetros	73
Tabla 10. Tiempo de ejecución de la etapa de clasificación	73
Tabla 11. Tiempo de ejecución de la etapa de predicción	73
Tabla 12. Especificaciones técnicas del equipo de cómputo	73
Tabla 13. Handoff totales experimento 2	75
Tabla 14. Porcentaje comparativo de handoff en tráfico alto	81
Tabla 15. Porcentaje comparativo de handoff en tráfico bajo	82

Lista de abreviaturas

Abreviatura	Término
VER	Bit Error Rate
CR	Cognitive Radio
CRN	Cognitive Radio Networks
GRA	Grey Relational Analysis
GSM	Global System Mobile
KNN	K-Nearest Neighbors
PU	Primary User
RF	Radio Frequency
SNR	Signal to noise ratio
SU	Secondary User
SVM	Support Vector Machine

Introducción

En los últimos años, las redes inalámbricas han sido de gran interés para la investigación debido al crecimiento de las tecnologías que utilizan el espectro para comunicarse. La demanda de usuarios y la evolución de las tecnologías generan indirectamente escasez de bandas de frecuencia haciendo cada vez más compleja la asignación de espectro [1].

En la actualidad, los sistemas de comunicación móvil presentan deficiencias en los servicios de voz y datos debido a la saturación de las bandas de frecuencia asignadas y a la escasez de nuevas bandas de frecuencia en el espectro. La saturación de las bandas de frecuencia es ocasionada por el aumento considerable de dispositivos móviles en la red de radio. Según estudios, se espera que el tráfico IP aumente en 168 exabytes para el 2019 con un número de dispositivos móviles que será tres veces la población mundial [2].

Por este motivo, el acceso oportunista al espectro busca aprovechar aquellas bandas de frecuencia licenciadas que no sean utilizadas eficientemente, para hacer uso de ellas sin generar impactos negativos en los usuarios licenciados, y así imponer un desafío en el estudio de la utilización eficiente del espectro.

Debido a estas razones, la radio cognitiva nace como una red de nueva generación donde los dispositivos son capaces de cambiar sus parámetros de transmisión en función de su interacción con el ambiente de radio para dar lugar a la negociación activa o comunicación con otros usuarios del espectro [3]. Para ello, la radio cognitiva presenta un sistema inteligente dividido en ciclos que describen el proceso cognitivo y permite tener una mejor gestión de los recursos y rendimiento de la red; el ciclo cognitivo es capaz de detectar usuarios en el espectro, tomar decisiones, moverse por otros canales de frecuencia y compartir el espectro con otros usuarios [4].

Una de las principales funciones para acceder de manera oportunista al espectro por parte de los usuarios secundarios (SU) es tomar decisiones en diferentes situaciones que no generen afectación en el tráfico de los usuarios primarios (PU) y que permitan hacer un uso eficiente de los canales disponibles [5]. Inicialmente, el SU realiza un proceso de detección que se encarga de analizar el entorno de radiofrecuencia (RF) e identifica la oportunidad espectral de cada canal, en este proceso también detecta posibles intervenciones de los PU en los canales a utilizar [6]. Cuando un SU establece una comunicación en un canal, este usuario debe continuar monitoreando el entorno de RF para poder identificar instantes futuros en los que se presente una intervención de un PU, en ese momento, el SU requiere elegir un canal próximo en el que pueda continuar con la comunicación y que sea adecuado para finalizar la misma. Basado en estas situaciones, un SU necesita establecer

comunicación en varios canales hasta llegar al receptor, y tomar decisiones que le permitan realizar una transmisión adecuada.

Por estas razones, el objetivo principal de este trabajo es diseñar un modelo inteligente de decisión de espectro que permita al SU elegir el mejor canal para transmitir, permaneciendo la mayor parte de tiempo y prediciendo posibles intervenciones del PU, que obliguen al SU a realizar un cambio a otro canal apropiado y oportuno. El modelo propuesto inicialmente realiza una etapa de clasificación que organiza por prioridad los canales menos utilizados y define el canal más adecuado para ser utilizado por el SU, luego de esto, el modelo implementa técnicas de predicción para identificar instantes en que el canal puede ser intervenido por un PU y, de esta manera, elegir otro canal para continuar la transmisión de un servicio específico sin generar interferencia al PU.

Objetivos de la investigación

El objetivo general de esta investigación es:

Desarrollar un modelo de predicción de la ocupación espectral en un entorno urbano que sirva como herramienta para el análisis y diseño de redes de radio cognitiva.

El cual se pretende alcanzar a partir de los siguientes objetivos específicos:

- Analizar estadísticamente las tendencias de ocupación espectral con base en mediciones de ocupación espectral, realizadas en redes móviles para un entorno urbano.
- Diseñar un modelo de predicción de la ocupación espectral con base en mediciones de ocupación espectral realizadas.
- Evaluar y validar el modelo desarrollado a través de datos de ocupación espectral reales.

Financiamiento del proyecto de investigación

El presente libro de investigación es producto de los resultados alcanzados en el proyecto de investigación sin financiación denominado: “Modelo inteligente de decisión de espectro para mejorar el desempeño en redes de radio cognitiva”, institucionalizado por el Consejo de Facultad de la Facultad Tecnológica y el Centro de Investigaciones y Desarrollo Científico, de la Universidad Distrital Francisco José de Caldas, en el 2017.

Organización del libro de investigación

El libro está estructurado como sigue. En el primer capítulo se realiza una descripción de los fundamentos teóricos de la CR. En el segundo apartado se describe la metodología de la investigación realizada. La tercera sección presenta los resultados alcanzados en la investigación. Y finalmente, en el cuarto capítulo se presentan las conclusiones.

Radio cognitiva

El espectro electromagnético es un recurso natural, por el cual dispositivos inalámbricos transmiten y reciben información en diferentes rangos de frecuencia, los cuales son asignados y controlados por una entidad del gobierno de cada país [1].

Existen bandas de frecuencia que son asignadas para ser utilizadas por diferentes tecnologías que con el pasar del tiempo han sido relevadas o que pasan largos períodos sin ser utilizadas. Esta problemática ha venido creciendo debido a la evolución y aumento de la demanda de las tecnologías inalámbricas.

Debido a esta problemática, los investigadores se han enfocado en idear un software de radio que pueda mejorar la flexibilidad del espectro a través de un lenguaje de representación de radio basado en el conocimiento y razonamiento automatizado que cumpla con las necesidades del usuario [7]. Esta propuesta inicial fue ideada por Mitola, quien dio pautas para una profunda investigación y nuevas problemáticas como el acceso al espectro el cual resulta aún más importante que la misma escasez física del espectro [8].

Así pues, la CR está definida como un sistema que puede cambiar sus parámetros de transmisión en función de su interacción para dar lugar a la negociación activa o comunicación con otros usuarios del espectro [1].

Esta tecnología debe cumplir ciertas condiciones que le permitirán desempeñarse en una red de radio frecuencia, entre ellas están: determinar las partes libres en el espectro, seleccionar el mejor canal disponible, coordinar el acceso al canal con otros usuarios, desalojar el canal cuando se detecta un SU [4]. Estas características se analizarán con más detalle, más adelante.

Para lograr estas condiciones, la CR está fundamentada por dos características en su operación, una de ellas es la capacidad cognitiva, la cual es la facultad de interactuar en tiempo real con el entorno de radio, identificando los posibles agujeros en el espectro, haciendo uso de ellos de manera temporal y oportunista, y, asimismo, detectando usuarios con licencia que puedan intervenir en la comunicación [4]. La reconfiguración es otra característica importante ya que le permite programarse para transmitir y recibir información en varias bandas de frecuencia seleccionadas, y así realizar la configuración y ajuste de los parámetros de funcionamiento ideales para la comunicación como lo son las tecnologías de hardware de RF, antena de banda ancha, amplificador de potencia y el filtro adaptativo [4].

Una manera de catalogar la CR es en el tipo de arquitectura, centralizada o distribuida. Las redes clásicas o centralizadas son identificadas porque tienen en su infraestructura un control central que administra la comunicación entre usuarios como

por ejemplo las estaciones bases de redes de telefonía móvil o un punto de acceso en redes inalámbricas de área local WLAN. Por otro lado, las redes CRAHN o redes distribuidas se diferencian en que no tienen una columna vertebral en su infraestructura, es decir, que un usuario mediante CR puede conectarse en una arquitectura ad hoc con otro usuario en ambas bandas del espectro, con o sin licencia y sin una coordinación central [9].

En la figura 1, se puede observar al costado izquierdo una red centralizada en la que todos los dispositivos se conectan directamente a un nodo central de comunicación, quien es el encargado de administrar las bandas de frecuencia y controlar el acceso al medio. Al costado derecho, se observa una red distribuida o *ad hoc*, donde los dispositivos establecen una comunicación directa entre ellos sin necesitar otro dispositivo de comunicación. En esta tecnología los dispositivos observan el ambiente de RF local y toman decisiones coordinadas con sus vecinos de manera cooperativa [10].

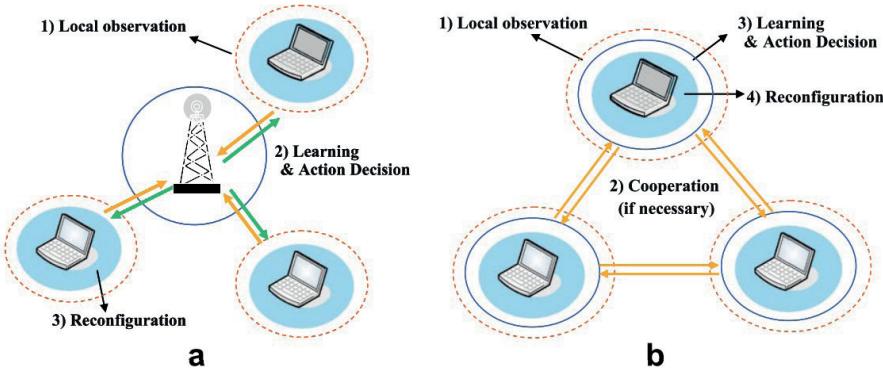


Fig. 2. Comparison between CR capabilities for: (a) infrastructure-based CR networks, and (b) CRAHNs.

Figura 1. Arquitectura de red de radio cognitiva

Fuente: tomada de [9].

Las interacciones en diferentes redes de radio cognitiva (CRN) son realizadas primero por los PU, quienes tienen un canal y un ancho de banda específico asignado por su proveedor de servicio, y en segundo lugar por los SU, que no tienen un canal determinado por lo que pretenden acceder de manera oportunista a canales con licencia sin generar interrupciones, ni afectar los servicios de los PU [11]. El SU debe actuar de manera inteligente al momento de acceder a un canal específico, para ello se debe estudiar el ciclo cognitivo.

Ciclo cognitivo

La CR presenta un sistema inteligente dividido en ciclos que describen el proceso cognitivo que debe tener un SU para mejorar la gestión de los recursos y el rendimiento de la red, al momento de presentarse diferentes situaciones y lograr mantener una comunicación estable. El ciclo cognitivo está compuesto por las funciones:

detección, decisión, movilidad y compartición del espectro. El diagrama del ciclo se muestra en la figura 2.

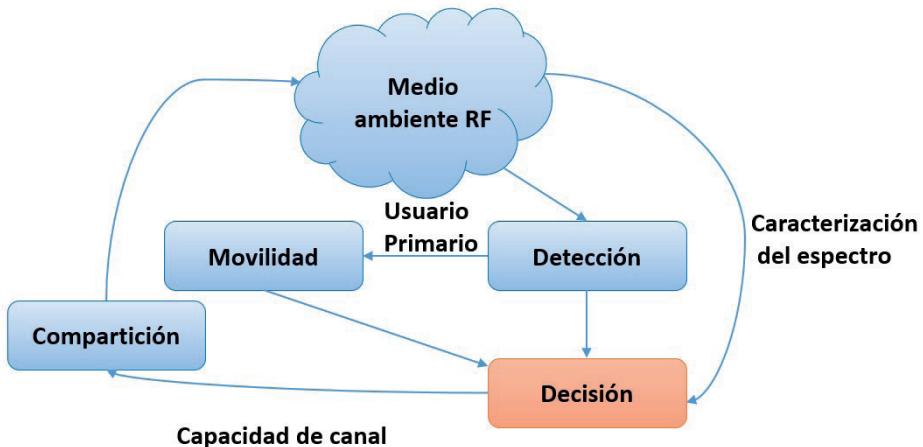


Figura 2. Ciclo cognitivo

Fuente: adaptada de [3].

Para describir la figura 2, se presentarán dos situaciones en la interacción de los SU en la red, enfocando el proceso en la decisión de espectro.

El SU inicia una transmisión: cuando nace el requerimiento de transmitir, el SU debe analizar el ambiente RF y localizar una porción no usada del espectro. La detección consiste en monitorear las bandas disponibles del espectro, capturar información y detectar agujeros en el espectro [3]. Una vez detectado el canal, se requiere tomar decisiones y elegir el mejor canal para transmitir, o el canal operativo en ese instante, es decir, que la decisión de espectro se encarga de tomar decisiones basado en la disponibilidad del espectro y en la caracterización de los canales, analizando la capacidad de ancho de banda, interferencia y retardos, entre otras características [3].

Luego de elegir el canal apropiado se debe evitar posibles colisiones con otros SU; para ello, la compartición de espectro permite que múltiples SU accedan al espectro, y evita problemas de comunicación entre ellos. La CRN puede coordinar y prevenir múltiples colisiones o solapamiento en el espectro [12]. Por último, se procede a reconfigurar los parámetros y realizar la transmisión.

El SU transmite: al momento de transmitir, el SU debe tener la capacidad de detectar la posible intervención de un PU, lo que genera la necesidad de cambiar de un canal a otro para continuar la comunicación, esta acción es denominada movilidad. El sistema debe estar en la capacidad de detectar otros canales disponibles y tomar decisiones de manera eficaz y continuar la comunicación en otra porción del espectro [13].

Decisión de espectro en redes de radio cognitiva

La decisión de espectro es una de las principales funciones dentro del ciclo cognitivo, la cual tiene la finalidad de elegir el mejor canal para transmitir, dependiendo de los agujeros en el espectro y cambios de canal que sea necesarios realizar por parte del SU cuando se presente una intervención de un PU.

Existen varias situaciones generadas en la interacción de los SU en una CRN, donde es esencial decidir de manera oportuna un canal apropiado. La principal decisión inicia cuando el SU requiere realizar una transmisión, en ese momento se hace fundamental determinar el mejor canal a utilizar que resulta de un estudio previo de la banda de frecuencia. Otra situación, y quizás más importante, es la intervención del PU cuando el SU está realizando una transmisión, en ese momento el SU debe seleccionar de manera eficaz otro canal disponible para completar la transmisión, este proceso es llamado handoff espectral [14].

En las dos situaciones anteriores, la decisión de espectro requiere de la funcionalidad de monitorear el entorno de RF para lograr el objetivo de evitar colisiones y disminuir la interferencia entre PU y SU. Un SU debe vigilar y guardar información estadística que le permita identificar situaciones futuras y tomar decisiones acertadas, esta información adquirida se convierte en parámetros útiles que describen el comportamiento de un canal y permiten caracterizar el espectro de tal manera que el canal elegido sea el adecuado y oportuno. A continuación, se definen aquellos parámetros útiles en la caracterización de espectro.

Caracterización del espectro

La caracterización del espectro determina y describe el comportamiento de los canales, lo que permite distinguir unos de otros, de acuerdo con su tráfico, ocupación y configuración. Dentro de las características de transmisión en un canal, existen parámetros que influyen en su comportamiento y es importante tenerlos en cuenta a la hora de seleccionar un canal. Por tal motivo, se deben estudiar estos factores con el fin de identificar algunos beneficios que permitan obtener una mejora en el desempeño de estas redes. Algunas características son [3]:

- Identificar el canal
- Capacidad del canal
- Retardo de conmutación de espectro
- Interferencia del canal
- Canal de tiempo (CHT)
- Tasa de error del canal
- Posición del abonado

Identificar el canal

La utilización del canal por parte del PU es quizás el factor más importante, ya que definirá los espacios y tiempos libres en un canal para ser ocupados por un SU. Esta ocupación se realiza de manera aleatoria, donde los tiempos de utilización no son determinísticos, sino impredecibles y varían en diferentes aplicaciones. Esta actividad de ocupación del canal puede ser modelada como un proceso estocástico aplicando técnicas de inteligencia artificial como redes neuronales, modelos de Markov y Support Vector Machine (SVM) [15].

Capacidad del canal

Cada banda del espectro en un sistema de multiplexación por división de frecuencia ortogonal (OFDM), tiene un ancho de banda diferente, el cual está compuesto por varias subportadoras que estiman una capacidad normalizada del canal. Las investigaciones se han enfocado en estimar la capacidad de ancho de banda mediante el estudio de otros parámetros como el nivel de interferencia, tasa de errores y propagación [16].

Se ha demostrado que el método tradicional de la estimación de la capacidad de canal, utilizando la relación señal a ruido (SNR), conduce a una decisión de espectro no óptima [3].

Las capacidades de ancho de banda de un canal licenciado están limitadas por la entidad reguladora, esto hace que el SU requiera analizar recursos cuando el ancho de banda del PU sea menor que el requerido por el SU, para el modelo propuesto, el ancho de banda no es utilizado.

Retardo de conmutación de espectro

Esta característica nace como consecuencia de la intervención de un PU cuando está operando el SU en un canal licenciado, en ese momento, el SU debe detectar nuevos canales con diferentes frecuencias y conmutar reconfigurando sus parámetros de transmisión. En ese proceso de conmutación hay una duración considerable que afecta el desempeño de las CRN [17].

El desafío presentado está en reducir el retardo de detección del canal, el tiempo que tarda el SU en configurar sus parámetros de transmisión y disminuir el tiempo que gasta el SU en acceder al nuevo canal, con el fin de mejorar el rendimiento de retardo en CRN.

Interferencia del canal

La interferencia es la mayor consecuencia generada en el proceso de interacción entre usuarios en una red. Al acceder al espectro, un SU puede afectar la señal y alterar los servicios del PU, por lo tanto, la interferencia está definida como la perturbación de la señal debido a la coexistencia entre PU y SU en un área de cobertura del PU [18].

Existen diferentes estudios para evitar interferencias entre PU y SU en áreas de

cobertura específica, donde es importante que el SU no transmita mientras exista presencia de un PU.

Tiempo de retención de un canal (CHT)

Esta característica es importante al momento de modelar el acceso al canal, debido a que estudia los tiempos de activación y de inactividad de los PU y SU y permite el acceso a canales ranurados, regulando y sincronizando este acceso [19].

Inicialmente, se estudian los tiempos de duración de activación de los usuarios y tiempos en que el canal permanece libre, luego se definen bloques de tiempo que serán recursos libres que podrían ser utilizados por los SU con bloques de similar tamaño, para finalmente, acceder al canal que más se ajuste con las mediciones realizadas, lo que reduce interferencias en la interacción [19].

Tasa de error de canal

Este factor está directamente relacionado con el nivel de interferencia, ancho de banda y la banda de frecuencia disponible, los cuales influyen directamente en la recepción o transmisión de errores de bit en un canal. Esta tasa de error de bit (BER) es indicada con la SNR [20].

Localización del abonado

El SU dentro de su funcionamiento y proceso de detección debe obtener información geográfica y del ambiente de RF, mediante una función del sistema de posicionamiento global (GPS), para coordinar información entre los nodos o servidores centrales que identifiquen la ubicación de cada SU y poder construir un mapa de actualización mundial. Este proceso permitirá predecir situaciones futuras de intervenciones de los PU [21].

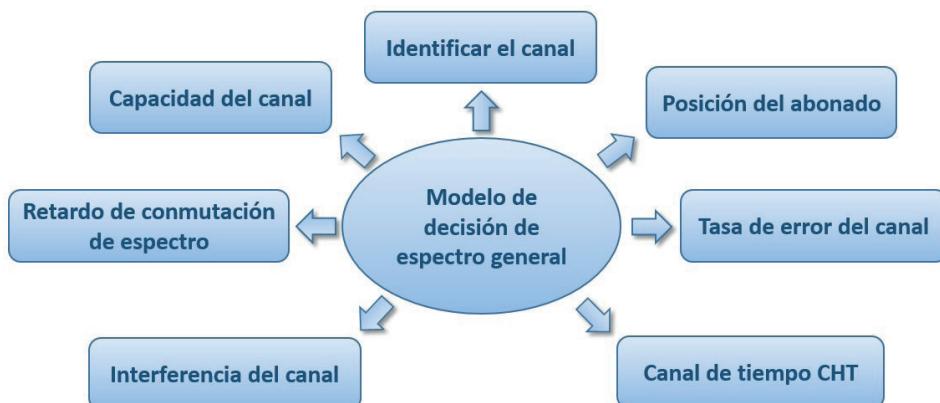


Figura 3. Caracterización del ambiente de radio

Fuente: adaptada de [3].

La figura 3 presenta en síntesis los componentes que debe tener en cuenta un modelo de decisión de espectro para cumplir con el objetivo de acceder al espectro oportunamente. Aunque diseñar un modelo que evalúe todas las características puede ser robusto, complejo y computacionalmente poco eficaz, si es misión del ingeniero idear un modelo que utilice las características principales y necesarias, y que también reduzca el tiempo de ejecución del mismo.

Dentro del modelo de decisión de espectro es importante definir la manera en que se accede al espectro dependiendo de las características evaluadas en la caracterización. A continuación se estudiarán brevemente los métodos de acceso oportuno al espectro.

Métodos de acceso oportuno al espectro

Los SU implementan diferentes técnicas para acceder al espectro basado en la premisa de no afectar el tráfico de los PU; con tal fin, se han ideado diferentes maneras de acceder al espectro y se presentan ventajas y desventajas relacionadas con los retardos ocasionados en el proceso de selección del canal, reconfiguración y la generación de interferencia al ingresar a un canal ocupado.

Acceso reactivo

El acceso reactivo es el método por el cual los SU actúan de acuerdo a las intervenciones imprevistas de los PU, sin realizar una evaluación previa del comportamiento del canal, ni predecir agujeros en el espectro relacionados con la actividad del PU.

Este método puede generar ventajas en la disminución del retardo al momento de comutar a otro canal, pero a su vez, presenta el riesgo de generar mayor interferencia en otros canales.

Acceso predictivo

Los modelos predictivos asumen modelos de ocupación del canal en bandas de frecuencia licenciadas y no licenciadas basado en información recolectada del pasado. Estos conjuntos de datos estadísticos sirven de base para la implementación de técnicas de inteligencia artificial que permitan predecir estados futuros de acuerdo a sus estados anteriores, creando un patrón de uso de un canal específico. Dentro de las técnicas mayormente implementadas se encuentran los modelos ocultos de Markov, técnicas de regresión y redes neuronales. El trabajo realizado en [22] permite el acceso al espectro prediciendo el tráfico del PU mediante un modelo oculto de Markov.

El SU, dentro de su funcionamiento, podrá implementar alguna técnica para decidir en qué momento puede cambiar de canal y acceder a otro, de acuerdo con sus resultados de predicción. Este método tiene la ventaja de evitar la interferencia con otros PU, siempre y cuando las predicciones futuras sean acertadas.

Acceso proactivo

Este sistema de acceso oportunista es el más apropiado en la implementación de las CRN, debido a que el SU prevé con anticipación la intervención de un PU, lo que le permite tomar acciones previstas y rápidas que evitarán generar obstrucciones y colisiones entre usuarios, reaccionando anticipadamente para realizar un cambio a otro canal y continuar la comunicación. En [23] podemos observar un modelo de acceso proactivo del espectro.

Los esquemas de cambio de canal proactivos logran que un SU pueda cambiar de canal, ya sea cuando se anticipa a la llegada de un PU o cuando un canal con un mayor tiempo de inactividad puede ser ocupado basándose en la predicción. Es claro que este método de acceso oportunista es beneficioso en la implementación debido a que disminuye los riesgos de generar interferencia y logra aprovechar el canal desocupado de una manera eficiente.

Técnicas de inteligencia computacional para radio cognitiva

Las técnicas de inteligencia computacional brindan soluciones en el diseño de redes inalámbricas y en especial en las CRN, donde los métodos de clasificación y aprendizaje son importantes para brindar soluciones en el funcionamiento y mejoramiento de la prestación de servicios en una red inalámbrica.

En una CRN es indispensable utilizar herramientas que permitan tomar decisiones acordes a la variación del comportamiento del PU, tanto en el dominio del tiempo como del espacio, por lo que se hace necesario realizar un estudio previo de los métodos utilizados para clasificar canales y predecir espacios en el espectro.

En la construcción de este trabajo, inicialmente se realizó un estudio del estado del arte de las técnicas de inteligencia artificial para identificar los métodos más utilizados en la actualidad. El resultado de este estudio se ve representado en la elaboración de la tabla 1, donde se evalúan diferentes algoritmos, se determina su aplicación, se identifican sus fortalezas y limitaciones, e incluso en algunos de ellos, se informa la capacidad de predicción y su rendimiento.

La tabla 1 también presenta la discusión y el análisis de los resultados obtenidos al realizar la comparación entre las diferentes técnicas, que permitirá definir cuáles pueden servir de apoyo en el diseño del modelo propuesto. En la primera mitad del cuadro se presentan los principales algoritmos de toma de decisiones basados en el método MADM (Multiple attribute decision making), y en la otra mitad se muestran los algoritmos usados en la predicción.

SAW

La aplicación de este algoritmo está fundamentada en la toma de decisiones y en la evaluación de criterios múltiples. Este método es utilizado para resolver problemas de decisión y se caracteriza por elegir la mejor alternativa basada en atributos que describen dichas alternativas. Dentro de las fortalezas más importantes, se puede considerar que este método permite asignar diferentes tipos de datos, tanto cuantita-

tivos como cualitativos. Una limitación importante para este método es la construcción de matrices de decisión y la normalización de los datos, lo que genera posibles retardos en su ejecución que impedirán evaluar el procesamiento y tomar decisiones en tiempo real [24].

VIKOR

Este algoritmo es implementado para la toma de decisiones multicriterio. VIKOR es un algoritmo de decisión que elige la alternativa que más se ajusta a la alternativa ideal, alcanzando un rendimiento medio en la asignación de varias características como por ejemplo: costo, ancho de banda, frecuencia, entre otras [24]. Al igual que SAW este método podría tener problemas en la ejecución en tiempo real si la base de datos es de gran tamaño.

Tabla 1. Comparativa de técnicas de inteligencia computacional

Técnica de aprendizaje	Aplicación	Fortalezas	Limitación
SAW		Se pueden asignar diferentes tipos de datos que caracterizan un canal, pueden ser cuantitativos o cualitativos.	
VIKOR	Toma de decisiones con evaluación de criterios múltiples	VIKOR es un algoritmo de decisión que elige la alternativa que se ajusta a la ideal, alcanza un rendimiento medio en la asignación de varias características como costo, ancho de banda y frecuencia.	Requieren de una matriz de decisión y normalización de los datos. Pendiente evaluar el procesamiento en tiempo real.
TOPSIS		Igual que SAW es un algoritmo de decisión que trabaja con atributos cuantitativos. Elige la alternativa más adecuada con la distancia euclíadiana más corta.	

Técnica de aprendizaje	Aplicación	Fortalezas	Limitación
GRA	Clasificar redes	Permite organizar las redes por prioridad según el resultado del algoritmo M-AHP.	No reduce la cantidad de handoff en la toma de decisiones.
	Clasificar redes candidatas	Es eficiente en el proceso de clasificación.	Depende del algoritmo FAHP para determinar los pesos de los criterios objetivos y subjetivos. No es diseñado para predecir la actividad de los PU.
Aprendizaje por refuerzo RL	Aprendizaje de acciones para toma de decisiones óptimas y detección de canal	Es un algoritmo de adaptación que aprende de su experiencia interactiva y crea una política óptima que puede ser utilizada para la toma de decisiones y es aplicado para: asignación de canal, control de la congestión y minimización de la interferencia.	No utiliza la experiencia pasada para el aprendizaje, converge a una política óptima luego de un centenar de episodios o etapas de entrenamiento. Es asincrónico, no especifica qué acción tomar en determinado tiempo. No actúa en tiempo real.
POMDP Proceso de decisión de Markov	Modela estados de un canal discreto en el tiempo	Predicción de la ocupación del espectro. Predice la ocurrencia y duración de agujeros en el espectro.	Alto nivel de computación, e interpretación del modelo matemático.

Técnica de aprendizaje	Aplicación	Fortalezas	Limitación
Algoritmos genéticos	Reconfiguración de antenas, optimización, detección de espectro	Resuelve problemas de optimización en la reconfiguración de antenas. Define el rendimiento deseado de la antena.	No es aplicable para toma de decisiones, ni predicciones.
Lógica difusa	Toma de decisiones y handoff espectral	No hay mayor dificultad en la implementación, utiliza parámetros de entrada como los datos de caracterización del canal, es menos complejo computacionalmente y toma decisiones rápidas en tiempo real.	Necesita de otra herramienta para realizar otro tipo de función, es exclusivo para toma de decisiones.
MLP Perceptrón Multicapa ANN	Predicción actividad PU	Presenta alta precisión en la probabilidad de error y en la predicción.	Ineficaz al aumentar las redes neuronales.
Recurrente ANN	Predicción actividad PU	Está bajo investigación aplicar redes neuronales recurrentes para el modelo de predicción del mapa espacio-temporal para la probabilidad de ocupación de espectro.	
LSVM núcleo lineal y GSVM Kernel Gaussiano	Clasificador, predicción de la actividad del PU	LSVM resulta ser el mejor algoritmo para predecir el periodo OFF de un canal dentro de los algoritmos. Ofrece mayor precisión.	Complejidad en la implementación. Requiere de amplia información para el entrenamiento y actualización del sistema.

Técnica de aprendizaje	Aplicación	Fortalezas	Limitación
Teoría de juegos	Comparación de espectro en redes ad hoc, control de potencia y asignación de espectro	Puede ser aplicado para optimizar un proceso de compartición de espectro, controlar potencia para evitar interferencia e identificar usuarios malignos para realizar enrutamiento seguro.	Complejidad en el planteamiento de las estrategias y el modelado matemático.
Árbol de decisión	Toma de decisiones	Se genera con datos de entrenamiento que pueden ser las características del canal. No presenta complejidad en la implementación.	No se tienen valores estadísticos de referencia sobre la utilidad en la implementación de este algoritmo.

TOPSIS

Es otro de los algoritmos de decisión multicriterio que trabaja con atributos cuantitativos. Elige la alternativa más adecuada, de acuerdo con la distancia euclíadiana más corta.

Aprendizaje por refuerzo RL

Es una técnica de aprendizaje automático, cuyo enfoque es determinar qué acciones tomar por parte de los agentes dependiendo de su entorno, con el fin de maximizar una recompensa que puede ser utilizada para la toma de decisiones. Esta técnica es aplicada para asignación de canal, control de la congestión y minimización de la interferencia [25].

Una limitación de esta técnica de aprendizaje es que no utiliza la experiencia o datos históricos para el aprendizaje y toma de acciones. Converge a una política óptima luego de un centenar de episodios o etapas de entrenamiento, es asincrónico y no específica qué acción tomar en determinado tiempo. No actúa en tiempo real [26].

El aprendizaje por refuerzo inicia cuando un SU evalúa ciertos parámetros del entorno para construir un estado actual. Luego de producirse las acciones de todo el estado actual en su entorno, el SU, o agente, realiza el estudio de acciones y selecciona una del conjunto de acciones obtenidas para realizar la ejecución. Finalizada la ejecución, el SU calcula una recompensa que servirá como dato para las próximas acciones a tomar, actualizando un conjunto de valores [25].

Proceso de decisión de Markov

El modelo de Markov en tiempo discreto es adecuado para modelar la actividad del PU en un canal, recordemos que la actividad del PU puede ser modelada como el cambio de estados activos e inactivos durante un tiempo determinado. También puede ser descrito como un proceso estocástico en el que la generación de símbolos de observación depende de las propiedades de emisión de los estados. El modelo oculto de Markov es una máquina de estados finita, en el que la secuencia de observaciones es una función probabilística de estados [27]. La fortaleza más importante de esta técnica es la predicción de ocupación del espectro; predice la ocurrencia y duración de agujeros en el espectro. Dentro de sus limitaciones está el alto nivel de computación y su complejidad en el diseño del modelo matemático.

Algoritmos genéticos

Estos algoritmos son ideados para encontrar soluciones óptimas utilizando cierta población de individuos y sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica [28]. Es muy utilizado para resolver problemas de optimización en la reconfiguración de antenas, ya que define el rendimiento deseado de la antena. En nuestro modelo no se aplica la optimización en la toma de decisiones.

Lógica difusa

La lógica difusa es una técnica empleada en modelos en los que el tipo de información toma valores aleatorios que oscilan entre extremos de lo falso o lo verdadero. El sistema de lógica difusa se compone de fuzzificador, motor de inferencia y defuzzificador [29]. No hay mayor dificultad en la implementación. Utiliza parámetros de entrada como los datos de caracterización del canal. La lógica difusa es menos compleja en la ejecución computacional y toma decisiones rápidas en tiempo real.

Teoría de juegos

La teoría de juegos se basa en una colección de casos prácticos para desarrollar modelos matemáticos de cooperación y conflicto. Esta técnica puede ser aplicada para optimizar un proceso de compartición de espectro, controlar potencia y evitar interferencia; también puede ser usada para identificar usuarios malignos que afecten el enrutamiento seguro [30]. El diseño del modelo presenta complejidad en el planteamiento de las estrategias y el modelado matemático.

Árbol de decisión

El proceso de desarrollo de este algoritmo inicia clasificando los valores de referencia, atributos y objetivo. Luego de ello, el algoritmo calcula el valor medio de la entropía de información, calcula la ganancia de información en cada atributo y elige el atributo con mayor ganancia como el nodo inicial o raíz. El árbol de decisión es implementado con datos de entrenamiento como por ejemplo los parámetros de caracterización del canal. No presenta complejidad en la implementación [31].

Algoritmo de decisión multicriterio MADM

Los algoritmos de decisión multicriterio MADM son utilizados en sistemas donde se requiere dar una solución que depende de diversos parámetros que están en constante variación. La información obtenida de cada parámetro sirve para desarrollar juicios a la hora de tomar decisiones; las redes inalámbricas no son ajenas a estos sistemas [24].

La toma de decisiones se ha convertido en un estudio desarrollado de manera operativa donde han surgido diferentes metodologías con procesos matemáticos que ayudan a solucionar diversos problemas. Es por eso que en la CR es necesario un recurso como la decisión multicriterio para poder realizar una selección y clasificación de canales. En el funcionamiento de la decisión multicriterio es fundamental tener en cuenta la mayor cantidad de parámetros que influyen en las redes inalámbricas y los más predominantes. Para nuestro modelo propuesto, la mejor decisión está representada en elegir el canal menos utilizado de acuerdo a la experiencia pasada. Esos parámetros, también llamados criterios, son entradas para el algoritmo el cual evalúa los valores dependiendo si son convenientes en el resultado que se desea obtener.

En consecuencia, se concluye que GRA puede ser una técnica apropiada para la clasificación debido a su sencilla implementación y su poco uso en la tecnología de CR. En la etapa de predicción se eligieron las técnicas de redes neuronales: SVM y KNN, las cuales se evaluarán con el objetivo de seleccionar la más apropiada.

Algoritmo GRA (Grey Relational Analysis)

El algoritmo GRA es también llamado modelo de análisis de la incidencia gris, fue desarrollado por el profesor chino Julong Deng de la universidad de Ciencia y Tecnología de Huazhong. Este modelo utiliza un concepto específico de información que puede operar con datos cuantitativos y cualitativos.

La principal función de este algoritmo es clasificar, para permitir la organización de los datos por prioridad. Dentro de sus aplicaciones, el algoritmo puede elegir la mejor interfaz según los atributos de la red como retardo, ancho de banda y costo, con el objetivo de reducir el handoff y mejorar la satisfacción del cliente.

En el desarrollo de la metodología se expondrá el funcionamiento e implementación del algoritmo GRA.

Técnicas de predicción en redes de radio cognitiva

Esta sección se alejará un poco de la temática de la decisión de espectro, pero contribuirá a definir las técnicas de inteligencia artificial que se pueden aplicar para el diseño del modelo de decisión de espectro según el enfoque y el objetivo propuesto en el proyecto.

La inteligencia artificial tiene como objetivo hacer máquinas que realicen tareas mediante la formulación matemática, y que sean capaces de solucionar problemas,

deducir, razonar, representar conocimiento y aprendizaje, dichas tareas pueden ser aplicadas en una CRN.

Estas técnicas son aplicadas en algunas áreas del ciclo cognitivo en donde se presentan diferentes desafíos, como por ejemplo la detección de espectro en bandas de frecuencia licenciadas, en donde es necesario estudiar parámetros de calidad de servicio como el ancho de banda, la SNR y BER. Otro desafío importante es mantener una observación constante del ambiente de RF, que permita identificar interacciones entre los PU y SU, ayudando a los SU a definir ciertas características que servirán de apoyo en el aprendizaje automático, de acuerdo a su experiencia pasada, y adaptar este conocimiento para mejorar la toma de decisiones [32].

En este apartado se discutirán las tres técnicas de inteligencia artificial seleccionadas: redes neuronales, SVM y KNN, que son aplicadas para solucionar problemas específicos de predicción y acceso oportunista al espectro.

Redes neuronales

Las redes neuronales son un modelo computacional que sirve para aprender y almacenar diversos modos de entrada y salida de un sistema. En la figura 4 se describe el diagrama de una red neuronal artificial dividida en tres capas: (1) capa de entrada, la cual recibe la información del exterior y la entrega a la capa media; (2) la capa media, que conserva una gran cantidad de unidades neuronales y se utiliza para el procesamiento de la información; y finalmente, (3) la capa de salida, que entrega un resultado, el cual es comparado con el resultado esperado, para identificar los errores e iniciar una transmisión hacia atrás de los mismos. Cada unidad neuronal modificará sus pesos, haciendo los errores más pequeños y la salida cada vez más cercana a la salida real deseada [33].

Las redes neuronales se utilizan para llevar a cabo el proceso de aprendizaje, observar el medio ambiente de la red y mejorar el proceso de la toma de decisiones, por medio de la identificación de nuevos patrones que ayuden a predecir las probabilidades de detección de espectro.

Estas redes neuronales tienen la capacidad de detectar el patrón de comportamiento del PU y tomar decisiones.

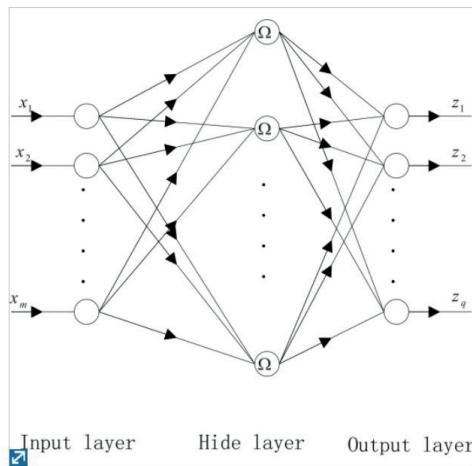


Figura 4. Composición de una red neuronal
Fuente: Tomada de [33].

Soporte de máquina vectorial (SVM)

Esta técnica la definen como un modelo de aprendizaje [34]. También existen aplicaciones donde utilizan SVM para modelar el aprendizaje de la detección del espectro en tiempo real.

Las pruebas realizadas en el modelo de clasificación SVM son las siguientes: si una señal es detectada y se genera ruido, se identificará como un PU detectado. Cuando la señal es única sin ruido, indicará que no hay PU. Los parámetros considerados en este trabajo incluyen soporte, secuencia de pulsos, la extensión de repetición y el prefijo de la circulación [34].

El SVM funciona identificando un hiperplano con vectores para cada una de las clases que se desean clasificar, luego de esto, el algoritmo maximiza las distancias entre los puntos mas cercanos a cada hiperplano definido para cada clase.

K-algoritmo de vecinos más cercanos (KNN)

KNN es un algoritmo de clasificación supervisada basado en el entrenamiento de datos históricos. Este algoritmo estima la función de probabilidad de las unidades predictores por cada clase.

Básicamente, este algoritmo funciona entrenando su base de datos, que son vectores característicos junto con las etiquetas de clase, los nuevos datos serán clasificados según su clase y almacenados en vectores. Luego de esto, se calcula la distancia entre los vectores almacenados y los nuevos, seleccionando los vecinos más cercanos. Dicha distancia euclídea está dada por la ecuación (1).

$$d(x, \bar{x}) = \sqrt{(x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2 + \dots + (x_n - \bar{x}_n)^2} \quad (1)$$

Dado un entero positivo K, una observación x y una métrica similar d, KNN puede seguir dos pasos importantes:

Se ejecutan todos los datos informáticos d entre cada observación de entrenamiento x. Luego se estima la probabilidad condicional para cada clase y la salida x se asigna a la clase de mayor probabilidad, el modelo general puede ver en la ecuación (2).

$$P(y = i | X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j) \quad (2)$$

Trabajos relacionados

En la literatura actual se encuentran investigaciones que utilizan métodos que permiten solucionar problemas de toma de decisiones para elegir la red adecuada en diferentes tecnologías de redes inalámbricas, como es el caso del trabajo realizado en [6], donde evalúan 8 algoritmos MADM, entre ellos el algoritmo GRA, el cual clasifica entre las primeras técnicas que garantizan la mejor continuidad de los servicios transportados en estas redes.

En el trabajo realizado en [35] eligen la mejor interfaz según los atributos de la red, como retardo, ancho de banda y costo, con el objetivo de reducir la tasa de handoff y mejorar la satisfacción del cliente. En [36] se destaca la utilización del algoritmo FAHP para calcular los pesos de los criterios y GRA para clasificar las redes candidatas. El trabajo realizado en [37] y [38] utiliza GRA para clasificar redes en diferentes escenarios como UMTS, WLAN y WIMAX, basándose en múltiples criterios de costos y tráfico de paquetes. GRA proporciona un rendimiento aceptable al clasificar diferentes tipos de tráfico.

En cuanto a los algoritmos de predicción, se puede observar un resumen en [39] y en [40], donde se predice la probabilidad de aparición de un PU al canal, mediante redes neuronales, las cuales aprenden el comportamiento previo del PU. Si es alta la probabilidad de aparición, el SU está obligado a cambiar de canal. Este modelo está bien ajustado con el método de error cuadrático promedio (MSE). En [41] utilizan una red neuronal para predecir los estados del canal mediante Sigma-IF y MLP (Multilayer Perceptron), los resultados demuestran que Sigma-IF presenta una mejor predicción y disminuye el tiempo de detección. En [42] se utiliza SVM para predecir el punto en el que el SU debe pasar a otro canal antes de que el canal sea ocupado, lo que reduce el tiempo de handoff. En el artículo realizado en [43] utilizan cuatro técnicas de inteligencia artificial, dentro de las cuales se encuentran las redes neuronales y SVM, donde SVM se comporta de manera consistente y precisa.

Los métodos de decisión de espectro pueden clasificarse de la siguiente manera:

- Decisión de espectro sin carga [44]: es el método donde el SU elige un canal operativo basándose en la carga de tráfico, la probabilidad de que este inactivo y el periodo de inactividad [45].
- Decisión de espectro basado en la probabilidad: este método evalúa las cargas de tráfico de los SU.
- Decisión de espectro basada en la detección: este método de decisión analiza la propuesta de diferentes canales y verifica el canal operativo más adecuado mediante la candidatura [44].

Continuando con el estudio de la literatura actual, observamos cómo aún se realizan investigaciones en la decisión de espectro donde se pretende mejorar la toma decisiones aplicadas al servicio multimedia para mejorar su calidad de servicio, como en el caso del trabajo realizado en [46], donde desarrollan un modelo que se adapta a las decisiones y condiciones del canal cambiante, para asegurar una calidad de servicio óptima.

En otros trabajos, como [47], buscan maximizar la probabilidad de detección y de esta manera lograr definir el umbral de decisión que permita adaptar los SU a los cambios del canal. En [48] maximizan el acceso dinámico al espectro basándose en el método grafo de coloración, también llamado etiqueta de grafos.

Metodología

Al inicio de este proyecto se evaluaron diferentes técnicas de inteligencia artificial mediante la construcción de un cuadro comparativo con el objetivo de identificar fortalezas y limitaciones en las técnicas implementadas y elegir los algoritmos apropiados en cuanto al desempeño y rendimiento computacional para proponer un diseño adecuado y mejorable a los demás.

Teniendo en cuenta el resultado del cuadro comparativo se propone la construcción del modelo propuesto para la decisión de espectro, compuesto de una serie de fases, iniciando con el cálculo de los parámetros que inciden en la caracterización del espectro, en segundo lugar, se realiza la implementación del algoritmo GRA, clasificando los mejores canales y finalmente, se predicen posibles intervenciones del PU, implementando tres algoritmos de predicción y eligiendo el más pertinente.

Modelo conceptual

El objetivo principal de este proyecto es mejorar el desempeño de las CRN enfocado en la toma de decisiones para la asignación de espectro. Para ello, la principal misión del SU es elegir el canal menos ocupado y evitar transmitir en canales que han sido clasificados como los menos adecuados para la comunicación.

Basado en esta premisa, el diseño del modelo conceptual presentado, describe los componentes que integran el modelo propuesto para la decisión de espectro y su relación entre sí, mediante el diseño del modelado de sistemas de software basado en UML (*Unified Modeling Language*). Dentro del modelo conceptual el principal componente que influye en la decisión de espectro es el canal de frecuencia, el cual es el principal recurso espectral en el que los PU y SU transmiten la información.

Para sacar provecho de los canales libres, el SU debe analizar ciertos parámetros que favorezcan la elección del canal menos ocupado y además, no interferir en la comunicación de otros usuarios.

En la figura 5 se observa el modelo conceptual, las clases que lo componen y sus respectivos atributos.

La primera clase es llamada *captura de datos*, es la etapa inicial del modelo, donde el SU deberá realizar el monitoreo de la red y tomar mediciones de los niveles de potencia de su entorno. Esta clase contiene los atributos de medición de las bandas de frecuencia y su relación con los niveles de potencia de cada banda.

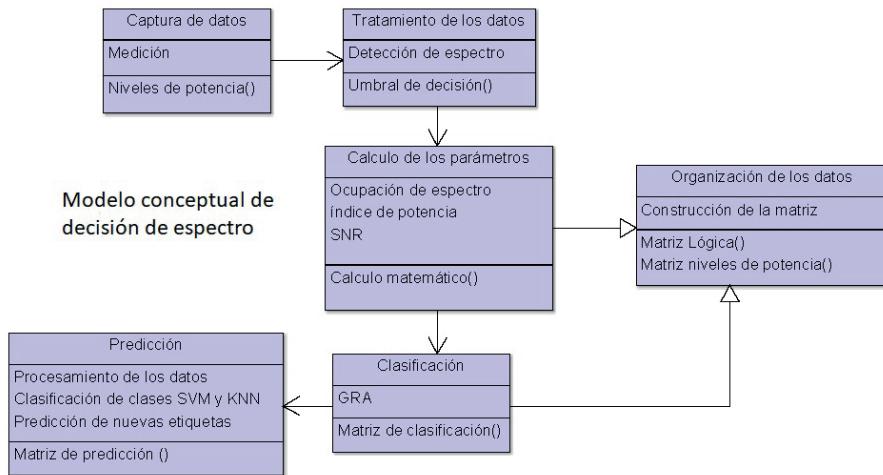


Figura 5. Modelo conceptual de decisión de espectro

La siguiente clase representa el tratamiento de los datos. En esta parte se realiza el sistema de detección de espectro donde son separados los datos e identificados los instantes de tiempo en que el canal está ocupado o desocupado.

Una vez establecida la base de datos, la cual está organizada de tal manera que las filas describirán los canales de la banda GSM y las columnas serán el tiempo de barrido de cada de cada canal, se procede a calcular los parámetros de caracterización seleccionados.

La clase cálculo de los parámetros determina los datos calculados que serán utilizados como entrada al algoritmo de clasificación. Un parámetro importante que describe el comportamiento del canal de frecuencia es la ocupación espectral, definida como la cantidad de tiempo que permanece el canal ocupado en un tiempo de medición dado. El siguiente parámetro presenta un índice de potencia obtenido de la relación del nivel de potencia máximo y mínimo en cada canal, y finalmente, el parámetro SNR indica la relación señal a ruido de cada canal de frecuencia, como se puede observar en la figura 5.

La clase clasificación reúne los resultados de los parámetros ocupación de espectro, índice de potencia y SNR e implementa el algoritmo GRA para organizar por prioridad los mejores canales.

El SU requiere de inteligencia artificial para elegir el canal y predecir las posibles intervenciones de los PU, de manera que este modelo conceptual involucra una clase llamada predicción. Por lo tanto, el modelo propuesto está conformado por un algoritmo de clasificación GRA y tres algoritmos de predicción, como lo son: redes neuronales, SVM (Support Vector Machine) y clasificador KNN (K-Nearest Neighbors algorithm). El ciclo del modelo conceptual se observa en la figura 5.

Reuniendo lo más importante, y de acuerdo con las reglas planteadas en este modelo, las funciones a realizar por un SU son: monitorear el ambiente de RF para

identificar los canales que van a ser desocupados por parte del PU, luego de ello, el SU realiza un cálculo de los parámetros que afectan el comportamiento de los canales y clasifica por prioridad los menos ocupados y más adecuados. Una vez elegidos los mejores canales, el SU debe detectar intervenciones de los PU en cada canal mediante una predicción aproximada.

Modelo inteligente de decisión de espectro propuesto

El modelo de decisión de espectro propuesto se muestra en la figura 6; en la primera parte se realiza el procesamiento de los datos obtenidos en la medición de la banda de frecuencia GSM (824 MHz - 874 MHz) y se calculan los parámetros que caracterizan dicha banda de frecuencia, estos parámetros son la ocupación de espectro, el índice de potencia y la SNR. Luego de obtener los resultados, se procede a clasificar los mejores canales para transmitir, mediante el algoritmo de decisión multicriterio GRA. Posterior a ello, el modelo organiza los mejores canales e inicia un proceso de predicción de cada uno, utilizando los datos de captura. El modelo es entrenado por medio de tres técnicas de inteligencia artificial, las cuales son: redes neuronales, SVM y KNN.

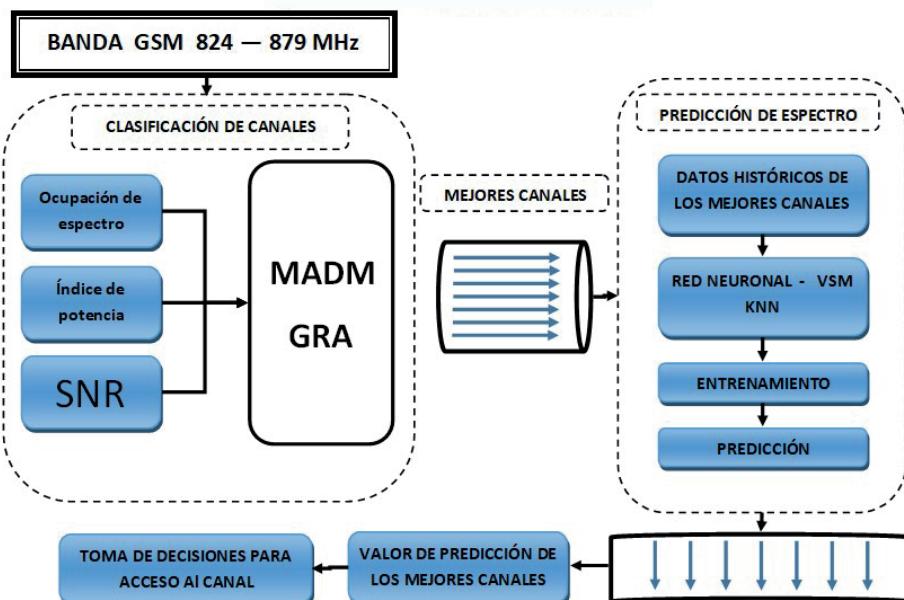


Figura 6. Modelo de decisión de espectro

Procesamiento de los datos

Para el desarrollo del modelo se utilizaron datos obtenidos en la medición realizada con un analizador de espectro, en la Facultad de Ingeniería de la Universidad Distrital Francisco José de Caldas. La tabla 2 muestra los parámetros técnicos más importantes.

Tabla 2. Parámetros técnicos de medición del espectro

Parámetros	Valor
Banda de frecuencia	824 MHz- 874 MHz
Tecnología de comunicación	GSM
Número de canales	551
Tiempo de barrido	290 ms
Resolución BW	100 KHz

La campaña de medición se realizó durante seis meses, siete días a la semana y 24 horas por día, con tiempos de barrido de 290 ms, aproximadamente 3 datos por segundo para cada canal de frecuencia dentro de la banda GSM. Posteriormente, se organizaron los datos capturados y se realizó un análisis estadístico para identificar las trazas de 70 minutos con tasa de ocupación más alta y más baja, a partir de la identificación de un umbral con la fórmula de probabilidad de falsa alarma. Los primeros 60 minutos serán utilizados para la etapa de entrenamiento equivalentes a 10800 time steps y los 10 minutos restantes son la etapa de evaluación que equivale a 1800 time steps. En la figura 7 se observan los datos obtenidos en la medición, distribuidos de tal manera, que las filas representan las ranuras de tiempo y las columnas representan cada canal.

Para determinar si los usuarios están presentes en un canal o no, se define un umbral de decisión de -95 dBm, el cual se obtuvo a partir del nivel de piso de ruido de la banda de frecuencia seleccionada y la probabilidad de falsa alarma. Si el dato en el respectivo canal está por debajo del umbral, se asigna un 0 lógico, para indicar que el canal está desocupado en ese instante de tiempo. Si, por el contrario, el dato es mayor al umbral, se asigna un 1 lógico, para indicar que el canal está ocupado. La detección de espectro es modelada en la ecuación (3).

$$y = \begin{cases} 1, & \text{si } Rx\ PU \geq -95 \text{ dBm} \\ 0, & \text{si } Rx\ PU < -95 \text{ dBm} \end{cases} \quad (3)$$

A11 : -94,484001

	C	D	E	F	G
1	-98,751999	-96,711998	-96,732002	-98,056	-96,592003
2	-97,084	-97,363998	-96,848	-99,236	-95,664001
3	-98,248001	-98,367996	-98,688004	-97,444	-97,396004
4	-97,199997	-98,068001	-99,292	-94,956001	-96,227997
5	-98,788002	-98,531998	-93,540001	-97,292	-98,019997
6	-95,388	-89,620003	-90,339996	-98,307999	-98,575996
7	-97,348	-99,903999	-96,587997	-97,779999	-98,264
8	-97,332001	-98,332001	-94,916	-95,632004	-96,244003
9	-95,540001	-96,816002	-99,068001	-100,488	-97,351997
10	-96,779999	-89,959999	-91,112	-97,800003	-98,480003
11	-98,344002	-97,987999	-95,360001	-98,919998	-95,468002
12	-97,863998	-100,672	-99,931999	-97,236	-96,987999
13	-100,564	-93,571999	-96,959999	-98,723999	-100,512
14	-95,323997	-94,711998	-99,744003	-95,896004	-97,540001
15	-97,832001	-97,112	-95,639999	-96,683998	-96,540001

Figura 7. Información de los niveles de potencia obtenidos en la medición

En la figura 8 se observa la variación de los niveles de potencia de un canal durante el tiempo, donde cada time step representa una ranura de tiempo de 0,290 ms.

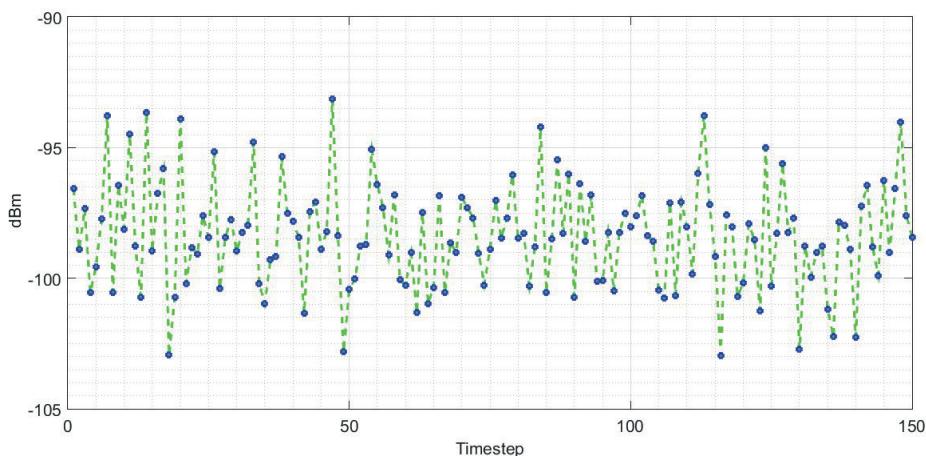


Figura 8. Información del tráfico espectral

Caracterización espectral

En este apartado se presenta el desarrollo preliminar del procesamiento de los datos obtenidos en la medición, se describe el proceso de normalización y luego, se procede con el cálculo de los tres parámetros requeridos como entradas al algoritmo GRA.

La caracterización del espectro permite a los SU identificar el comportamiento y la utilización de los canales en una banda de frecuencia. Es indispensable para los SU mantener la observación y establecer la disponibilidad del canal en instantes de tiempo debido a la actividad de cada PU [3].

Dentro de los parámetros de caracterización del espectro de este modelo se evaluaron la ocupación del espectro, el índice de potencia y la SNR.

Ocupación de espectral

La ocupación espectral permite observar la evolución temporal de cada canal de frecuencia, esta ocupación puede ser evaluada bajo el parámetro de ciclo de trabajo, el cual nos indica la relación entre el tiempo y el estado activo de los canales, o mediante el valor de disponibilidad absoluto, ‘0’ o ‘1’.

Ciclo de trabajo

El ciclo de trabajo es el porcentaje de tiempo en que un canal está siendo usado por un PU, es calculado mediante la ecuación (4).

$$\% \text{ Ocupación} = \frac{T_{on}}{\text{Tiempo total}} \quad (4)$$

Donde T_{on} representa el tiempo que estuvo ocupado el canal durante el tiempo total de medición, que para este caso es el tiempo de entrenamiento.

Otra manera de representar la ocupación de espectro es mediante la intensidad de espectro, la cual se calcula como lo muestra la ecuación (5) [43]:

$$\text{Intensidad} = \frac{T_{on}}{T_{on} + T_{off}} \quad (5)$$

En la figura 9 se observa el porcentaje de ocupación de cada canal en una fracción de tiempo de 60 minutos, en el intervalo de 824 MHz- 879 MHz, se puede observar que el foco de ocupación se presenta desde la frecuencia 854 MHz hasta 879 MHz.

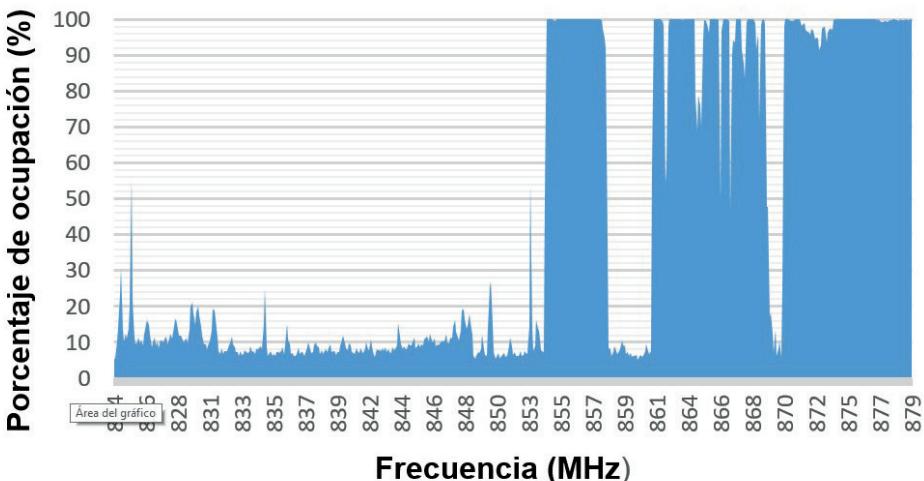


Figura 9. Ciclo de trabajo para la banda GSM

Índice de potencia

Es un dato que muestra la relación de los niveles potencia máxima $P_{\max k}$ y la mínima $P_{\min k}$ de cada canal de frecuencia k seleccionado [24] (ver ecuación (6)).

$$Pi_k = \frac{P_{\min k}}{P_{\max k}} \quad (6)$$

En la figura 10 se observa la relación de los niveles de potencia de cada canal que permite inferir que tan alto puede ser la señal de un PU cuando accede a un canal. Este cálculo permitirá saber qué índice de potencia es el más adecuado cuando hay una señal baja definiendo los canales, en donde es baja la intervención del PU.

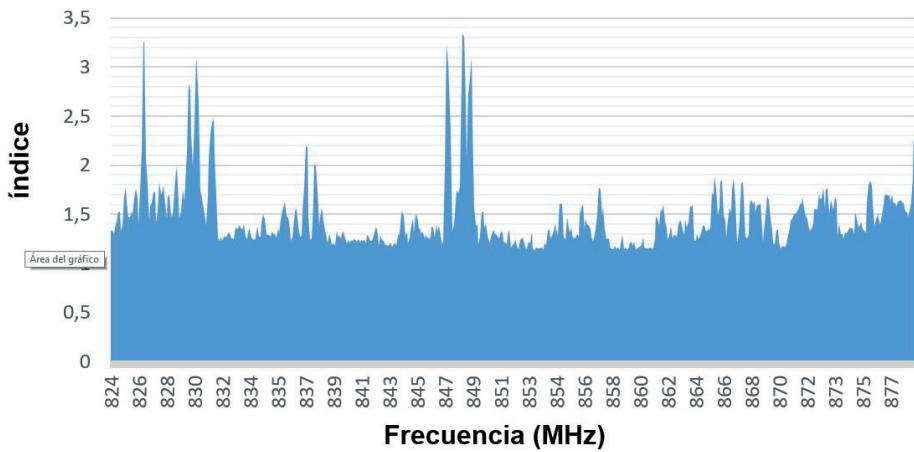


Figura 10. Índice de potencia por canal para la banda GSM

SNR

En esta parte, se calcula la SNR como la diferencia entre el nivel de recepción medido P_k en dBm y el nivel de piso de ruido obtenido con el analizador de espectro P_N como se muestra en la ecuación (7). El resultado es obtenido en dBm, como se observa en la figura 11.

$$SNR_k = P_k - P_N \quad (7)$$

Diseño de la etapa de clasificación

GRA es una técnica de ponderación de atributos efectiva de MADM que analiza la mejor alternativa con el mayor valor del coeficiente de relación gris (GRC) [49]. En el siguiente apartado se realizará paso a paso el diseño del algoritmo GRA.

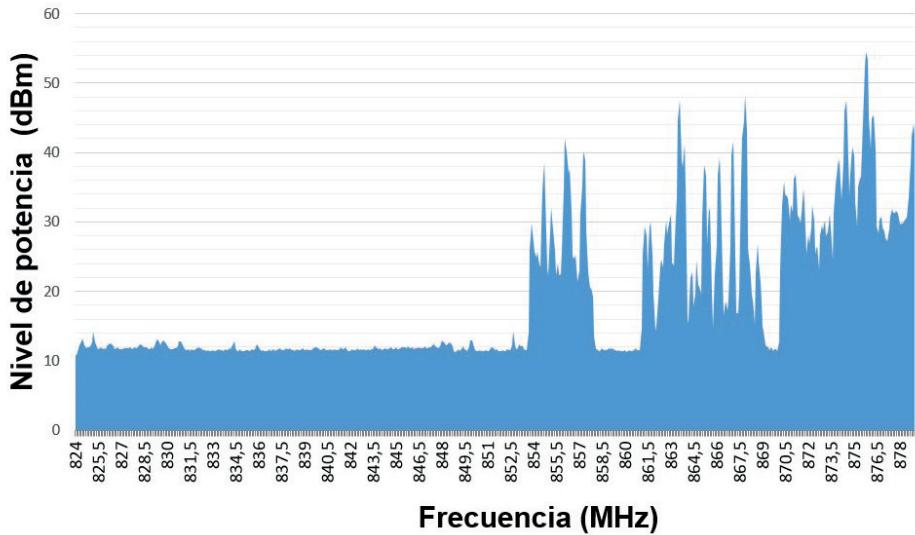


Figura 11. SNR para la banda 824 MHz - 874 MHz

Construcción del algoritmo GRA

Tomando como guía el trabajo realizado en [50], el método de implementación GRA fue desarrollado en los siguientes pasos:

Construir una matriz de decisión K para un modelo de k atributos y N datos, donde k define cada parámetro obtenido en el procesamiento de los datos como un atributo y N representa la cantidad de canales a analizar, como se observa en la ecuación (8).

$$K = \begin{pmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \\ \vdots & \vdots \\ k_{N,1} & k_{N,2} \end{pmatrix} \quad (8)$$

Normalizar la matriz de decisión K con los parámetros obtenidos de la caracterización del espectro para obtener la matriz S normalizada, a partir de la ecuación (9). Estos datos son parámetros de beneficio, ya que entre menor sea el valor de cada parámetro, podrá ser elegido en la toma de decisiones entre los canales menos ocupados.

$$S_{ij} = \frac{\max_{i \in N} k_{ij} - k_{ij}}{\max_{i \in N} k_{ij} - \min_{i \in N} k_{ij}} \quad (9)$$

Construcción de la matriz de juicios

En varias ocasiones el diseñador define un peso para cada atributo dependiendo de su criterio, dando prioridades a ciertas variables que pueden ser determinantes al momento de tomar decisiones. Para calcular los pesos de cada atributo, se realiza una comparación entre dos atributos basada en una tabla de valores numéricos absolutos llamada matriz de juicios. Para nuestro caso, se definen los pesos de cada atributo con base en el método Delphi el cual consistió en consultar a un conjunto de expertos la importancia y jerarquía de los tres parámetros elegidos.

Cálculo de los pesos normalizados

Luego de obtener la matriz de juicios de cada criterio, se calcula la media geométrica y el resultado se normaliza para obtener pesos entre 0 y 1, calculada de acuerdo con la ecuación (10).

$$w_i = \sqrt[n]{\prod_{j=1}^n a_{ij}} \quad (10)$$

Seguidamente, se construye la matriz ponderada normalizada. Esta matriz es el resultado de los pesos obtenidos multiplicados por cada elemento de la matriz, como lo muestra la ecuación (11).

$$v_i = w_j * S_{ij} \quad (11)$$

Acto seguido, se obtiene el valor máximo de cada criterio de beneficio calculado en la ecuación (12), para nuestro caso, se elige el canal con el valor máximo de cada criterio.

$$x_j = \max \{v_i, j = 1, \dots, N\} \quad (12)$$

Finalmente, se procede con el cálculo de la posición de las alternativas dado el valor GRC, el cual es calculado mediante la ecuación (13).

$$GRC_i = \frac{1}{N} \sum_{j=1}^N \frac{\Delta_{min} + \Delta_{max}}{\Delta_i + \Delta_{max}} \quad (13)$$

Donde $\Delta_{i=|x_i - v_{ij}|}$ es la distancia correlacionada gris, $\Delta_{max} = \max(\Delta_i)_{i \in k}$ y $\Delta_{min} = \min(\Delta_i)_{i \in k}$ son los valores máximos y mínimos de la distancia correlacionada gris.

En el apartado de resultados se podrá observar el grado de importancia de cada criterio a la hora de elegir el mejor canal para acceder. En definitiva, se obtendrá una tabla con los valores de frecuencia de los canales menos ocupados basados en los datos estadísticos medidos.

Diseño de la etapa de predicción

Una vez obtenida la clasificación por prioridad de los mejores canales a utilizar, se inicia un proceso de predicción basado en las estadísticas tomadas en la medición, para tráfico bajo y tráfico alto. Esta predicción le permitirá al SU identificar instantes de tiempo en los que puede existir una llegada del PU al canal y pueda tomar decisiones eficientes para realizar el cambio al segundo mejor canal dependiendo de su clasificación, como el trabajo realizado en [51], donde realizan predicciones implementando redes neuronales.

Para este modelo se utilizan 3 algoritmos de aprendizaje supervisado: redes neuronales NAR (sistema no lineal autoregresivo), SVM y KNN, los cuales serán evaluados para elegir el más consistente y pragmático. En cada algoritmo se utilizan los datos de entrenamiento como entradas al modelo y los datos de evaluación para medir que tan eficaz es la predicción. La base de datos es dividida de tal manera que los 60 minutos iniciales serán de entrenamiento y los 10 minutos finales serán de evaluación. En estos tres algoritmos se evalúa el desempeño para elegir el más acertado al momento de predecir y realizar el cambio de canal.

Implementación de redes neuronales

Mediante la herramienta de Matlab se realiza la implementación de la predicción utilizando redes neuronales artificiales, como primer paso, se realiza la normalización de los datos. Se entrena la red con los datos de entrada de cada canal y por último, se obtiene el resultado de la red que será la predicción para cada canal.

La red neuronal empleada se observa en la figura 12, esta red neuronal está basada en el modelo NAR, el cual consta de la capa de entrada, la capa oculta que tiene 10 neuronas y la capa de salida que utiliza una neurona. Se realizan pruebas con un número de neuronas menor a 10 en la capa oculta y se observa que con 10 neuronas se obtiene el error más pequeño. Cada dato de entrada está organizado de acuerdo con el canal y tiempo al que corresponde, para un total de 551 (canales) x 10800 (time step) datos de entrada.

Antes de ingresar los datos de entrenamiento es necesario ajustar las entradas a un rango específico que para este caso es [-1,1], como se observa en la ecuación (14).

$$y = \begin{cases} 1, & \text{si } Rx\ PU \geq -95 \text{ dBm} \\ 0, & \text{si } Rx\ PU < -95 \text{ dBm} \end{cases} \quad (14)$$

Luego, se entrena la red mediante el algoritmo Levenberg-Marquadt. Este método fue elegido ya que es implementado para mejorar la velocidad, optimiza el tiempo de entrenamiento y puede mejorar computacionalmente en la ejecución del modelo con respecto a otros algoritmos, ya que el tamaño de entrada de los datos para esta investigación es considerablemente alto.

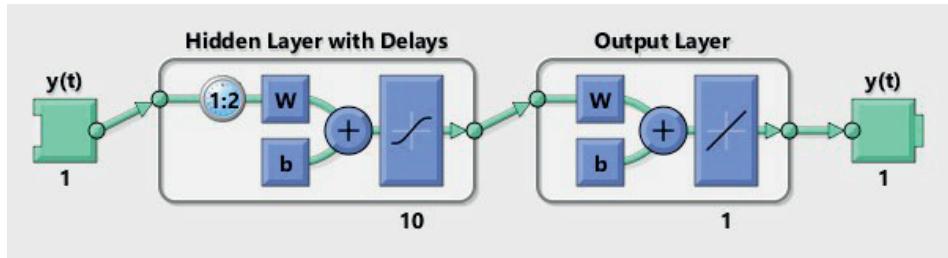


Figura 12. Sistema no lineal auto regresivo en redes neuronales

Fuente: tomada del software Matlab.

Luego de obtener el resultado de la red para cada canal, se realiza el proceso de conversión de los datos para obtener los valores reales de predicción. La formulación matemática se muestra en la ecuación (15), donde $y(t)$ será el próximo valor de los datos previos de $y(t)$ y d es la cantidad de retardos anteriores.

$$y(t) = f(y(t-1), \dots, y(t-d)) \quad (15)$$

Implementación de SVM

En la implementación de SVM, inicialmente, se realiza el procesamiento de los datos de entrada que servirán de entrenamiento en la etapa de clasificación. Estos datos son normalizados entre -1 y 1, donde los niveles por debajo de -95 dBm pertenecen al valor -1, el cual indica que el canal está desocupado en ese time step y los valores mayores a -95 dBm que se relacionan al valor 1, como un estado que indica que el canal está ocupado, esta definición puede verse en la ecuación (14).

En estos modelos se definen dos clases que serán las entradas para la función de clasificación. La clase 1 son los datos normalizados en el rango [-1,1] y la clase 2 son los niveles de potencia de cada canal medidos en dBm. Los conjuntos de datos de entrada están organizados por vectores de 1 canal x 10800 time steps, para un total de 551 x 10800.

Luego de obtener los datos de entrada definidos se inicia el proceso de clasificación aplicando la función `fitcsvm` de la herramienta Matlab. Para cada canal entrenado se obtiene un modelo de clasificación SVM que será la entrada en la etapa de predicción. Esta función de predicción está basada en el modelo matemático de la ecuación (16).

$$\hat{y} = \arg \min_{y=1:K} \sum_{k=1}^K \hat{P}(k|x) C(y|k) \quad (16)$$

Donde \hat{y} es el resultado de la predicción, K es el número de clases, $\hat{P}(k|x)$ es la probabilidad posterior del número de clases para observación x y $C(y|k)$ es el costo de clasificar una observación cuando su clave verdadera es K.

La función *predict* (de la herramienta Matlab) retornará un vector de etiquetas de clase con valores entre [-1,1] según el modelo de regresión entrenado en la clasificación. El vector resultante tiene un tamaño de 1 x 10800 time steps por cada canal. Al final, la predicción construirá el conjunto de datos de 551 canales x 10800 time steps. En la figura 13 se puede observar el proceso de implementación del algoritmo SVM.

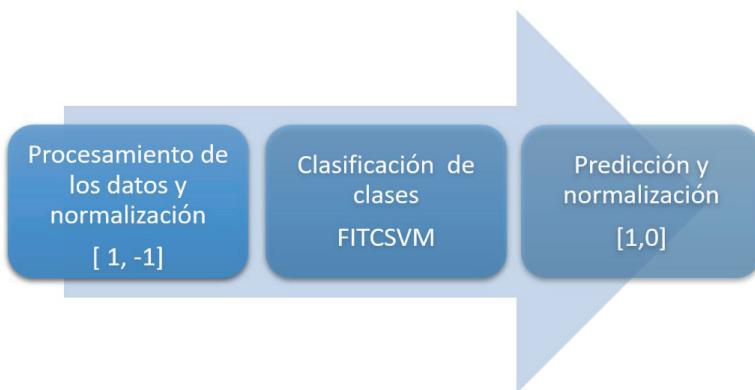


Figura 13. Proceso de predicción SVM

Implementación de KNN

Para KNN se ejecuta un procedimiento semejante a la técnica SVM. Primero, el algoritmo realiza el procesamiento de los datos normalizándolos entre 0 y 1, donde 0 señala que el canal está desocupado (valores menores a -95 dBm) y 1 determina un canal ocupado (valores mayores a -95 dBm). Ver ecuación (3).

A partir de la construcción de los datos de entrada se procede a entrenar los datos, implementando el algoritmo de clasificación KNN con las mismas dimensiones del conjunto de datos utilizado en SVM. Para cada traza de entrada 1 x 10800 time steps, la función *fitcknn* construirá un modelo de clasificación que será la entrada en el proceso de predicción. Al igual que en SVM se utiliza la función *predict* para predecir los nuevos datos con valores entre 0 y 1, y se construye el mismo conjunto de datos de 551 canales por 10800 time steps, como resultado de la predicción. En la figura 14 se puede observar la implementación del algoritmo KNN.

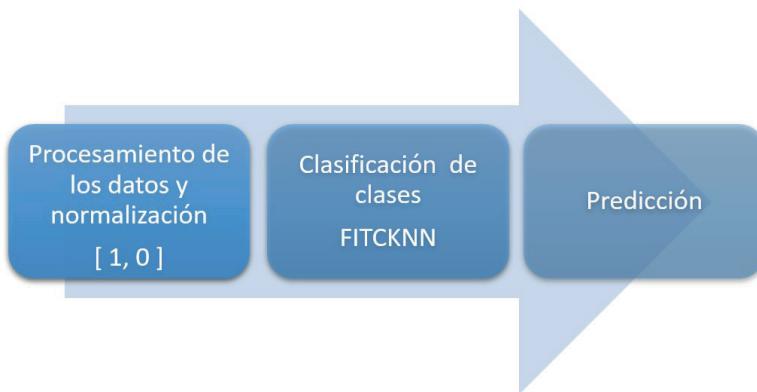


Figura 14. Proceso de predicción KNN

Modelo inteligente de decisión de espectro definitivo

Luego del estudio realizado en la comparación de los algoritmos de inteligencia artificial utilizados en la predicción y los algoritmos aplicados en los modelos de toma de decisiones en el espectro, se propone un modelo definitivo que se adapte al propósito general de este proyecto.

Dentro de las cualidades principales que debe contener el modelo está la capacidad de tomar de decisiones inteligentes y oportunas, buscando la manera de lograr una adaptación entre el SU y el ambiente de RF. Otra calidad importante es administrar la movilidad de los SU con el fin de cada salto de canal o handoff espectral sea acertado y que en una transmisión se realicen la mínima cantidad de saltos con el propósito de reducir las colisiones y la interferencia entre señales.

Basado en lo anterior, se definieron dos algoritmos fundamentales dentro del modelo que permitirán a los SU acceder al espectro de una manera dinámica. El modelo está construido en fases como se observa en la figura 15. La primera fase del modelo está definida como la etapa de caracterización del espectro. Luego de obtener los resultados de caracterización, el modelo continúa su ejecución a la fase 2, donde se desarrolla la clasificación del espectro empleando el algoritmo GRA. Finalizada la etapa de clasificación, el modelo ejecuta la fase 3, predecir la ocupación espectral en los canales seleccionados mediante el algoritmo elegido, y finalmente, realizar la correspondiente evaluación y comprobación.

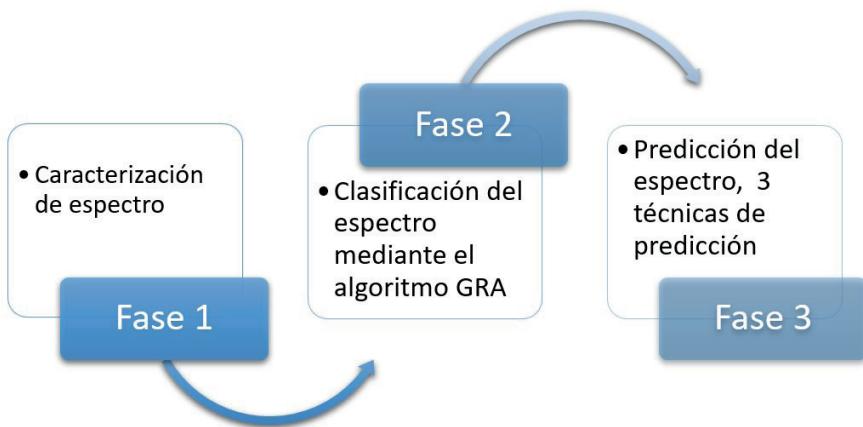


Figura 15. Modelo de decisión de espectro definitivo

Metodología de validación y evaluación

En esta sección se explica el procedimiento de evaluación del desempeño del modelo propuesto, analizando por partes los algoritmos de clasificación y predicción a partir del siguiente proceso de análisis: primero, en la etapa de clasificación, se evalúa la precisión con que se eligen los canales con menos ocupación; segundo, en la etapa de predicción, se analiza el error porcentual de las técnicas de inteligencia artificial implementadas en este modelo; tercero, se presenta el análisis y evaluación del comportamiento del SU en los handoff realizados por el SU; y finalmente, se evalúa el costo computacional del modelo.

Métricas de evaluación

Las métricas de evaluación son distintas para cada fase del modelo. En la etapa de clasificación (fase 2) se evalúa la precisión con que el algoritmo de clasificación realiza su proceso utilizando diferentes mediciones para varias tecnologías inalámbricas, mediante el uso de histogramas.

Para evaluar los resultados obtenidos en la etapa de predicción (fase 3), se realizan dos mediciones que permitirán evaluar el desempeño de las tres técnicas de predicción implementadas. La primera medición calcula el error relativo porcentual para los resultados de tráfico bajo y tráfico alto, utilizando los datos de evaluación versus el resultado de la predicción. La segunda medición calcula la cantidad de cambios de canal o handoff espectrales realizados por el SU durante la transmisión, para esta métrica es necesario construir un simulador que permita estimar los diferentes tipos de handoff considerados en esta investigación, tales como: handoff totales, handoff fallidos, handoff perfectos, handoff anticipados, handoff con interferencia y handoff sin interferencia, que serán explicados en la siguiente sección con más detalle.

Precisión en la clasificación

Es necesario asegurar que la clasificación de los canales sea acertada y que no existan errores en cada cálculo que puedan afectar la posterior elección, y que la predicción sea aprovechada eficazmente. Para ello se realizó un experimento que consiste en calcular la cantidad de handoff mínimos a realizar por el SU en diferentes trazas y de esta manera comparar los resultados con los del simulador diseñado. Estos experimentos serán explicados con más detalles en el capítulo de descripción de los experimentos.

Error relativo porcentual

El error relativo es calculado como se observa en la ecuación (17). Donde X_i representa el resultado obtenido en cada canal en un instante de tiempo y X_v es el dato de la etapa de evaluación, debido a que los valores resultantes están en el rango de -1 a 1, se procede a normalizar estos datos antes de aplicar la ecuación (17).

$$error = \frac{1}{n} \sum_{i=1}^n \left(\frac{|X_v - X_i|}{|X_v|} * 100 \right) \quad (17)$$

Este cálculo es ejecutado con los datos de medición obtenidos para tráfico bajo y tráfico alto de la tecnología GSM.

Métricas de handoff espectral

Para el cálculo del handoff se diseñó un algoritmo capaz de medir la cantidad de saltos que debe realizar un SU en una transmisión de 10 minutos, aproximadamente. El algoritmo determina esta medición basado en el resultado de la predicción de las tres técnicas con el fin de determinar la técnica que más se ajuste al modelo propuesto.

El escenario diseñado compara en cada time step los datos obtenidos en la predicción junto con los datos reales de evaluación, calculando 6 métricas de handoff, las cuales son:

- Handoff total
- Handoff fallidos
- Handoff con interferencia
- Handoff sin interferencia
- Handoff anticipados
- Handoff perfectos

El *handoff total* es el acumulado total de cambios de canal. En esta métrica se describe la cantidad de saltos realizados por el SU al ejecutar el modelo diseñado hasta completar una transmisión de 10 minutos. El simulador es el encargado de ir contando cada salto de canal.

A medida que el SU recorre el espectro y realiza cambios de canal, se pueden seleccionar canales donde hay algún PU ocupándolo, a este tipo de cambio de canal se denominó *handoff fallido*, que indican el número de handoff cuyo canal objetivo está siendo ocupado por un PU.

Otra métrica importante a determinar es la cantidad de cambios de canal que generan interferencia. Los *handoff con interferencia* muestra la cantidad de saltos realizados con posterioridad a la llegada de un PU. Al contrario, los *handoff sin interferencia* indican la cantidad de saltos realizados con anterioridad a la llegada de un PU.

Los *handoff anticipados* son los cambios de canal sin interferencia que se realizaron con mucha anterioridad a la llegada del PU. En esta acción la predicción no puede ser tomada como acertada dado que no se utiliza todo el tiempo que el canal permanece desocupado.

Finalmente, los *handoff perfectos* son los handoff sin interferencia que se realizaron muy cerca de la llegada de un PU, esta métrica permitirá evaluar la certeza en la predicción de los algoritmos evaluados.

Costo computacional

Para el análisis del rendimiento computacional del modelo, se elabora la medición del tiempo de ejecución del modelo empleando la herramienta de Matlab. Esta medición es realizada tomando mediciones del tiempo de ejecución de cada fase que compone el modelo. La medición inicia con la fase de caracterización del espectro y el cálculo de los parámetros utilizados, luego se mide el tiempo que tarda el algoritmo GRA en clasificar los 551 canales y finalmente se determina el retardo de la predicción de cada canal implementando las tres técnicas de aprendizaje automático.

La herramienta Matlab cuenta con una función TIC y TOC que permite medir el tiempo transcurrido de un algoritmo desarrollado simplemente colocando el TIC en la parte inicial del algoritmo y la función TOC en la parte final, y se reflejará el tiempo transcurrido en la ventana de comandos.

Descripción de los experimentos

Para evaluar el desempeño de los algoritmos de clasificación y predicción empleados en este estudio, se diseñaron dos experimentos, los cuales se describen a continuación.

Experimento 1

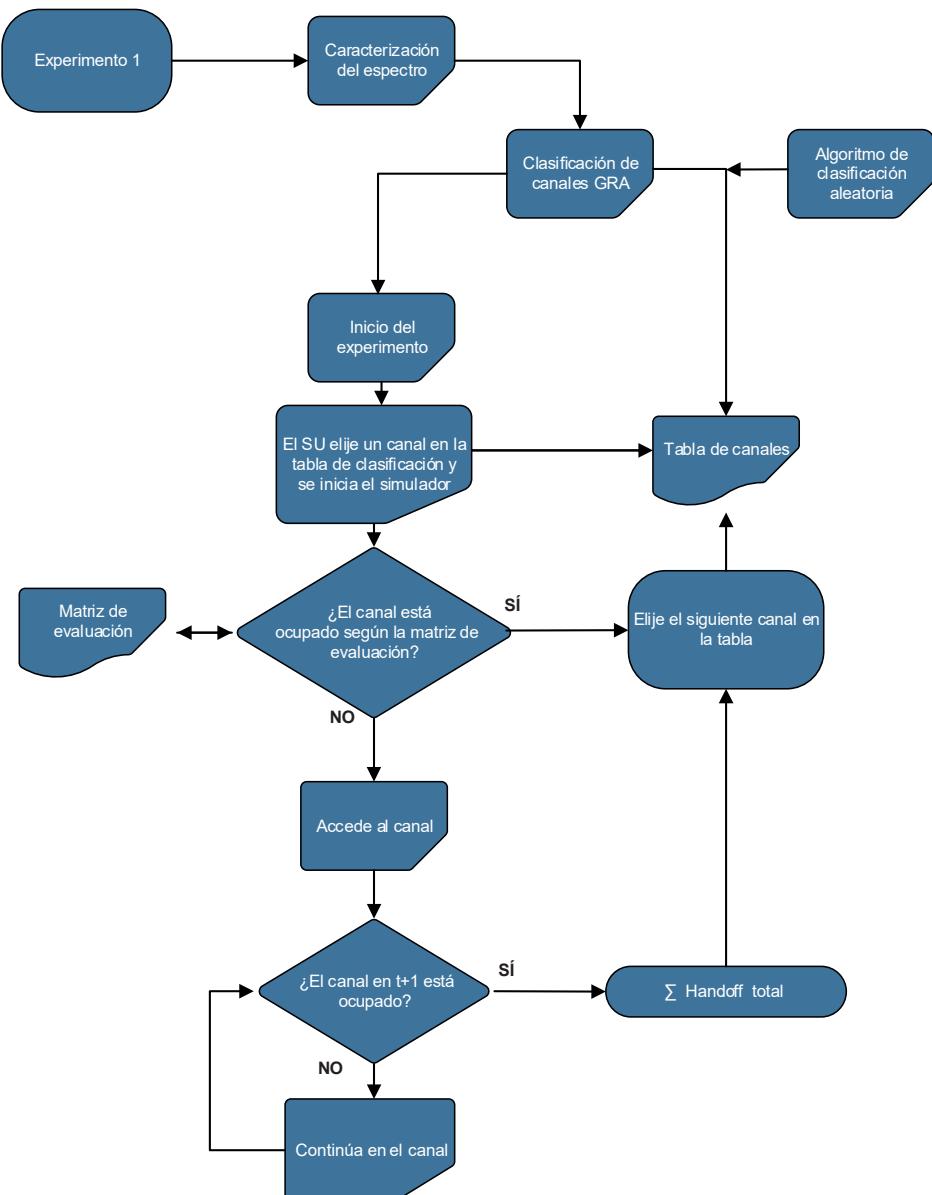
Este experimento consiste en implementar un algoritmo de clasificación aleatoria de los canales y otra mediante el algoritmo GRA, luego se organizan los mejores canales operativos basados en las trazas de tráfico bajo y tráfico alto con la finalidad de evaluar ambos procesos de clasificación.

Este experimento es realizado 50 veces para obtener un valor promedio de handoff mínimos, tanto de la clasificación aleatoria como del algoritmo GRA.

- *Paso 1:* el experimento inicia calculando los parámetros de caracterización de espectro, que serán datos de entrada para el algoritmo de clasificación GRA.

- *Paso 2:* se diseña un algoritmo que organiza los canales de manera aleatoria, utilizando la función rand de Matlab, la cual retorna un número de distribución uniforme. Los datos son ajustados en el intervalo de 1 a 551.
- *Paso 3:* una vez ordenados los canales por prioridad, se ejecuta el simulador desarrollado.
- *Paso 4:* el SU realiza dos recorridos por las trazas de la etapa de evaluación; el primero, ejecutando el algoritmo aleatorio y el segundo, ejecutando el algoritmo GRA. El SU inicia observando si en el instante $t+1$ el canal seleccionado está ocupado por un PU, en cuyo caso procederá a cambiar al siguiente canal de acuerdo con la clasificación previa.
- *Paso 5:* el SU realiza el mismo proceso de acceso al espectro trasmitiendo en los canales necesarios hasta completar la transmisión de 10 minutos, mientras va contando los handoff requeridos para no generar interferencia. Este proceso es realizado 50 veces para los dos tipos de clasificación, al final se promedia la cantidad de handoff de las 50 pruebas.

El proceso del experimento se puede observar en la figura 16, donde se observan los dos algoritmos de predicción planteados, cabe aclarar que el proceso es ejecutado con un solo algoritmo de clasificación a la vez.

**Figura 16.** Diagrama de flujo del experimento 1

Experimento 2

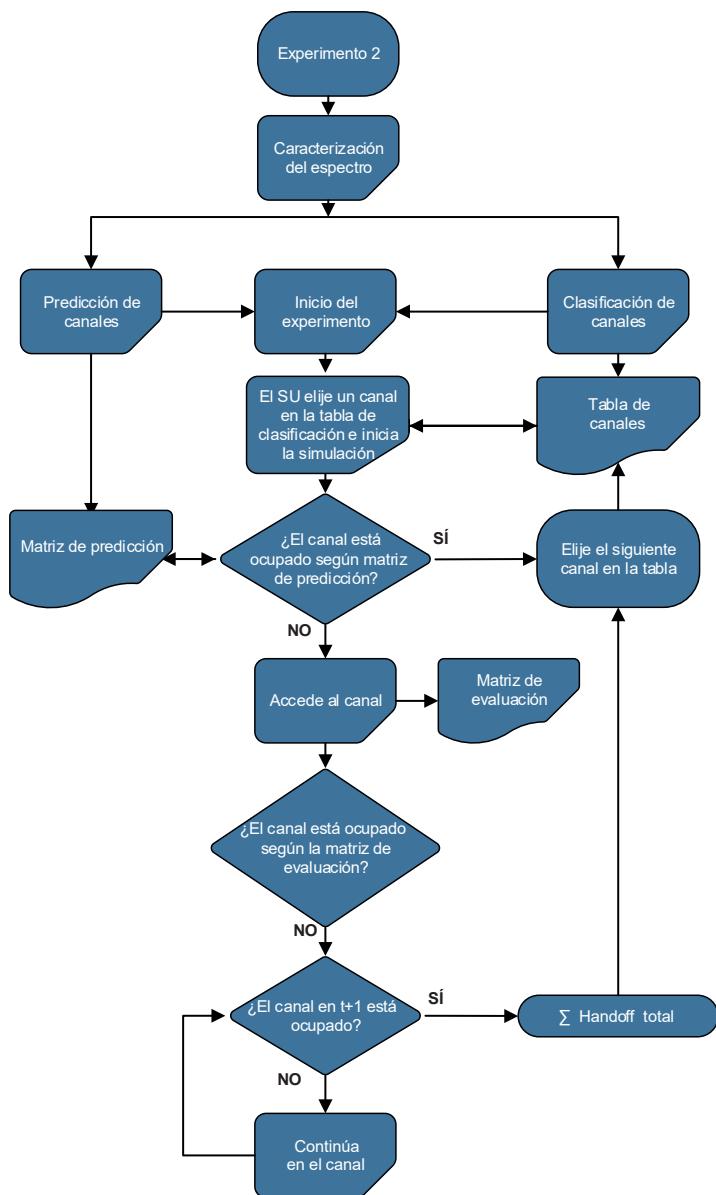
En el diagrama de flujo de la figura 17, se pretende obtener resultados que describan el comportamiento de los tres algoritmos de predicción en un escenario de entorno espectral de tráfico bajo y tráfico alto, en la banda de frecuencia GSM. Este experi-

mento simula el recorrido de un SU transmitiendo con niveles de potencia ideales durante 10 minutos por las trazas de evaluación de tráfico bajo y tráfico alto, para calcular la cantidad de los diferentes tipos de handoff espectrales realizados.

El experimento consta de los siguientes pasos:

- *Paso 1:* el SU inicia ejecutando los parámetros de caracterización de espectro y clasifica los canales, organizándolos del menos ocupado al más ocupado, de acuerdo con el historial de tráfico de la última hora.
- *Paso 2:* el SU inicia la transmisión en el canal seleccionado durante 5 minutos.
- *Paso 3:* cuando el SU accede al espectro, calcula la predicción para el canal seleccionado. En cada instante observa la predicción para saber en qué momento llega un PU. De esta manera el SU actúa cambiando a otro canal y continúa la transmisión hasta terminar los 5 minutos.
- *Paso 4:* durante este proceso el simulador se encarga de crear el escenario de ocupación espectral con los datos reales obtenidos de la campaña de medición, y a su vez, realiza el conteo del número de handoff realizados en el proceso de predicción.

Este proceso es realizado 50 veces. En cada repetición el SU inicia 18 time step más adelante y durante 900 time step que equivalen a 5 minutos, este desplazamiento inicial se realiza hasta barrer toda la matriz de evaluación que consta de 10 minutos, de esta forma se garantiza un escenario diferente para cada repetición. En cada ejecución del experimento se tomarán medidas de los diferentes tipos de handoff. El resultado de este experimento es presentado en el apartado “Evaluación comparativa”.

**Figura 17.** Diagrama de flujo del experimento 2

Simulador

Para medir y evaluar el modelo diseñado es necesario construir un simulador que pueda reflejar, mediante un experimento, el comportamiento del tráfico real de una red inalámbrica, de tal manera que se pueda evaluar el modelo diseñado. En este simulador, el SU puede realizar su proceso de ejecución, observando su

funcionamiento mediante la captura de datos y mediciones necesarias para evaluar los algoritmos de predicción.

El simulador es diseñado utilizando las trazas de entrenamiento y evaluación, obtenidas en la campaña de medición, de tal manera que la traza de entrenamiento servirá para configurar el modelo y la traza de evaluación será el entorno de RF donde se desplazará el SU y realizará su interacción.

De esta manera el modelo es ejecutado implementando las trazas reales de tráfico, donde inicialmente el SU calcula todos los parámetros del espectro y genera la tabla de prioridad de los canales menos utilizados. El SU tiene dentro de su base de datos la tabla de clasificación. Luego de ello, el SU inicia su recorrido de transmisión realizando la predicción del primer canal seleccionado. El SU accede al canal y predice en qué momento puede llegar el PU al canal, en caso de que se dé en el instante $t+1$, el SU cambiará de canal en el instante t , eligiendo el siguiente canal dentro de la tabla de clasificación.

El diseño del simulador está basado en un algoritmo que compara dos matrices, la primera matriz son los últimos 10 minutos de la medición realizada en la captura de datos, también llamada etapa de evaluación, la cual además es utilizada para evaluar la predicción, y la segunda matriz, representa el resultado de la predicción. Para este caso, el SU realiza la predicción de todos los canales según el orden de clasificación y construye su propia matriz.

Las matrices son construidas de tal manera que las columnas representan cada time step o paso en el tiempo y las filas son los canales organizados por prioridad, de tal manera que la primera fila será el primer canal a elegir. La matriz del simulador se muestra en la figura 18, donde los recuadros azules son espacios en el espectro y la señal en rojo muestra la interacción del SU.

Dentro de una transmisión, el SU cambiará de canal, si es necesario, hasta finalizar la transmisión. Cada vez que el SU requiera realizar una transmisión actualizará su tabla de clasificación de canales.

Frecuencia (MHz)	Prioridad	Canal	1	1	1	1	1	1	0	1	1
			839,1	839,2	839,3	839,4	839,5	839,6			
	1	0	0	1	1	0	1	1	1	1	1
	2	1	1	0	1	1	1	1	1	1	1
	4	1	1	1	1	0	0	1	1	1	1
	3	1	1	1	0	1	1	1	1	1	1
	5	1	1	1	1	1	1	0	1	1	1
	6	1	1	1	1	1	1	1	0	0	0
SU →			0,29	0,58	0,87	1,16	1,45	1,74	2,03	2,32	2,61
			Tiempo (seg)								

Figura 18. Construcción de matrices del simulador

El SU inicia el recorrido en la matriz eligiendo la primera fila y analizando en qué instantes de tiempo puede existir una llegada, según la matriz de predicción. De esta manera, el SU analizará estados futuros y accederá proactivamente al próximo canal hasta el final de su transmisión.

Con este funcionamiento el simulador realiza una serie de cálculos y mediciones de handoff que permiten obtener información estadística de los tres algoritmos de predicción para definir que técnica es la más adecuada para su implementación.

La figura 19 describe el diagrama de flujo del simulador.

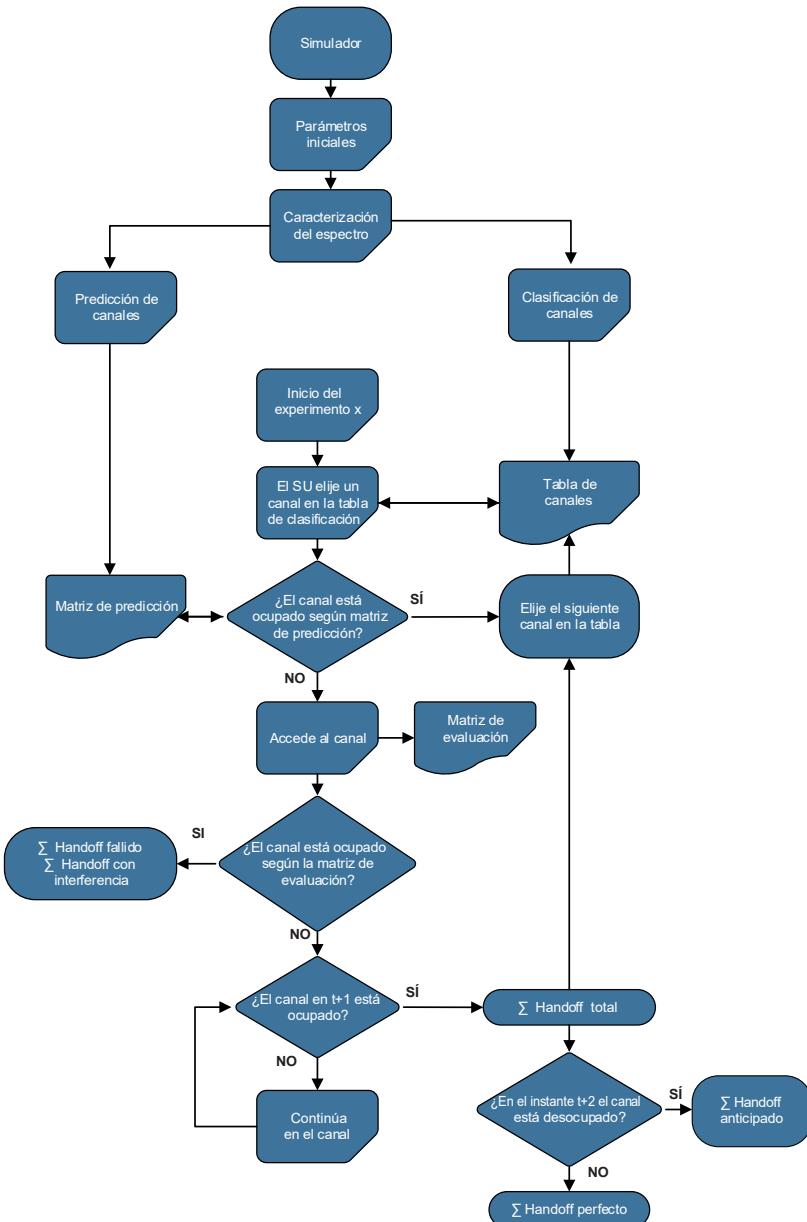


Figura 19. Diagrama de flujo del simulador

Resultados

Los resultados alcanzados en el actual proyecto fueron obtenidos de la simulación desarrollada con la herramienta Matlab. Estos resultados serán presentados mediante la evaluación del algoritmo de clasificación y el algoritmo de predicción. Luego de ello, se realiza el cálculo del error de predicción de las tres técnicas que permitirán elegir la más conveniente para ser implementada en el modelo final y de esta manera lograr mejorar el desempeño de las redes de radio cognitiva.

Algoritmo GRA

En la etapa de evaluación del algoritmo de clasificación se valoran inicialmente los parámetros seleccionados como criterios para la entrada del algoritmo, luego se analiza el cálculo de los pesos normalizados y finalmente, se evalúa el valor obtenido del análisis relacional gris.

Parámetros elegidos como criterios

Luego de un importante estudio de diferentes investigaciones, se han definido tres parámetros que describen el comportamiento de cada canal y pueden funcionar como criterios en el análisis de decisiones multicriterio.

Estas variables seleccionadas son:

- Ocupación espectral
- Índice de potencia
- Relación señal a ruido

Otras variables que inciden en el diseño de un modelo de toma de decisiones son el ancho de banda y la posición del SU, pero debido al objetivo propuesto en este modelo, el cual pretende elegir el canal menos ocupado sin generar interferencias en los PU, no serán tenidos en cuenta; para estos parámetros se asignan valores fijos, como por ejemplo el ancho de banda de 100 kHz por cada canal.

Para evitar la interferencia es necesario escoger un canal que haya tenido históricamente poca actividad, por lo que el parámetro de ocupación de espectral resulta ser el más importante. Como se explicó anteriormente, este parámetro es medido con base en los datos históricos, donde el porcentaje de ocupación más bajo corresponde al canal menos ocupado.

El índice de potencia es un parámetro que permite identificar la relación de los niveles de potencia que generan los PU en cada canal, en este modelo este parámetro precisa que entre menor sea esta relación más baja es la utilización de respectivo canal.

La SNR describe la relación de la señal con respecto al ruido en cada canal, el cual es calculado tomando como base el nivel de piso de ruido obtenido en la medición. Los datos calculados de los parámetros descritos se muestran en la tabla 3, donde los resultados han sido normalizados entre 0 y 1. Cada fila es un canal de 100 KHz en la banda de frecuencia 824 MHz - 874 MHz.

Tabla 3. Normalización de los parámetros para los 10 primeros canales

Ocupación de espectro	Índice de potencia	SNR
1	0,9125	1
1	0,9125	1
0,9559	0,9272	0,9815
0,9179	0,8994	0,9744
0,9234	0,8673	0,9762
0,9120	0,8269	0,9740
0,9277	0,8238	0,9744
0,9520	0,9120	0,9811
0,9471	0,8879	0,9798
0,9286	0,7508	0,9779

Tabla de criterios

Se realiza la consulta de la importancia de los tres parámetros seleccionados a una serie de evaluadores que, debido a su experiencia y conocimiento adquirido en la literatura estudiada de radio cognitiva, tienen la potestad de brindar opiniones acerca de los criterios. En la tabla 4 se observan los criterios.

Tabla 4. Criterios para establecer pesos en el método GRA

Criterios	Ocupación de espectro	Índice de potencia	SNR
Ocupación de espectro	1	5	9
Índice de potencia	1/5	1	3
SNR	1/9	1/3	1

Pesos normalizados

Los pesos obtenidos describen el grado de importancia de cada parámetro o criterio al momento de tomar decisiones. Los pesos normalizados son valores obtenidos a partir de los datos de la tabla 4 y de la media geométrica descrita en la ecuación (10). En la tabla 5 se observan los pesos asignados a cada criterio.

Tabla 5. Pesos normalizados de los criterios

Criterios	Ocupación de espectro	Índice de potencia	SNR
Pesos normalizados	0,7235	0,1932	0,083

Los pesos obtenidos se evalúan en todos los canales disponibles de la banda de frecuencia seleccionada. El resultado de la tabla 5 permite inferir que al decidir el mejor canal un 72,35 % dependerá de que el canal que disponible, el 19,32 % depende de que el índice de potencia sea bajo y un 8,3 % será producto de la relación señal a ruido.

En la tabla 6 se puede observar la matriz ponderada normalizada para los 10 primeros canales, la cual es calculada mediante la ecuación (11).

Tabla 6. Matriz ponderada normalizada

Ocupación de espectro	Índice de potencia	SNR
1	0,21639	0,09371
1	0,216392	0,09371
0,95599	0,219882	0,09198
0,91797	0,21327	0,09131
0,92347	0,205667	0,09148
0,91209	0,196098	0,09128
0,92773	0,195359	0,09131
0,95204	0,216280	0,09194
0,94711	0,210544	0,09182
0,92868	0,178046	0,09164

Análisis relacional gris

Luego de definir los pesos para cada criterio, el algoritmo realiza el procedimiento del cálculo de las distancias de gris para cada canal. Luego, estos valores son organizados de mayor a menor en un rango entre 0 y 1, donde el valor más cercano a 1 será la primera opción de elección de canal.

En la tabla 7 se puede observar la selección de canales obtenida de la ejecución del algoritmo GRA utilizando los datos medidos de tráfico bajo, para los 10 mejores canales.

Tabla 7. Distancia correlacionada gris GRC tráfico bajo

Prioridad	Frecuencia (MHz)	GRC
1	839,3	0,97903
2	843,5	0,97810
3	839,1	0,97717
4	842,3	0,97648
5	843,2	0,97629
6	838,80	0,97495
7	843,3	0,97439
8	839	0,97344
9	842,8	0,97325
10	824	0,97319

La tabla 7 es una lista de respaldo de canales, en el caso de requerir un cambio de canal. Cuando el SU necesite cambiar de canal elegirá el siguiente canal en su tabla con mayor distancia correlacionada gris.

Este proceso de clasificación es realizado cada vez que el SU requiera hacer una transmisión, por lo que la tabla de clasificación cambiará constantemente.

Métricas de evaluación

Una métrica de evaluación importante es conocer los tiempos de ejecución de cada proceso del modelo para analizar qué técnicas utilizan menos recurso computacional, a continuación, se muestran las medidas obtenidas.

La evaluación del algoritmo de predicción permite determinar la mejor técnica que al predecir los nuevos datos se ajusten a los datos históricos de cada canal y se comporte de manera semejante a su experiencia. El mejor algoritmo debe ser eficaz y assertivo en la predicción.

Una de las maneras de analizar los resultados de la predicción e identificar el algoritmo correcto es calculando el error relativo porcentual para cada técnica como se presenta en la siguiente sección.

Evaluación comparativa para la selección de canal

En la evaluación comparativa se muestran los resultados obtenidos en los experimentos realizados. Primero, se muestran los cálculos de handoff totales del experimento 1, indicando para cada minuto la cantidad de handoff realizados en los dos tipos de tráfico. En la figura 20 se puede observar el comportamiento de la clasificación en un escenario de tráfico bajo donde el SU realiza un total de 135 handoff, eligiendo primero los canales menos ocupados. En el mismo escenario, pero implementando un algoritmo aleatorio el SU realiza 220 handoff en el mismo tiempo de transmisión. Para tráfico alto también se puede identificar la importancia de clasificar canales, ya que el SU realiza 180 handoff mientras que de manera aleatoria realiza 350, este resultado se muestra en la figura 21.

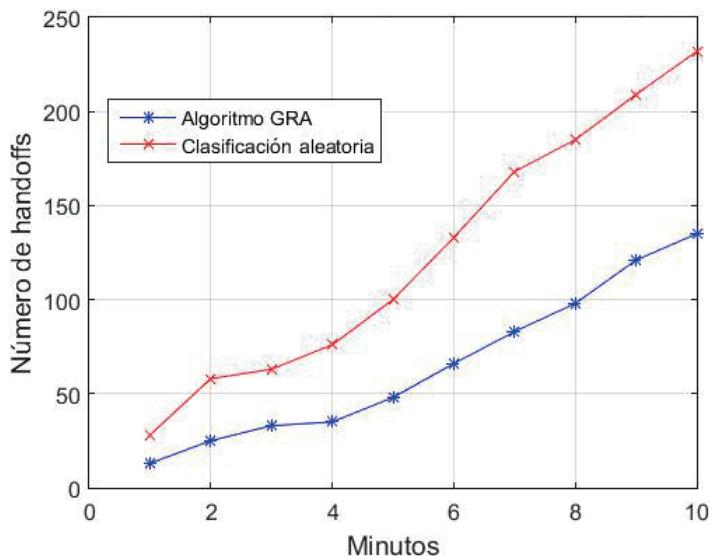


Figura 20. Handoff totales tráfico bajo experimento 1

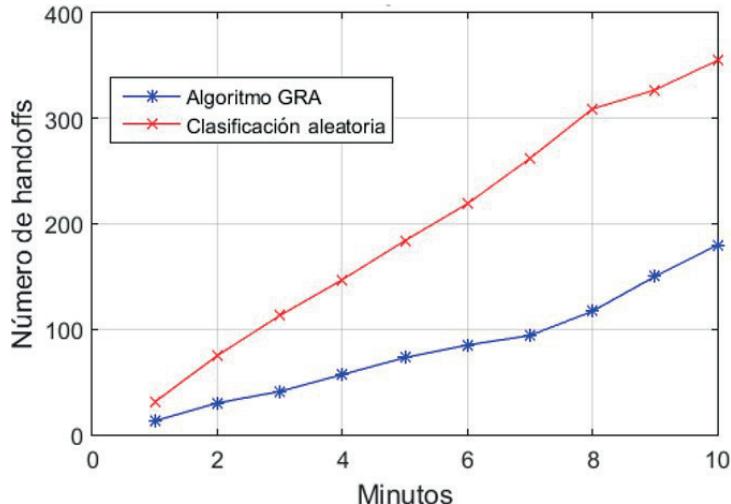


Figura 21. Handoff totales tráfico alto experimento 1

En la figura 22 se presenta el promedio de handoff del experimento 1, después de repetirlo 50 veces.

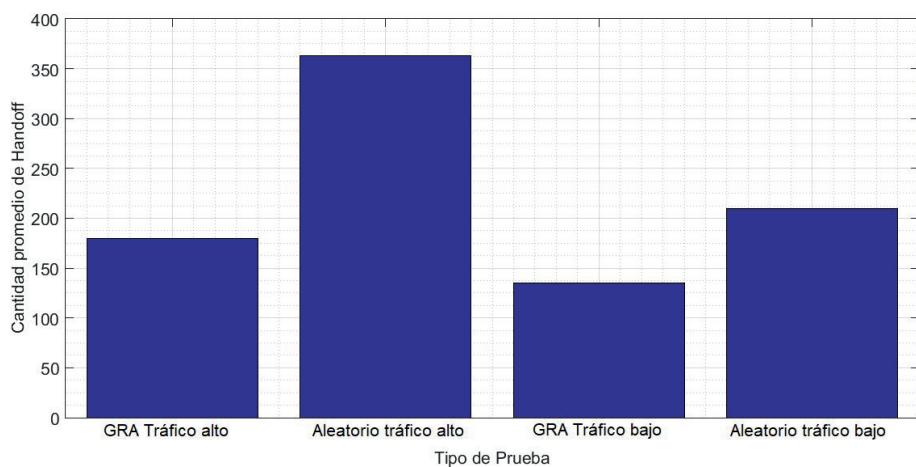


Figura 22. Promedio de handoff experimento 1

Cálculo del error de predicción

El resultado del error relativo promedio por cada técnica de predicción para tráfico alto y tráfico bajo, se observa en la tabla 8. Este valor permite mostrar el nivel de efectividad en la predicción, a pesar que la traza de evaluación utilizada es solo de 10 minutos de medición, en tráfico alto se puede tener una certeza en la predicción del 88,03 % para el caso de SVM.

Tabla 8. Error relativo promedio para tráfico bajo y tráfico alto

Error relativo promedio	SVM	Índice de potencia	SNR
Tráfico alto	11,97	13,09	16,44
Tráfico bajo	13,85	14,21	18,42

De la tabla 8 se puede observar que para tráfico alto, el algoritmo SVM presenta una probabilidad del 88,03 % y KNN del 86,15 %, de acertar en la predicción, a diferencia de la red neuronal que tiene una probabilidad del 83,5 %. Para tráfico bajo el comportamiento es similar en los tres algoritmos bajando la probabilidad de predicción a un 86,15 % para SVM, 85,79 % para KNN y 81,58 % para la red neuronal. El resultado del error relativo es el promedio del resultado de los 551 canales, donde se observa que para los canales menos ocupados la probabilidad de predicción llega al 92 % con la técnica SVM. Las técnicas SVM y KNN pueden presentar mejor estabilidad y coherencia en los resultados, así como mejorar el rendimiento en la toma de decisiones.

Análisis del costo computacional

Dentro de las métricas de evaluación, consideramos inicialmente la medición del tiempo que tarda en realizar el proceso de cálculo de los parámetros para los 551 canales, el cual se puede observar en la tabla 9. También se determina el tiempo que el algoritmo GRA tarda en clasificar los canales, observado en la tabla 10. Por último, se realiza la medición de la predicción por canal para ambos tráficos, estos resultados se pueden ver en la tabla 11.

Tabla 9. Tiempo de ejecución del cálculo de los parámetros

Ciclo de trabajo	Índice de potencia	SNR	Cálculo de los parámetros
0,37719 S	1,01948 S	0,09312 S	1,6666 S

Tabla 10. Tiempo de ejecución de la etapa de clasificación

Clasificación GRA	
Tráfico alto	Tráfico bajo
0,041782 Seg	0,024856 Seg

Tabla 11. Tiempo de ejecución de la etapa de predicción

Predicción	SVM	KNN	Neuronal
Predicción tráfico alto por canal	0,440 Seg	1,415 Seg	0,611 Seg
Predicción tráfico bajo por canal	0,392 Seg	1,452eg	0,601 Seg

Las especificaciones técnicas del equipo de cómputo utilizado en el desarrollo del proyecto son mostrados en la tabla 12.

Tabla 12. Especificaciones técnicas del equipo de cómputo

Característica	Descripción
Equipo y marca	Portátil Lenovo
Serie	Idea pad 500
Procesador	Intel ® Core™ i7 6500U CPU @ 2.50 GHz - 2.6GHz

Característica	Descripción
Memoria RAM	8 GB
Sistema de almacenamiento	1 TB
Sistema operativo	Windows 10, 64 bits

Evaluación comparativa para la predicción

El segundo experimento consistió en medir la cantidad de handoff realizados por el SU durante una transmisión de 5 minutos. En cada ejecución se realizó el acceso al espectro aumentando cada 6 segundos el tiempo de inicio del experimento, hasta completar toda la traza de 10 minutos. Para este ejercicio solo se tuvo en cuenta la métrica de handoff total. Los datos obtenidos en este experimento son presentados en la figura 23, para los dos tipos de tráfico.

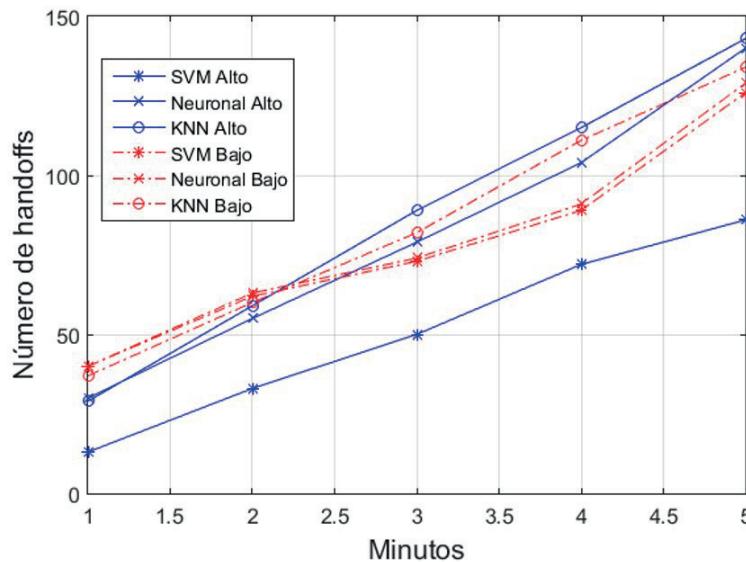


Figura 23. Handoff totales experimento 2

Para tráfico alto el algoritmo de redes neuronales junto con KNN son las técnicas que realizan más handoff en comparación con SVM, como se observa en la figura 23. La tabla 13 muestra los valores absolutos de dicho experimento.

Tabla 13. Handoff totales experimento 2

Prueba	Rango time step		SVM Total Handoff		KNN Total Handoff		Neuronal Total Handoff	
			Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo
1	1	901	87	107	131	105	137	107
2	19	919	91	106	133	107	143	106
3	37	937	94	105	134	107	146	107
4	55	955	98	105	136	109	150	109
5	73	973	99	107	141	112	150	109
6	91	991	100	107	141	111	110	109
7	109	1009	104	109	144	112	112	114
8	127	1027	104	113	145	94	113	113
9	145	1045	104	116	128	93	117	115
10	163	1063	105	118	129	96	120	118
11	181	1081	92	122	130	104	123	122
12	199	1099	92	123	132	104	126	124
13	217	1117	97	125	133	92	126	123
14	235	1135	98	126	138	92	126	125
15	253	1153	100	131	128	96	128	131
16	271	1171	103	131	131	101	131	133
17	289	1189	104	134	132	103	131	139

Prueba	Rango time step		SVM Total Handoff		KNN Total Handoff		Neuronal Total Handoff	
			Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo
18	307	1207	107	119	134	109	106	133
19	325	1225	109	122	138	111	108	133
20	343	1243	113	98	139	115	112	125
21	361	1261	117	106	141	117	116	124
22	379	1279	122	110	144	117	120	126
23	397	1297	123	112	146	117	122	130
24	415	1315	127	113	149	122	125	134
25	433	1333	131	114	134	124	129	117
26	451	1351	132	118	136	127	132	121
27	469	1369	134	125	140	129	141	128
28	487	1387	103	126	143	134	140	129
29	505	1405	105	126	144	134	140	129
30	523	1423	108	101	145	135	142	131
31	541	1441	109	100	145	119	110	134
32	559	1459	107	101	148	121	114	135
33	577	1477	109	105	151	123	114	137
34	595	1495	110	106	133	126	114	137
35	613	1513	113	108	137	128	118	140

Prueba	Rango time step		SVM Total Handoff		KNN Total Handoff		Neuronal Total Handoff	
			Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo	Tráfico alto	Tráfico bajo
36	631	1531	115	108	138	129	122	112
37	649	1549	117	108	141	132	123	114
38	667	1567	124	111	144	118	127	115
39	685	1585	125	109	145	118	130	123
40	703	1603	128	109	127	118	130	126
41	721	1621	88	108	128	118	105	115
42	739	1639	91	108	131	122	106	121
43	757	1657	93	138	138	123	92	123
44	775	1675	96	110	140	110	95	123
45	793	1693	96	114	143	110	95	127
46	811	1711	98	114	143	114	98	127
47	829	1729	104	117	146	115	102	131
48	847	1747	110	103	152	118	108	107
49	865	1765	113	103	157	121	111	112
50	883	1783	114	105	149	124	112	113

Como segunda parte del experimento, se realiza la simulación del modelo con una duración de transmisión de 10 minutos para los dos tipos de tráfico. En esta parte del experimento se obtienen los 6 tipos de handoff descritos anteriormente y calculados a partir del simulador: handoff totales (figura 24), handoff anticipados (figura 25), handoff fallidos (figura 26), handoff perfectos (figura 27), handoff sin interferencia (figura 28) y handoff con interferencia (figura 29).

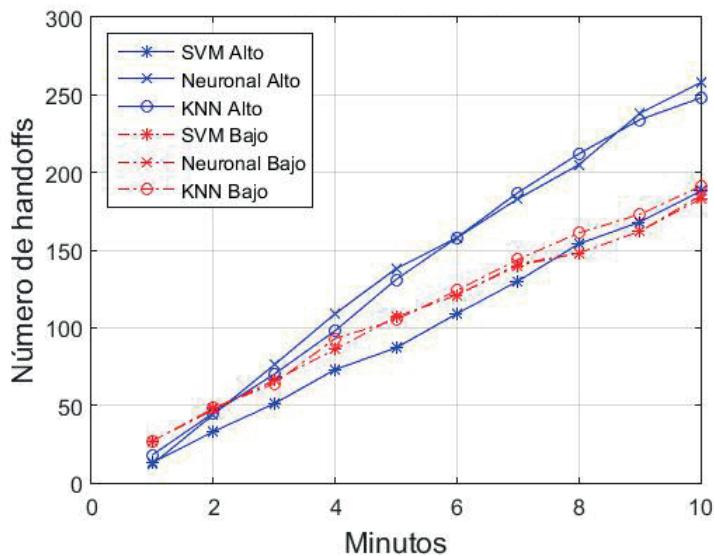


Figura 24. Handoff totales

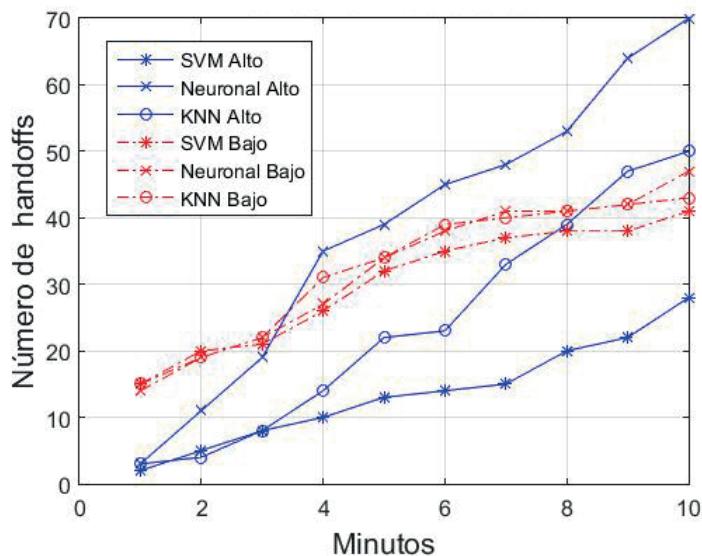


Figura 25. Handoff anticipados

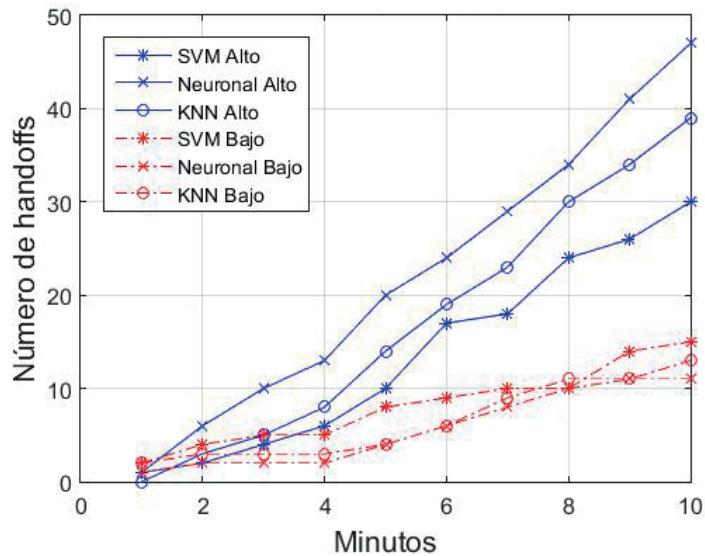


Figura 26. Handoff fallidos

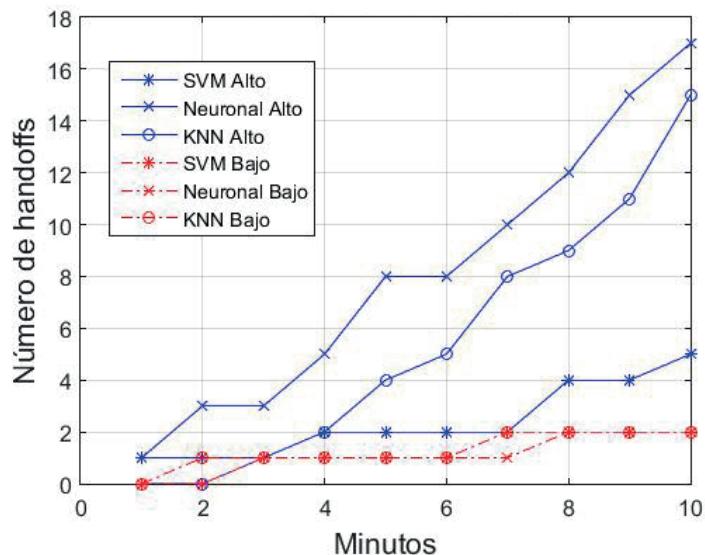


Figura 27. Handoff perfectos

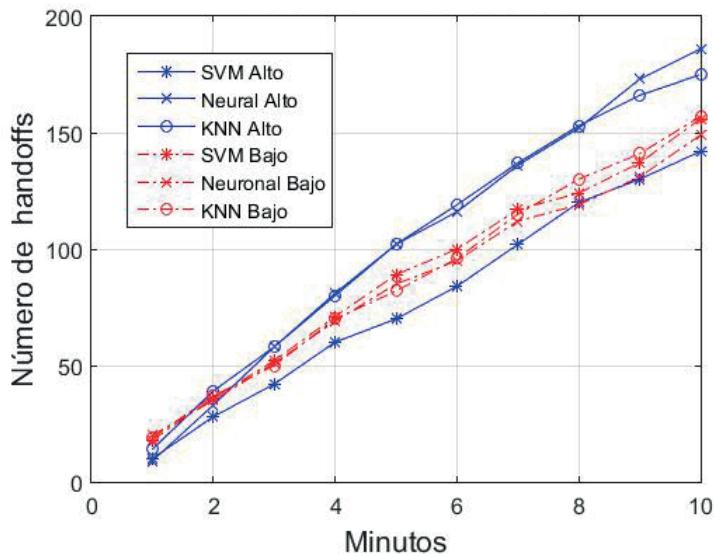


Figura 28. Handoff sin interferencia

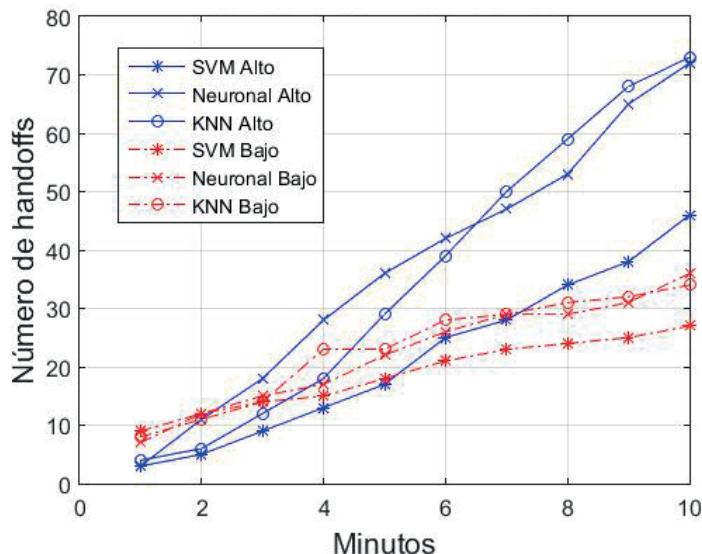


Figura 29. Handoff con interferencia

Análisis estadístico de los resultados

Las tablas 12 y 13 resumen los resultados de las figuras 24 a la 29, en términos comparativos a través de porcentajes. Vale la pena aclarar que los porcentajes están calculados para que cuanto mejor sea el resultado más se acerque al 100 %, sin importar si la métrica es de beneficio o costo.

En las tablas 12 y 13 se puede analizar los resultados obtenidos del cálculo de los 6 diferentes tipos de handoff, en los tres algoritmos evaluados, para los dos tipos de tráfico: alto y bajo. De la tabla 14 se observa que, en la ponderación total para tráfico alto, las redes neuronales obtienen un porcentaje menor del 81,57 % y KNN del 83 %, siendo las técnicas menos favorables en la implementación de este escenario en comparación con SVM, que obtuvo un porcentaje del 90,49 %. SVM presenta mejores resultados en el cálculo de handoff, aunque no acierta en los handoff perfectos (42,95 %) en comparación con redes neuronales y KNN. Por otro lado, SVM sí se comporta muy bien en los handoff anticipados 100 % y en los handoff con interferencia 100 %, en comparación con las demás técnicas.

De acuerdo con las tablas 14 y 15, SVM y KNN son las técnicas que mejor rendimiento muestran para tráfico alto y tráfico bajo. Por un lado, SVM es el algoritmo que menor handoff realiza y el que genera menos interferencia en los dos escenarios (tráfico alto y tráfico bajo), en comparación con los demás algoritmos. Pero cuando se observa la ponderación total, se puede decir que para escenarios de tráfico alto la mejor técnica es SVM, mientras que para escenarios de tráfico bajo es KNN.

Tabla 14. Porcentaje comparativo de handoff en tráfico alto

Tráfico alto	Red neuronal (%)	SVM (%)	KNN (%)
Handoff totales	82,17	100	75,18
Handoff fallido	85,45	100	77,01
Handoff con interferencia	82,82	100	91,8
Handoff sin interferencia	93,77	100	97,32
Handoff perfectos	100	42,95	86,11
Handoff anticipados	45,2	100	70,58
Ponderación total	81,57	90,49	83

Tabla 15. Porcentaje comparativo de handoff en tráfico bajo

Tráfico alto	Red neuronal (%)	SVM (%)	KNN (%)
Handoff totales	78,38	100	76,32
Handoff fallido	91,49	63,61	100
Handoff con interferencia	55,91	91,88	100
Handoff sin interferencia	96,4	100	99,45
Handoff perfectos	72,87	88,68	100
Handoff anticipados	33,09	79,88	100
Ponderación total	71,36	87,34	95,96

Conclusiones

En las diferentes mediciones realizadas en Bogotá y para distintos tipos de servicios, que involucró la banda de frecuencia GSM (824 MHz - 874 MHz), se logró evidenciar que el espectro radioeléctrico en esta región del país está siendo altamente subutilizado, lo que lleva a confirmar la necesidad de un cambio en la actual política de asignación fija del espectro, y permite la generación de regulaciones para tecnologías de acceso dinámico al espectro, como la radio cognitiva. Para el desarrollo de las mediciones se recomienda hacer uso de tiempos de barrido cortos, en las bandas de las tecnologías variantes en el tiempo como GSM.

La ejecución del modelo implementado en Matlab permite concluir que el costo de procesamiento es bastante bajo para el cálculo de parámetros de cada canal, pues obtiene el resultado en 1,6 segundos para los 551 canales, aproximadamente 3 ms por cada canal.

Al organizar los canales por prioridad, el SU puede elegir qué canales son lo menos utilizados basado en datos históricos, permitiéndole hacer uso del canal con menor probabilidad de generar interferencia y permanecer por más tiempo. La importancia de clasificar los canales reduce la cantidad de handoff requeridos por el SU para transmitir su información disminuyendo la interferencia que pueda generar a los usuarios primarios.

El algoritmo de decisión multicriterio GRA permite obtener un resultado efectivo y computacionalmente bajo para el cálculo de los parámetros que varían con el tiempo. GRA clasifica los 551 canales en 24 ms para tráfico bajo y 48 ms para tráfico alto comparado con el trabajo realizado en [50], donde GRA es ejecutado en un rango de tiempo de 25 a 50 ms para la banda de UMTS. Este rendimiento computacional permite que el modelo pueda adaptarse rápidamente con diferentes datos de entrada y pueda ser implementado en otros entornos geográficos donde exista una banda de frecuencia que esté siendo subutilizada.

Las técnicas de SVM y KNN son las más adecuadas según las pruebas realizadas, ya que permiten predecir el próximo estado de un canal específico con una probabilidad de acierto del 88% para tráfico alto y 86% para tráfico bajo, en SVM; y con un porcentaje del 86% en tráfico alto y 85% para tráfico bajo, en KNN. Por otra parte, SVM es la técnica que demuestra mejor comportamiento en los cambios de canal ya que disminuye la interferencia al realizar un menor número de handoff y realizar mayores cambios de canal sin afectar el tráfico del usuario primario en los dos escenarios. KNN puede mostrar mejor rendimiento en tráfico bajo y ser más asertivo en los handoff perfectos, pero para tráfico alto no tiene buen desempeño, lo que muestra

irregularidad. En términos computacionales el algoritmo KNN y redes neuronales son más eficaces comparados con SVM, y esto permite concluir que no hay un método de predicción perfecto para todo tipo de escenario o ambiente de simulación.

Trabajo futuro

Para un trabajo futuro se propone evaluar otras técnicas de predicción que permitan mejorar el rendimiento computacional del modelo propuesto que mejore la probabilidad de acierto obtenida en este modelo.

Es necesario realizar experimentos en redes de radio cognitiva con datos de otras bandas de frecuencia implementadas actualmente como UMTS y LTE, que permitan evidenciar ajustes en el modelo y mejorar su implementación en el campo real.

Implementar el modelo diseñado en un escenario de red móvil utilizando más de un usuario secundario en la interacción permitirá observar si las decisiones tomadas por los SU conllevan a un comportamiento inteligente de autoorganización o si esta interacción entre usuarios tiende a comportarse al borde del caos, para ello un trabajo futuro será modelar la actividad de los SU en el espectro.

Este modelo no tiene en cuenta la variable espacio y ubicación de los diferentes usuarios secundarios en una red inalámbrica, lo cual es necesario en una topología de red distribuida donde el usuario más cercano pueda ser localizado y asociarse para realizar una comunicación. Podría ser interesante incluir otros parámetros en el desarrollo de este modelo como el ancho de banda o la regulación de la potencia de transmisión lo cual será una propuesta futura.

Referencias

- [1] E. Tragos, S. Zeadally, A. Fragkiadakis y V. Siris, “Spectrum assignment in cognitive radio networks: a comprehensive survey”, *IEEE Commun. Surv. Tutorials*, vol. 15, n.º 3, pp. 1108-1135, 2013.
- [2] N. Abbas, Y. Nasser y K. El Ahmad, “Recent advances on artificial intelligence and learning techniques in cognitive radio networks”, *EURASIP J. Wirel. Commun. Netw.*, vol. 174, 2011. [En línea] Disponible en: <https://doi.org/10.1186/s13638-015-0381-7>.
- [3] M. T. Masonta, M. Mzyece y N. Ntlatlapa, “spectrum decision in cognitive radio networks: a survey”, *IEEE Commun. Surv. Tutorials*, vol. 15, n.º 3, pp. 1088-1107, 2013.
- [4] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran y S. Mohanty, “A survey on Spectrum management in cognitive radio networks”, *Commun. Mag. IEEE*, vol. 46, n.º 4, pp. 40-48, 2008.
- [5] C. Hernández, *Modelo adaptativo de handoff espectral para la mejora en el desempeño de la movilidad en redes móviles de radio cognitiva*, Bogotá: Universidad Nacional de Colombia, 2017.
- [6] M. Lahby, S. Baghla y A. Sekkaki, “Survey and comparison of MADM methods for network selection access in heterogeneous networks”, en *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, París, Francia, 2015.
- [7] J. Mitola y G. Q. Maguire, “Cognitive radio: making software radios more personal”, *IEEE Pers. Commun.*, vol. 6, n.º 4, pp. 13-18, 1999.
- [8] M. Delgado y B. Rodríguez, “Opportunities for a more efficient use of the spectrum based in cognitive radio”, *IEEE Lat. Am. Trans.*, vol. 14, n.º 2, pp. 610-616, 2016.
- [9] I. F. Akyildiz, W.-Y. Lee y K. R. Chowdhury, “Crahns: Cognitive Radio Ad Hoc Networks”, *J. Ad Hoc Networks*, vol. 7, pp. 810-836, 2009.
- [10] S. Ju y J. B. Evans, “Scalable cognitive routing protocol for mobile ad-hoc networks”, en *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, Estados Unidos, 2010, pp. 1-6.

- [11] J. Marinho y E. Monteiro, "Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions", *Wirel. Networks*, vol. 18, n.º 2, pp. 147-164, 2012.
- [12] Y. Chen, Q. Zhao y A. Swami, "Distributed spectrum sensing and access in cognitive radio networks with energy constraint", *IEEE Trans. Signal Process.*, vol. 57, n.º 2, pp. 783-797, 2009.
- [13] P. Ren, Y. Wang, Q. Du y J. Xu, "A survey on dynamic spectrum access protocols for distributed cognitive wireless networks", *EURASIP J. Wirel. Commun. Netw.*, vol. 2012, n.º 1, p. 60, 2012.
- [14] E. Trigui, M. Esseghir y L. M. Boulahia, "Cognitive radio spectrum assignment and handoff decision", en *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Londres, Reino Unido, 2013, pp. 2881-2886.
- [15] J. Wang, M. Ghosh y K. Challapali, "Emerging cognitive radio applications: A survey", *IEEE Commun. Mag.*, vol. 49, n.º 3, pp. 74-81, 2011.
- [16] W.-Y. L. I. F. Akyildiz, "A spectrum decision framework for cognitive radio networks", *IEEE Trans. Mob. Comput.*, vol. 10, n.º 2, 2011.
- [17] *Standard for Wireless Regional Area Networks (WRAN)—Specific Requirements—Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands*, The Institute of Electrical and Electronics Engineering, IEEE Standard 802.22, 2011.
- [18] M. Amir, A. El-Keyi y M. Nafie, "Constrained interference alignment and the spatial degrees of freedom of mimo cognitive networks", *IEEE Trans. Inf. Theory*, vol. 57, n.º 5, pp. 2994-3004, 2011.
- [19] I. F. Akyildiz, W. Y. Lee, M. C. Vuran y S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey", *Comput. Networks*, vol. 50, n.º 13, pp. 2127-2159, 2006.
- [20] M. Ozger y O. B. Akan, "On the utilization of spectrum opportunity in cognitive radio networks", *IEEE Commun. Lett.*, vol. 20, n.º 1, pp. 157-160, 2016.
- [21] A. Azarfara, J.-F. Frigon y B. Sanso, "Improving the reliability of wireless networks using cognitive radios", *IEEE Commun. Surv. Tutorials*, vol. 14, n.º 2, pp. 338-354, 2012.
- [22] C. Devanarayana y A. S. Alfa, "Predictive channel access in cognitive radio networks based on variable order Markov models", en *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Kathmandu, Nepal, 2011, pp. 1-6.
- [23] C. Devanarayana y A. S. Alfa, "Proactive channel access in cognitive radio networks based on users statistics", en *2014 1st International Workshop on Cognitive Cellular Systems (CCS)*, Alemania, 2014.

- [24] R. Aguilar-González *et al.*, “Performance of MADM algorithms with real spectrum measurements for spectrum decision in cognitive radio networks”, en *2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Campeche, México, 2014.
- [25] S. Pandit y G. Singh, “Spectrum sharing in cognitive radio using game theory”, en *2013 3rd IEEE International Advance Computing Conference (IACC)*, Ghaziabad, India, 2012, pp. 1503-1506.
- [26] Y. Wu, F. Hu, S. Kumar *et al.*, “Apprenticeship learning based spectrum decision in multi-channel wireless mesh networks with multi-beam antennas”, *IEEE T Mobile Comput*, vol. 16, n.º 2, pp. 314-325, 2017.
- [27] I. Akbar y W. Tranter, “Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case”, en *Proceedings 2007 IEEE SoutheastCon*, Richmond, VA, Estados Unidos, 2007, pp. 196-201.
- [28] P. S. Aizaz Zainab, “A survey of cognitive radio reconfigurable antenna design and proposed design using genetic algorithm”, en *2016 IEEE Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2016.
- [29] M. Matinmikko, J. Del Ser, T. Rauma y M. Mustonen, “Fuzzy-logic based framework for spectrum availability assessment in cognitive radio systems”, *IEEE J. Sel. Areas Commun.*, vol. 31, n.º 11, pp. 2173-2184, 2013.
- [30] S. M. S, S. B. Mafra, G. S. Member, E. M. G. Fernández *et al.*, “Power control and relay selection in cognitive radio ad hoc networks using game theory”, *IEEE Syst J*, vol. 12, n.º 3, pp. 1-12, 2016.
- [31] F. Cai, Y. Gao, L. Cheng *et al.*, “Spectrum sharing for LTE and WiFi coexistence using decision tree and game theory”, en *2016 IEEE Wireless Communications and Networking Conference*, Doha, Qatar, 2016.
- [32] Y. Xu, A. Anpalagan, Q. Wu *et al.*, “Decision- theoretic distributed channel selection for opportunistic spectrum access: Strategies, challenges and solutions”, *IEEE Commun. Surv. Tutorials*, vol. 15, n.º 4, pp. 1689-1713, 2013.
- [33] X. Tan, H. Huang y L. Ma, “Frequency allocation with Artificial Neural Networks in cognitive radio system”, en *IEEE 2013 Tencon - Spring*, Sydney, NSW, Australia, 2013, pp. 366-370.
- [34] L. F. Pedraza, C. Hernández, K. Galeano, E. Rodríguez-Colina *et al.*, *Ocupación espectral y modelo de radio cognitiva para Bogotá*, Bogotá: Editorial UD, 2016.
- [35] Y. Zhao, Z. Hong, Y. Luo, et al., “Prediction-Based Spectrum Management in Cognitive Radio Networks”, *IEEE Syst J*, vol. 12, n.º 4, pp. 3303-3314, dic. 2018.
- [36] X. Song, W. Liu, M. Zhang *et al.*, “A network selection algorithm based on FAHP/GRA in heterogeneous wireless networks”, en *2016 2nd IEEE International Conference on Computer and Communications (ICCC)* Chengdu, China, 2016, pp. 1445-1449.

- [37] M. Lahby y A. Adib, “Network selection mechanism by using M-AHP/GRA for heterogeneous networks”, en *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, Dubai, Emiratos Árabes Unidos, 2013, pp. 1-6.
- [38] M. Mansouri y C. Leghris, “A comparison between fuzzy TOPSIS and fuzzy GRA for the vertical handover decision making”, en *2017 Intelligent Systems and Computer Vision (ISCV)*, Fez, Marruecos, 2017, pp. 1-6.
- [39] G. Ding *et al.*, “Spectrum inference in cognitive radio networks: algorithms and applications”, *IEEE Commun. Surv. Tutorials*, vol. 20, n.º 1, pp. 150-182, 2017.
- [40] N. Gupta, S. K. Dhurandher y I. Woungang, “On the probability of appearance of primary user in IEEE 802 . 22 WRAN using an artificial neural network learning technique”, en *2016 1st India International Conference on Information Processing (IICIP)*, Delhi, 2016, pp. 1-5.
- [41] M. Huk y J. Mizera-Pietraszko, “Contextual neural-network based spectrum prediction for cognitive radio”, en *2015 Fourth International Conference on Future Generation Communication Technology (FGCT)*, Luton, Reino Unido, 2015.
- [42] J. Guo, H. Ji, Y. Li y X. Li, “A novel spectrum handoff management scheme based on SVM in cognitive radio networks”, en *A Novel Spectrum Handoff Management Scheme based on SVM in Cognitive Radio Networks*, Harbin, China, 2011, pp. 645-649.
- [43] A. Agarwal, S. Dubey, M. A. Khan *et al.*, “Learning based primary user activity prediction in cognitive radio networks for efficient dynamic spectrum access”, en *2016 1st India International Conference on Information Processing (IICIP)*, Delhi, 2016, pp. 1-5.
- [44] M. Kyryk, N. Pleskanka y V. Yanyshyn, “Performance evaluation model for spectrum decision methods in cognitive radio”, en *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, Lviv, Ucrania, 2017, pp. 289-291.
- [45] Y. Liu, R. Yu y M. Pan, “SD-MAC : Spectrum Database-Driven MAC Protocol for Cognitive Machine-to-Machine Networks”, *IEEE T Veh Technol*, vol. 66, n.º 2, pp. 1456–1467, 2017.
- [46] L. Wang, J. Yang, and X. Song, “A QoE-Driven Spectrum Decision Scheme for Multimedia Transmissions over Cognitive Radio Networks,” en *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, Vancouver, BC, Canadá, 2017.
- [47] A. Surampudi, K. Kalimuthu y B. Tech, “An adaptive decision threshold scheme for the matched filter method of spectrum sensing in cognitive radio using artificial neural networks”, en *2016 1st India International Conference on Information Processing (IICIP)*, Delhi, 2016, pp. 1-5.

- [48] F. Liu, J. Ma, R. Du y J. Wu, “ICSGC-based dynamic spectrum access algorithm for cognitive radio”, en *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 5692-5697.
- [49] C. Hernández, I. Páez y D. Giral, “Modelo AHP-VIKOR para handoff espectral en redes de radio cognitiva”, vol. 19, n.º 45, pp. 29-39, 2015.
- [50] A. F. Almutairi, “Weighting selection in GRA-based MADM for vertical handover in wireless networks”, en *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim)*, Cambridge, 2016, pp. 331-336.
- [51] S. Iliya, E. Goodyer, J. Gow et al., “Application of artificial neural network and support vector regression in cognitive radio networks for RF power prediction using compact differential evolution algorithm”, en *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Lodz, Polonia, 2015, pp. 55-56.

Anexos

Desarrollo de software simulador en matlab.

A. Algoritmo general de decisión de espectro

```
clc;clear ; close all;
load('C:\Users\rokli\OneDrive\Documentos\Universidad\Matlab\Machine
Learning\Trazas de trafico GSM\Trafico_GSM_trazas_entrenamiento -95.mat');
load('C:\Users\rokli\OneDrive\Documentos\Universidad\Matlab\Machine
Learning\Trazas de trafico GSM\Trafico_GSM_trazas_evaluacion -95.mat');
High= Power_t.High;
EvaHigh=Power_e.High;
EvaLow=Power_e.Low;
Low=Power_t.Low;
[Mev,Nev]=size(EvaHigh);
[Mevl,Nevl]=size(EvaLow);
[Mh,Nh]=size(High);
[Ml,Nl]=size(Low);
datosHigh= High((1:Mh),(1:Nh));
datosLow=Low((1:Ml),(1:Nl));
Clasificacion= zeros(1,Nh);
Evaluacion= zeros(1,Nh);
ValorEs = zeros(1,Nh);
Matriz= zeros(Nh,3);
banda= 824:0.1:879;
banda=banda';
canaleslibres=zeros(551,3);
canalesocupados=zeros(551,3);
canal1=zeros(551,1);
salidaE=zeros(Mev,551);
salidane=zeros(Mev,551);
ocu=zeros(551,2);
ener=zeros(551,2);
grasnr=zeros(551,2);
%% calculo de los criterios
[Procupacion]=ciclotrabajo (datosHigh,datosLow);
[trazaden,trazaenergia]=energia(datosHigh,datosLow);
```

```
[snrdb]=SNR(High,Nh,Mh,Low,Ml,Nl);
% Graficas Criterios
ocu(:,1)=banda(:,1);
ocu(:,2)=Procupacion(:,2)*100;
figure
plot(ocu(:,1),ocu(:,2));
grid on
grid minor
title('CICLO DE TRABAJO ','fontsize',16);
ylabel('Porcentaje de Ocupación (%)','fontsize',16);
xlabel('Frencuencia (MHz)','fontsize',16);
axis([820 880 0 110])
set(gca,'fontsize',12)
figure
ener(:,1)=banda(:,1);
ener(:,2)=trazanergia(:,1);
plot(ener(:,1),ener(:,2));
set(gca,'fontsize',12)
grid on
grid minor
title(' INDICE DE POTENCIA ','fontsize',16);
ylabel('Nivel de potencia','fontsize',16);
xlabel('Frencuencia (MHz)','fontsize',16);
figure
grasnr(:,1)=banda(:,1);
grasnr(:,2)=snrdb(:,1);
plot(grasnr(:,1),grasnr(:,2));
set(gca,'fontsize',12)
grid on
grid minor
title(' SNR ','fontsize',16);
ylabel(' Nivel SNR (dBm)','fontsize',16);
xlabel('Frencuencia (MHz)','fontsize',16);
%% Matriz de criterios
MatrizHigh=criteriosHigh(Procupacion,trazanergia,snrdb);
MatrizLow=criteriosLow(Procupacion,trazanergia,snrdb);
%% Clasificacion
[mejorcanalHigh]= GRA(MatrizHigh);
[mejorcanalLow]= GRAlow(MatrizLow);
%% Prediccion red neuronal y SVM
[preneuronalH,preneuronalL,presvmH,presvmL,mcanalH,preKNNH,preKNNL]= neur
oprediccion(datosHigh,datosLow,mejorcanalHigh,mejorcanalLow);
%% Calculo del error para SVM y red neuronal
[errorsvmH,errorneuralH,errorknnH,errorH,mcaH,promedioH]
=errorpre(EvaHigh,mejorcanalHigh,banda,presvmH,preneuronalH,preKNNH);
[errorsvmL,errorneuralL,errorknnL,errorL,mcaL,promedioL]
```

```

=errorprelow(EvaLow,mejorcanalLow,banda,presvML,preneuronalL,preKNNL);
%% experimento sin predicción
[totalalto]=experimentoalto(mcaH,EvaHigh);
[totalbajo]=experimentobajo(mcaL,EvaLow);
%% interacción High
[totalhand, totalfallido, totalanti, totalinte, totalperfecto]=handoffsvm(pre
svmH,mcaH,EvaHigh);
[totalhandn, totalfallidon, totalantin, totalinten, totalperfecton]=handoffne
uronan(preneuronalH,mcaH,EvaHigh);
[totalhandk, totalfallidok, totalantik, totalintek, totalperfectok]=handoffKN
N(preKNNH,mcaH,EvaHigh);
totalsininte=totalhand-totalinte;
totalsininten=totalhandn-totalinten;
totalsinintek=totalhandk-totalintek;
%% interacción Low
[totalhandL, totalfallidoL, totalantiL, totalintel, totalperfectoL]=handoffsv
mL(presvML,mcaL,EvaLow);
[totalhandnL, totalfallidonL, totalantinL, totalintenL, totalperfectonL]=hand
offneuronanL(preneuronalL,mcaL,EvaLow);
[totalhandkL, totalfallidokL, totalantikL, totalintekL, totalperfectokL]=hand
offKNNL(preKNNL,mcaL,EvaLow);
totalsininteL=totalhandL-totalinteL;
totalsinintenL=totalhandnL-totalintenL;
totalsinintekL=totalhandkL-totalintekL;
%% Grafica Experimento
plot(totalalto,'b-*')
hold on
grid on
title('Handoffs Totales Experimento tráfico alto','fontsize',14)
set(gca,'fontsize',12)
ylabel('Número de handoffs','fontsize',14)
xlabel('Minutos','fontsize',14)
figure
plot(totalbajo,'b-*')
hold on
grid on
title('Handoffs Totales Experimento tráfico bajo','fontsize',14)
set(gca,'fontsize',12)
ylabel('Número de handoffs','fontsize',14)
xlabel('Minutos','fontsize',14)
figure
%% graficas Handoff
plot(totalhand,'b-*')
hold on
plot(totalhandn,'r-x')
grid on

```

```
plot(totalhandk, 'g-o')
plot(totalhandL, 'k-+')
plot(totalhandnL, 'c-d')
plot(totalhandkL, 'm-p')
title('Handoffs Totales', 'fontsize', 16)
set(gca, 'fontsize', 12)
ylabel('Número de handoffs', 'fontsize', 16)
xlabel('Minutos', 'fontsize', 16)
legend('SVM Alto', 'Neuronal Alto', 'KNN Alto', 'SVM Bajo', 'Neuronal Bajo', 'KNN Bajo')
figure
plot(totalfallido, 'b-*')
hold on
plot(totalfallidon, 'r-x')
grid on
plot(totalfallidok, 'g-o')
plot(totalfallidol, 'k-+')
plot(totalfallidonL, 'c-d')
plot(totalfallidokL, 'm-p')
title('Handoffs Fallidos', 'fontsize', 16)
set(gca, 'fontsize', 12)
ylabel('Número de handoffs', 'fontsize', 16)
xlabel('Minutos', 'fontsize', 16)
legend('SVM Alto', 'Neuronal Alto', 'KNN Alto', 'SVM Bajo', 'Neuronal Bajo', 'KNN Bajo')
figure
plot(totalinte, 'b-*')
hold on
plot(totalinten, 'r-x')
grid on
plot(totalintek, 'g-o')
plot(totalintel, 'k-+')
plot(totalintenL, 'c-d')
plot(totalintekL, 'm-p')
title('Handoffs con interferencia', 'fontsize', 16)
set(gca, 'fontsize', 12)
ylabel('Número de handoffs', 'fontsize', 16)
xlabel('Minutos', 'fontsize', 16)
legend('SVM Alto', 'Neuronal Alto', 'KNN Alto', 'SVM Bajo', 'Neuronal Bajo', 'KNN Bajo')
figure
plot(totalsininte, 'b-*')
hold on
plot(totalsininten, 'r-x')
grid on
plot(totalsinintek, 'g-o')
```

```
plot(totalsininteL,'k-+')
plot(totalsinintenL,'c-d')
plot(totalsinintekL,'m-p')
title('Handoffs sin interferencia','fontsize',16)
set(gca,'fontsize',12)
ylabel('Número de handoffs','fontsize',16)
xlabel('Minutos','fontsize',16)
legend('SVM Alto','Neural Alto','KNN Alto','SVM Bajo','Neuronal
Bajo','KNN Bajo')
figure
plot(totalperfecto,'b-*')
hold on
plot(totalperfecton,'r-x')
grid on
plot(totalperfectok,'g-o')
plot(totalperfectoL,'k-+')
plot(totalperfectonL,'c-d')
plot(totalperfectokL,'m-p')
title('Handoffs Perfectos','fontsize',16)
set(gca,'fontsize',12)
ylabel('Número de handoffs','fontsize',16)
xlabel('Minutos','fontsize',16)
legend('SVM Alto','Neuronal Alto','KNN Alto','SVM Bajo','Neuronal
Bajo','KNN Bajo')
figure
plot(totalanti,'b-*')
hold on
plot(totalantin,'r-x')
grid on
plot(totalantik,'g-o')
plot(totalantiL,'k-+')
plot(totalantinL,'c-d')
plot(totalantikL,'m-p')
title('Handoffs Anticipados','fontsize',16)
set(gca,'fontsize',12)
ylabel('Número de handoffs','fontsize',16)
xlabel('Minutos','fontsize',16)
legend('SVM Alto','Neuronal Alto','KNN Alto','SVM Bajo','Neuronal
Bajo','KNN Bajo')
figure
```

B. Cálculo de los parámetros

Ocupación de espectro

```
function [Procupacion]=ciclotrabajo (datosHigh,datosLow)
%% Entrenamiento
ocupacionH= lt(-95,datosHigh);
ocupacionH=ocupacionH';
ocupacionL= lt(-95,datosLow);
ocupacionL=ocupacionL';
[Fh,Ch]=size(ocupacionH);
[Fl,C1]=size(ocupacionL);
TOFFh=zeros(Fh,2);
TONh =zeros(Fh,2);
TOFFl=zeros(Fl,2);
TONl =zeros(Fl,2);
intensidad= zeros((Fh),3);
Procupacion= zeros((Fh),3);
Prolibre= zeros((Fh),3);
Total= Ch*0.290;
% High
for i=1:Fh
    libreh=0;
    ocupadoh=0;
    for j=1:Ch
        if ocupacionH (i,j)== 0
            libreh=libreh+0.290;
            TOFFh(i,2)=libreh;
            TONh (i,1)=i;
        else
            ocupadoh=ocupadoh+0.290;
            TONh (i,2)= ocupadoh;
            TONh(i,1)=i;
        end
    end
    intensidad(i,2)= TONh(i,2)/(TONh(i,2)+TOFFh(i,2));
    intensidad(i,1)= i;
    Procupacion(i,2)= TONh(i,2)/Total;
    Procupacion(i,1)= i;
    Prolibre(i,2)= 1-Procupacion(i,2);
    Prolibre(i,1)=i;
end

%% Low
for i=1:Fl
    librel=0;
```

```

ocupadol=0;
for j=1:C1
    if ocupacionL (i,j)== 0
        libreL=libreL+0.290;
        TOFFL(i,2)=libreL;
        TOFFL(i,1)=i;
    else
        ocupadol=ocupadol+0.290;
        TONL(i,2)= ocupadol;
        TONL(i,1)=i;
    end
    intensidad(i,3)= TONL(i,2)/(TONL(i,2)+TOFFL(i,2));
    Procupacion(i,3)= TONL(i,2)/Total;
    Prolibre(i,3)= 1-Procupacion(i,3);
    end
end

```

Índice de potencia

```

function [trazaden, trazaenergia] = energia(datosHigh, datosLow)
%% Índice de potencia en dBm datos de entrenamiento
Emaxh=max(datosHigh); Eminh=min(datosHigh);
Eminl=min(datosLow); Emaxl=max(datosLow);
Energiahight = Eminh./Emaxh; Energiahight=Energiahight';
Energialow = Eminl./Emaxl; Energialow=Energialow';
[~,c]=size(datosHigh);
trazaenergia=zeros(c,2);
trazaenergia(:,1)=Energiahight;
trazaenergia(:,2)=Energialow;

%% Índice de potencia en mV datos de entrenamiento
denHigh=10.^ (datosHigh./10);
denLow=10.^ (datosLow./10);
trazaden=zeros(c,2);
Dmaxh=max(denHigh); Dminh=min(denHigh);
Dminl=min(denLow); Dmaxl=max(denLow);
Densidadhigh = Dminh./Dmaxh; Densidadhigh=Densidadhigh';
Densidadlow=Dminl./Dmaxl; Densidadlow=Densidadlow';
trazaden(:,1)=Densidadhigh;
trazaden(:,2)=Densidadlow;
end

```

Relación señal a ruido

```
function [snrdb]=SNR(High,Nh,Mh,Low,Ml,Nl)
%% Calculo del ruido termico
snrdH=zeros(Mh,Nh);
snrdL=zeros(Ml,Nl);
potencial=-100;
snrdb=zeros(Nl,2);
for i=1:Nh
    snrdH(:,i)=High(:,i)-(potencial);
    snrdb(i,1)=sum(snrdH(:,i))/Mh;
    snrdL(:,i)=Low(:,i)-(potencial);
    snrdb(i,2)=sum(snrdL(:,i))/Mh;
end
```

C. Algoritmo GRA

```
function[mejorcanalHigh]= GRA(MatrizHigh)
[fi,co]=size(MatrizHigh);
r=zeros(fi,co);
p=zeros(fi,co);
v=zeros(fi,co);
deltai=zeros(fi,co);
S=zeros(fi,2);
CRC=zeros(fi,3);
salida=zeros(fi,2);
mejorcanalHigh=zeros(fi,3);
Mo=max(MatrizHigh(:,1));
Me=max(MatrizHigh(:,2));
Ms=max(MatrizHigh(:,3));
Mino=min(MatrizHigh(:,1));
Mine=min(MatrizHigh(:,2));
Mins=min(MatrizHigh(:,3));
%% Normalizar la matriz de decision de los tres atributos de beneficio
for j=1:co
    for i=1:fi
        if j == 1
            r(i,j) = (Mo-MatrizHigh(i,j))/(Mo-Mino);
        elseif j ==2
            r(i,j) = (Me-MatrizHigh(i,j))/(Me-Mine);
        else
            r(i,j) = (Ms-MatrizHigh(i,j))/(Ms-Mins);
        end
    end
end
```

```

%% Normalizado Ponderado
% Se asignan los pesos de los atributos de la siguiente manera:
% Para ocupacion se elige una prioridad de 7; Energia 5 y SNR 3; La
sumatoria es igual a 1
% pesos=zeros(1,3);
PeOc=7;
PeSinr=5;
PePot=3;
raiz=zeros(3,3);
pesos=zeros(3,3);
% [1 5 7]
% [1/5 1 3]
% [1/7 1/3 1]
% raizOc= 1*PeSinr*PeOc;
% raizSi=1/PeSinr*1*PePot;
% raizPo=1/PeOc*1/PePot*1;
% wo1=nthroot(raizOc,3);
% we=nthroot(raizSi,3);
% ws=nthroot(raizPo,3);
% pesos(1,1)=wo1;
% pesos(1,2)=we;
% pesos(1,3)=ws;
% maxpe=max(pesos);
% minpe=min(pesos);
% wo= (pesos(1,1)-minpe)/(maxpe - minpe);
% we= (pesos(1,2)-minpe)/(maxpe - minpe);
% ws= (pesos(1,3)-minpe)/(maxpe - minpe);
raiz(1,:)=[1      PeSinr      PeOc];
raiz(2,:)=[1/PeSinr    1      PePot];
raiz(3,:)=[1/PeOc    1/PePot      1];

wo1=max(cumsum(raiz(:,1)));
we1=max(cumsum(raiz(:,2)));
ws1=max(cumsum(raiz(:,3)));
pesos(1,1)=raiz(1,1)/wo1;
pesos(2,1)=raiz(2,1)/wo1;
pesos(3,1)=raiz(3,1)/wo1;
pesos(1,2)=raiz(1,2)/we1;
pesos(2,2)=raiz(2,2)/we1;
pesos(3,2)=raiz(3,2)/we1;
pesos(1,3)=raiz(1,3)/ws1;
pesos(2,3)=raiz(2,3)/ws1;
pesos(3,3)=raiz(3,3)/ws1;

wo=max(cumsum(pesos(1,:)))/3;
we=max(cumsum(pesos(2,:)))/3;

```

```
ws=max(cumsum(pesos(3,:))/3;
% wo=wo1/wo1
% we=we/wo1
P2 = wo + we + ws
% ws=ws/wo1;
v(:,1)= r(:,1).*wo;
v(:,2)= r(:,2).*we;
v(:,3)= r(:,3).*ws;

%% criterio de beneficio
x1=max(v(:,1));
x2=max(v(:,2));
x3=max(v(:,3));
%% Calculo GRC
for j=1:co
    for i=1: fi
        if j==1
            deltai(i,1)= x1-v(i,j);
        elseif j==2
            deltai(i,2)= x2-v(i,j);
        else
            deltai(i,3)= x3-v(i,j);
        end
    end
end
maxD1=max(deltai(:,1));
maxD2=max(deltai(:,2));
maxD3=max(deltai(:,3));
minD1=min(deltai(:,1));
minD2=min(deltai(:,2));
minD3=min(deltai(:,3));

for j=1:co
    for i=1:fi
        if j==1
            CRC (i,j) = ((minD1+maxD1) /(deltai(i,j)+ maxD1))/co;
        elseif j==2
            CRC (i,j) = ((minD2+maxD2) /(deltai(i,j)+ maxD2))/co;
        else
            CRC (i,j) = ((minD3+maxD3) /(deltai(i,j)+ maxD3))/co;
        end
    end
end

%% Salida
banda= 824:0.1:879;
```

```

banda=banda';
S(:,1)=banda(:,1);
S(:,2)=v(:,1)+v(:,2)+v(:,3);
salida(:,2)=CRC(:,1)+ CRC(:,2)+CRC(:,3);
salida(:,1)=banda(:,1);

%% Organiza los mejores canales para la predicción
for i=1:f
    [maxi,j]=max(salida(:,2));
    mejorcanalHigh(i,2)=salida(j,1);
    mejorcanalHigh(i,3)=maxi;
    salida(j,1:2)=0;
    prioridad=1:1:551;
    prioridad=prioridad';
    mejorcanalHigh(:,1)=prioridad;
end

```

D. Algoritmo de predicción

```

function
[preneuronalH, preneuronalL, presvmH, presvmL, mcanalH, preKNNH, preKNNL]= neur
oprediccion(datosHigh, datosLow, mejorcanalHigh, mejorcanalLow)
% load('C:\Users\rokli\OneDrive\Documentos\Universidad\Matlab\Machine
Learning\Trazas de trafico GSM\Trafico_GSM_trazas_evaluacion -95.mat');
% High= Power_e.Low;
[f1,~]=size(datosHigh);
% [f1,~]=size(High);
band= 824:0.1:879;
mcanalH=zeros(1,551);
mcanalL=zeros(1,551);
claseh=zeros(f1,2);
clasel=zeros(f1,2);
clasek=zeros(f1,2);
presvmH=zeros(f1,551);
presvmL=zeros(f1,551);
preKNNH=zeros(f1,551);
preKNNL=zeros(f1,551);
datosentradaH=(lt(-95,datosHigh));
datosentradaL=(lt(-95,datosLow));
[Y,PS]=mapminmax(datosentradaH);
[Y1,PS1]=mapminmax(datosentradaL);
canalentrenamientoL=zeros(f1,551);
preneuronalL=zeros(551,f1);
aprioriL=zeros(f1,551);
apriori1L=zeros(f1,551);

```

```
apriori1H=zeros(f1,551);
aprioriH=zeros(f1,551);
% High
for i=1:551
    primer=mejorcanalHigh(i,2);
    [~,c]=find(primer==band(1,:));
    mcanalH(1,i)=c;
end
% Low
for i=1:551
    primerL=mejorcanalLow(i,2);
    [~,c]=find(primerL==band(1,:));
    mcanalL(1,i)=c;
end

%% implementacion SVM y KNN HIGH

for i=1:551
aprioriH(:,i)= lt(-95,datosHigh(:,mcanalH(1,i)));
claseh(:,1)=datosHigh(:,mcanalH(1,i));
claseh(:,2)=aprioriH(:,i);
claseh(:,2)=(aprioriH(:,i).*2-1);
SVMModelH = fitcsvm(claseh(:,1), claseh(:,2));
[presvmH(:,i),~] = predict(SVMModelH,datosHigh(:,i));
presvmH(:,i)=((presvmH(:,i)+ 1)./2);
apriori1H(:,i)= lt(-95,datosHigh(:,mcanalH(1,i)));
clasek(:,2)=apriori1H(:,i);
clasek(:,1)=datosHigh(:,mcanalH(1,i));
KNNModel=fitcknn(clasek(:,1),clasek(:,2),'NumNeighbors',3,'NSMethod','exhaustive','Distance','minkowski','Standardize',1);
[preKNNH(:,i),~] = predict(KNNModel,datosHigh(:,i));
end

% LOW SVM y KNN

for i=1:551
aprioriL(:,i)= lt(-95,datosLow(:,mcanalL(1,i)));
clasel(:,1)=datosLow(:,mcanalL(1,i));
clasel(:,2)=aprioriL(:,i);
clasel(:,2)=(aprioriL(:,i).*2-1);
SVMModelL = fitcsvm(clasel(:,1), clasel(:,2));
[presvmL(:,i),~] = predict(SVMModelL,datosLow(:,i));
presvmL(:,i)=((presvmL(:,i)+ 1)./2);
apriori1L(:,i)= lt(-95,datosLow(:,mcanalL(1,i)));

```

```

clasekL(:,2)=apriori1L(:,i);
clasekL(:,1)=datosLow(:,mcanalL(1,i));
KNNModelL=fitcknn(clasekL(:,1),clasekL(:,2),'NumNeighbors',3,'NSMethod','e
xhaustive','Distance','minkowski','Standardize',1);
[preKNNL(:,i),~] = predict(KNNModelL,datosLow(:,i));

end

%% neuronal High
canalentrenamientoH=zeros(fil,551);
preneuronalH=zeros(551,fil);
aprioriH=zeros(fil,551);
for i=1:551
    canalentrenamientoH(:,i)=Y(:,mcanalH(1,i));
    x1=canalentrenamientoH(:,i);
    x1=x1';
    xi1=[-1 1];
    [y1,~] = normal1(x1,xi1);
    preneuronalH(i,:)=y1;
end
preneuronalH=preneuronalH';
preneuronalH=mapminmax('reverse',preneuronalH,PS);
%% low

% for i=1:551
%     canalentrenamientoL(:,i)=Yl(:,mcanalL(1,i));
%     x1=canalentrenamientoL(:,i);
%     x1=x1';
%     xi1=[-1 1];
%     [y1,~] = normallow(x1,xi1);
%     preneuronalL(i,:)=y1;
% end
for i=1:551
    canalentrenamientoL(:,i)=Yl(:,mcanalL(1,i));
    x1=canalentrenamientoL(:,i);
    x1=x1';
    xi1=[-1 1];
    [y1,~] = normal1(x1,xi1);
    preneuronalL(i,:)=y1;
end
preneuronalL=preneuronalL';
preneuronalL=mapminmax('reverse',preneuronalL,PSl);

end

```

E. Simulador

```
function [totalhand,totalfallido,totalanti,totalinte,totalperfecto]=hando
ffsvm(presvmH,mcaH,EvaHigh)
Matrizinteraccion=EvaHigh; % transpuesta de la matriz canal de
frecuencia(filas) tiempo columnas
Matrizinteraccion=lt(-95,Matrizinteraccion)';% matriz de ceros y unos
presvmH=presvmH'; % transpuesta de la matriz
presvmH(:,1:8999)=[];
cana=1;
hand1=0;hand2=0;hand3=0;hand4=0;hand5=0;
hand6=0;hand7=0;hand8=0;hand9=0;hand10=0;
fallido1=0;fallido2=0;fallido3=0;fallido4=0;fallido5=0;
fallido6=0;fallido7=0;fallido8=0;fallido9=0;fallido10=0;
handin1=0;handin2=0;handin3=0;handin4=0;handin5=0;
handin6=0;handin7=0;handin8=0;handin9=0;handin10=0;
anti1=0;anti2=0;anti3=0;anti4=0;anti5=0;
anti6=0;anti7=0;anti8=0;anti9=0;anti10=0;
inte1=0;inte2=0;inte3=0;inte4=0;inte5=0;
inte6=0;inte7=0;inte8=0;inte9=0;inte10=0;
perfecto1=0;perfecto2=0;perfecto3=0;perfecto4=0;perfecto5=0;
perfecto6=0;perfecto7=0;perfecto8=0;perfecto9=0;perfecto10=0;
tablahand=zeros(10,1);
tablafallido=zeros(10,1);
tablaanti=zeros(10,1);
tablainte=zeros(10,1);
tblaperfecto=zeros(10,1);
%% primer minuto
for i=1:180
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand1=hand1+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido1=fallido1+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
        handin1=handin1+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand1=hand1+1;
            cana=cana+1;
            intel1=intel1+1;
            handin1=0;
        end
    end
end
```

```

if Matrizinteraccion(mcaH(cana,1),i+1)==1
    fallido1=fallido1+1;
end
else
    handin1= handin1+1;
end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
    anti1=anti1+1;
    hand1=hand1+1;
    cana=cana+1;
    if handin1>0
        intel1=intel1+1;
    else
        handin1=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido1=fallido1+1;
    end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
ccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto1=perfecto1+1;
        cana=cana+1;
    end
    if presvmH(mcaH(cana,1),i+1)==1
        hand1=hand1+1;
        cana=cana+1;
        if handin1>0
            intel1=intel1+1;
        else
            handin1=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido1=fallido1+1;
        end
    end
end
end
end
tablahand(1,1)=hand1;
tablafallido(1,1)=fallido1;
tablaanti(1,1)=anti1;
tablainte(1,1)=intel1;
tablaperfecto(1,1)=perfecto1;

```

```
%% segundo minuto
for i=181:360
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand2=hand2+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido2=fallido2+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
        handin2=handin2+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand2=hand2+1;
            cana=cana+1;
            inte2=inte2+1;
            handin2=0;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido2=fallido2+1;
            end
        end
    else
        handin2= handin2+1;
    end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
        anti2=anti2+1;
        hand2=hand2+1;
        cana=cana+1;
        if handin2>0
            inte2=inte2+1;
        else
            handin2=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido2=fallido2+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1
        if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
cción(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
            perfecto2=perfecto2+1;
            cana=cana+1;
        end
        if presvmH(mcaH(cana,1),i+1)==1
```

```

        hand2=hand2+1;
        cana=cana+1;
        if handin2>0
            inte2=inte2+1;
        else
            handin2=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido2=fallido2+1;
        end
    end
end
tablahand(2,1)=hand2;
tablafallido(2,1)=fallido2;
tablaanti(2,1)=anti2;
tablainte(2,1)=inte2;
tablaperfecto(2,1)=perfecto2;
%% tercer minuto
for i=361:540
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand3=hand3+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido3=fallido3+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==1
        handin3=handin3+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand3=hand3+1;
            cana=cana+1;
            inte3=inte3+1;
            handin3=0;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido3=fallido3+1;
            end
        end
    else
        handin3= handin3+1;
    end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i)==0
        anti3=anti3+1;
        hand3=hand3+1;
    end
end

```

```
cana=cana+1;
if handin3>0
    inte3=inte3+1;
else
    handin3=0;
end
if Matrizinteraccion(mcaH(cana,1),i+1)==1
    fallido3=fallido3+1;
end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i)
)==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(cana,1),
i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizinteraccion(mcaH(cana,1),
i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto3=perfecto3+1;
        cana=cana+1;
    end
    if presvmH(mcaH(cana,1),i+1)==1
        hand3=hand3+1;
        cana=cana+1;
        if handin3>0
            inte3=inte3+1;
        else
            handin3=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido3=fallido3+1;
        end
    end
end
end
tablahand(3,1)=hand3;
tablafallido(3,1)=fallido3;
tablaanti(3,1)=anti3;
tablainte(3,1)=inte3;
tblaperfecto(3,1)=perfecto3;

%% cuarto minuto
for i=541:720
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand4=hand4+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido4=fallido4+1;
            end
        end
```

```

    end
elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
    handin4=handin4+1;
if presvmH(mcaH(cana,1),i+1)==1
    hand4=hand4+1;
cana=cana+1;
inte4=inte4+1;
handin4=0;
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido4=fallido4+1;
    end
else
    handin4=handin4+1;
end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
    anti4=anti4+1;
hand4=hand4+1;
cana=cana+1;
    if handin4>0
        inte4=inte4+1;
    else
        handin4=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido4=fallido4+1;
    end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i)
==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
cción(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto4=perfecto4+1;
        cana=cana+1;
    end
if presvmH(mcaH(cana,1),i+1)==1
    hand4=hand4+1;
cana=cana+1;
    if handin4>0
        inte4=inte4+1;
    else
        handin4=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido4=fallido4+1;

```

```
    end
end
end
end
end
tablahand(4,1)=hand4;
tablafallido(4,1)=fallido4;
tablaanti(4,1)=anti4;
tablainte(4,1)=inte4;
tablaperfecto(4,1)=perfecto4;
%% quinto minuto
for i=721:900
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand5=hand5+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido5=fallido5+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
        handin5=handin5+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand5=hand5+1;
            cana=cana+1;
            inte5=inte5+1;
            handin5=0;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido5=fallido5+1;
            end
        else
            handin5=handin5+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
        anti5=anti5+1;
        hand5=hand5+1;
        cana=cana+1;
        if handin5>0
            inte5=inte5+1;
        else
            handin5=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido5=fallido5+1;
        end
    end
```

```

elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i)
)==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
cción(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto5=perfecto5+1;
        cana=cana+1;
    end
    if presvmH(mcaH(cana,1),i+1)==1
        hand5=hand5+1;
        cana=cana+1;
        if handin5>0
            inte5=inte5+1;
        else
            handin5=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido5=fallido5+1;
        end
    end
end
end
tablahand(5,1)=hand5;
tablafallido(5,1)=fallido5;
tablaanti(5,1)=anti5;
tablainte(5,1)=inte5;
tablaperfecto(5,1)=perfecto5;
% sexto minuto
for i=901:1080
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand6=hand6+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido6=fallido6+1;
            end
        end
        elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
            handin6=handin6+1;
            if presvmH(mcaH(cana,1),i+1)==1
                hand6=hand6+1;
                cana=cana+1;
                inte6=inte6+1;
                handin6=0;
                if Matrizinteraccion(mcaH(cana,1),i+1)==1

```

```
    fallido6=fallido6+1;
end
else
    handin6=handin6+1;
end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
    anti6=anti6+1;
    hand6=hand6+1;
    cana=cana+1;
    if handin6>0
        inte6=inte6+1;
    else
        handin6=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido6=fallido6+1;
    end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
ccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto6=perfecto6+1;
        cana=cana+1;
    end
    if presvmH(mcaH(cana,1),i+1)==1
        hand6=hand6+1;
        cana=cana+1;
        if handin6>0
            inte6=inte6+1;
        else
            handin6=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido6=fallido6+1;
        end
    end
end
end
end
tablahand(6,1)=hand6;
tablafallido(6,1)=fallido6;
tablaanti(6,1)=anti6;
tablainte(6,1)=inte6;
tablaprofecto(6,1)=perfecto6;
%% séptimo minuto
```

```

for i=1081:1260
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand7=hand7+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido7=fallido7+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
        handin7=handin7+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand7=hand7+1;
            cana=cana+1;
            inte7=inte7+1;
            handin7=0;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido7=fallido7+1;
            end
        else
            handin7=handin7+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
        anti7=anti7+1;
        hand7=hand7+1;
        cana=cana+1;
        if handin7>0
            inte7=inte7+1;
        else
            handin7=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido7=fallido7+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1
        if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
ccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
            perfecto7=perfecto7+1;
            cana=cana+1;
        end
        if presvmH(mcaH(cana,1),i+1)==1
            hand7=hand7+1;
        end
    end
end

```

```
cana=cana+1;
if handin7>0
    inte7=inte7+1;
else
    handin7=0;
end
if Matrizinteraccion(mcaH(cana,1),i+1)==1
    fallido7=fallido7+1;
end
end
end
tablahand(7,1)=hand7;
tablafallido(7,1)=fallido7;
tablaanti(7,1)=anti7;
tablainte(7,1)=inte7;
tablaperfecto(7,1)=perfecto7;
%% octavo minuto
for i=1261:1440
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand8=hand8+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido8=fallido8+1;
            end
        end
        elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
            handin8=handin8+1;
            if presvmH(mcaH(cana,1),i+1)==1
                hand8=hand8+1;
                cana=cana+1;
                inte8=inte8+1;
                handin8=0;
                if Matrizinteraccion(mcaH(cana,1),i+1)==1
                    fallido8=fallido8+1;
                end
            end
            else
                handin8=handin8+1;
            end
        elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
            anti8=anti8+1;
            hand8=hand8+1;
            cana=cana+1;
```

```

        if handin8>0
            inte8=inte8+1;
        else
            handin8=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido8=fallido8+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i)
        ==1
            if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
                cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
                ccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
                perfecto8=perfecto8+1;
                cana=cana+1;
            end
            if presvmH(mcaH(cana,1),i+1)==1
                hand8=hand8+1;
                cana=cana+1;
                if handin8>0
                    inte8=inte8+1;
                else
                    handin8=0;
                end
                if Matrizinteraccion(mcaH(cana,1),i+1)==1
                    fallido8=fallido8+1;
                end
            end
        end
    end
    tablahand(8,1)=hand8;
    tablafallido(8,1)=fallido8;
    tablaanti(8,1)=anti8;
    tablainte(8,1)=inte8;
    tablaperfecto(8,1)=perfecto8;
    %% noveno minuto
    for i=1441:1620
        if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
            if presvmH(mcaH(cana,1),i+1)==1
                hand9=hand9+1;
                cana=cana+1;
                if Matrizinteraccion(mcaH(cana,1),i+1)==1
                    fallido9=fallido9+1;
                end
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),

```

```
i)==1
    handin9=handin9+1;
    if presvmH(mcaH(cana,1),i+1)==1
        hand9=hand9+1;
        cana=cana+1;
        inte9=inte9+1;
        handin9=0;
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido9=fallido9+1;
        end
    else
        handin9=handin9+1;
    end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
    anti9=anti9+1;
    hand9=hand9+1;
    cana=cana+1;
    if handin9>0
        inte9=inte9+1;
    else
        handin9=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido9=fallido9+1;
    end
elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1
    if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(
cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizintera
ccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
        perfecto9=perfecto9+1;
        cana=cana+1;
    end
    if presvmH(mcaH(cana,1),i+1)==1
        hand9=hand9+1;
        cana=cana+1;
        if handin9>0
            inte9=inte9+1;
        else
            handin9=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido9=fallido9+1;
        end
    end
end
```

```

    end
end
tablahand(9,1)=hand9;
tablafallido(9,1)=fallido9;
tablaanti(9,1)=anti9;
tablainte(9,1)=inte9;
tblaperfecto(9,1)=perfecto9;
% decimo minuto
for i=1620:1800
    if presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),i)==0
        if presvmH(mcaH(cana,1),i+1)==1
            hand10=hand10+1;
            cana=cana+1;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido10=fallido10+1;
            end
        end
    elseif presvmH(mcaH(cana,1),i)==0 && Matrizinteraccion(mcaH(cana,1),
i)==1
        handin10=handin10+1;
        if presvmH(mcaH(cana,1),i+1)==1
            hand10=hand10+1;
            cana=cana+1;
            inte10=inte10+1;
            handin10=0;
            if Matrizinteraccion(mcaH(cana,1),i+1)==1
                fallido10=fallido10+1;
            end
        else
            handin10= handin10+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),
i)==0
        anti10=anti10+1;
        hand10=hand10+1;
        cana=cana+1;
        if handin10>0
            inte10=inte10+1;
        else
            handin10=0;
        end
        if Matrizinteraccion(mcaH(cana,1),i+1)==1
            fallido10=fallido10+1;
        end
    elseif presvmH(mcaH(cana,1),i)==1 && Matrizinteraccion(mcaH(cana,1),i
)==1

```

```
if Matrizinteraccion(mcaH(cana,1),i)==1 || Matrizinteraccion(mcaH(cana,1),i+1)==1 || Matrizinteraccion(mcaH(cana,1),i+2)==1 || Matrizinteraccion(mcaH(cana,1),i+3)==1 || Matrizinteraccion(mcaH(cana,1),i+4)==1
    perfecto10=perfecto10+1;
    cana=cana+1;
end
if presvmH(mcaH(cana,1),i+1)==1
    hand10=hand10+1;
    cana=cana+1;
    if handin10>0
        inte10=inte10+1;
    else
        handin10=0;
    end
    if Matrizinteraccion(mcaH(cana,1),i+1)==1
        fallido10=fallido10+1;
    end
end
end
end
tablahand(10,1)=hand10;
tablafallido(10,1)=fallido10;
tablaanti(10,1)=anti10;
tablainte(10,1)=inte10;
tablaperfecto(10,1)=perfecto10;
totalhand=cumsum(tablahand);
totalfallido=cumsum(tablafallido);
totalanti=cumsum(tablaanti);
totalinte=cumsum(tablainte);
totalperfecto=cumsum(tablaperfecto);
```


Autores

Cristian Camilo Bernal Ariza

Ingeniero electrónico y magíster en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas. Actualmente se desempeña como ingeniero de Telefónica Colombia. Ha realizado proyectos de investigación y publicaciones en el área de las telecomunicaciones en revistas indexadas de categoría nacional e internacional.

César Augusto Hernández Suárez

Ingeniero electrónico con especialización en Interconexión de Redes; magíster en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas, y doctor en Ingeniería de la Universidad Nacional de Colombia. Profesor titular de la Universidad Distrital Francisco José de Caldas, adscrito a los programas de Tecnología en Sistemas Eléctricos de Media y Baja Tensión e Ingeniería Eléctrica de la Facultad Tecnológica. Investigador Senior de Colciencias, director del grupo de investigación SIREC con categoría A1 de Colciencias, e integrante de los grupos de investigación Armos e Internet Inteligente, en los que lidera investigaciones sobre sistemas y redes cognitivas y aplicaciones tecnológicas que contribuyen a mejorar la calidad de vida de comunidades vulnerables. Ha realizado publicaciones de patentes, libros de investigación y artículos en el área de las telecomunicaciones en revistas indexadas de categoría nacional e internacional.



Este libro se
terminó de imprimir
en noviembre de 2019
en la Editorial UD
Bogotá, Colombia