# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

## DETECCIÓN DE PATRONES DE COMPORTAMIENTO DE PARÁMETROS DE RF EN REDES DE COMUNICACIÓN MÓVIL MEDIANTE MEDICIONES DE CAMPO Y TÉCNICAS DE MACHINE LEARNING

## PROPUESTA DE UN MODELO PREDICTIVO PARA IDENTIFICAR FACTORES QUE SE MANIFIESTAN EN LAS FALLAS DE HANDOVER EN REDES LTE UTILIZANDO MEDICIONES DE CAMPO Y UN ÁRBOL DE DECISIÓN EN R STUDIO (CASO DE ESTUDIO VOIP).

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TELECOMUNICACIONES**

**CRISTIAN JAVIER ROCHA HIDROBO**
**(cristian.rocha@epn.edu.ec)**
**DIRECTOR: DR. PABLO LUPERA MORILLO**
**(pablo.lupera@epn.edu.ec)**

**DMQ, febrero 2022**

# CERTIFICACIONES

Yo, CRISTIAN JAVIER ROCHA HIDROBO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

**CRISTIAN JAVIER ROCHA HIDROBO**

Certifico que el presente trabajo de integración curricular fue desarrollado por CRISTIAN JAVIER ROCHA HIDROBO, bajo mi supervisión.

**DR. PABLO LUPERA MORILLO**
**DIRECTOR**

# DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CRISTIAN JAVIER ROCHA HIDROBO
PABLO LUPERA MORILLO

# DEDICATORIA

A mi madre y mi hermano que han estado conmigo siempre, les dedico un paso más hacia adelante.

# AGRADECIMIENTO

Quiero agradecer a mi madre que siempre ha buscado la manera de apoyarme en este camino difícil, ha logrado desde siempre darme inspiración y empujarme siempre más lejos.

A mi hermano que ha sido mi fuente de energía cuando las situaciones se volvían mas complicadas.

A Liz quien me ha apoyado desde el comienzo en este camino difícil y me ha acompañado hasta conseguir mis objetivos.

A mi abuelos que han estado presentes siempre y que nunca dejaron de creer en mi.

A mis compañeros de carrera con quienes compartí alegrías y tristezas, que me han ayudado en los momentos complicados y con su apoyo aligeraron muchos días difíciles.

Agradezco al Dr. Pablo Lupera por la dedicación brindada durante la realización de este trabajo y el empeño con el que ejerce su profesión.

# INDICE DE CONTENIDO

# RESUMEN

El presente trabajo busca elaborar un modelo que permita comprender cuales son los factores que llevan a un handover a fallar, esto mediante la aplicación de machine learning, concretamente árboles de decisión para obtener la proporción en la que los factores encontrados contribuyen a un handover fallido.

En el primer capítulo se estudian los elementos que conforman una red LTE, para luego abordar los tipos y causas comunes del handover, también se revisan los conceptos concernientes a machine learning y al uso del lenguaje de programación R para el análisis de grandes grupos de datos y se exploran las herramientas digitales que permiten recolectar los datos necesarios para el estudio.

En el segundo capítulo se propone el método para recolectar la información necesaria para construir el modelo, se describe el proceso para encontrar una zona con fallos de handover y luego el trazado de rutas para continuar con la toma de muestras; en este capítulo también se realiza el tratamiento de datos y la programación en R necesaria para construir el árbol de decisión que permita analizar las variables obtenidas.

En el tercer capítulo se estudian los resultados obtenidos, se obtiene un modelo y se estudian las características obtenidas en el árbol de decisiones final. A partir de este modelo se obtienen conclusiones y recomendaciones sobre el funcionamiento del handover en la zona estudiada.

**Palabras clave:** Handover LTE, VoIP, Árbol de decisión, R software.

# ABSTRACT

The present work seeks to develop a model that allows to understand what are the factors that lead to a failed handover, this through the use of machine learning techniques, specifically decision trees, to obtain the proportion in which the factors found contribute to a failed handover.

In the first chapter, the elements of an LTE architecture are studied, and then study the causes and common types of handover in LTE network, as well as a review of the concepts concerning machine learning and the use of the programming language R for analysis of large groups of data, digital tools that allow collecting the necessary data for the study are explored too.

The second chapter proposes the method to collect the necessary information to build the model, the process to find an area with handover failures is described and then the routing of routes to continue with the sampling; In this chapter, the data processing and programming in R necessary to build the decision tree that allows analyzing the variables obtained is also carried out.

In the third chapter the results obtained are studied, a model is obtained and the characteristics obtained in the final decision tree are studied, from this model conclusions about how handover works in the area of study are obtained.



**Keywords:** Handover LTE, VoIP, Decision tree, R software

# 1 INTRODUCCIÓN

Las telecomunicaciones móviles han evolucionado hasta convertirse en un elemento central para sus usuarios y al existir cada vez más demanda de calidad es importante que la infraestructura se adapte para responder a las necesidades que surgen con el tiempo.

La tecnología *Long Term Evolution (LTE)* forma parte de las redes de cuarta generación para sistemas celulares, esta tecnología es una evolución de *Universal Mobile Telecommunications System (UMTS)* que es el equivalente de LTE en las redes de tercera generación; ambas tecnologías son definidas por el *3rd Generation Partnership Project (3GPP)* que es el organismo que especifica los apartados técnicos de distintas generaciones para que cumplan requisitos específicos de infraestructura [1].

La infraestructura LTE está diseñada para soportar tráfico de paquetes en su totalidad, lo que ha hecho que se popularice el uso de servicios y aplicaciones que requieran conexión a internet, estos servicios en su mayoría no exigen recursos permanentes de la red celular, excepto en los servicios de tiempo real, como lo son las llamadas de video o voz, que exigen recursos permanentes de la infraestructura para mantener una calidad razonable.

Se puede dividir en dos partes la infraestructura de una red celular: *Radio Access Network (RAN) ,* que administra la comunicación con el equipo de usuario (UE) y el *Core Network,* que es la encargada de gestionar los servicios que requiere el equipo de usuario.

En LTE se tiene Evolved Universal Mobile Telecommunications System (E-UTRA) para la parte de radio y dentro de este elemento se cuenta específicamente con un componente que se comunica con el equipo de usuario directamente, llamado Evolved Node B(e-NodeB) [1].

Uno de los usos más extendidos para las redes celulares son las llamadas mediante servicios *Voice over IP (VoIP)* que se utilizan en aplicaciones como Whatsapp, Messenger, Skype, etc. Para que un dispositivo realice llamadas VoIP mediante la red LTE se deben cumplir ciertos requisitos de infraestructura para que no existan interferencias (interrupciones) y en el peor de los casos, la llamada finalice inesperadamente [2].

Una llamada de VoIP puede fallar por diversas razones; sin embargo, cuando se tiene una infraestructura con elementos inalámbricos es más probable que los fallos se den en esa etapa de comunicación, este es el caso de las redes celulares que poseen el enlace de comunicación entre el eNodeB y el equipo de usuario, debido a esto es

importante identificar los factores que afectan a la comunicación entre estos dos elementos.

Dentro de la comunicación inalámbrica existen muchos factores que pueden afectar a la transmisión de datos y es difícil identificarlos, ya que se requeriría estudiar cada caso en el que los dispositivos presentan fallas y debido a la cantidad de variables que se encuentran involucradas en este proceso, resulta difícil identificar cuál de todos es el más importante, para resolver estas situaciones existen varios métodos. En el presente trabajo se ha optado por utilizar técnicas de machine learning y así crear un modelo predictivo que ayude a identificar los elementos cruciales [2].

Utilizando un modelo concreto de machine learning, que en este caso son los árboles de decisiones, se pueden construir modelos que permitan predecir el comportamiento que va a tomar un sistema tomando como punto de partida una base de datos dada. En este caso se van a utilizar muestras de llamadas fallidas correlacionadas con datos de las antenas en puntos críticos donde las llamadas tienden a fallar. Actualmente, existen diversas herramientas con algoritmos propios que permiten el análisis de datos mediante la implementación de árboles de decisiones; en el caso de este trabajo se va a utilizar el software “R” que posee funciones que permiten implementar árboles de decisiones y además cuenta con funciones estadísticas adicionales para tratar los datos recolectados [3].

## 1.1 OBJETIVO GENERAL

Determinar los parámetros de radio enlace relevantes que se manifiestan en zonas de handover fallidos durante llamadas de VoIP en redes LTE mediante técnicas de Machine Learning.

## 1.2 OBJETIVOS ESPECÍFICOS

- Localizar y recolectar datos de zonas donde la calidad de las llamadas de VoIP se deteriora por handover fallidos.
- Elaborar un algoritmo en R Studio que permita preparar una base de datos con información de handover y calidad de llamadas.
- Explicar e implementar en R Studio un árbol de decisión que muestre los factores críticos durante el proceso de handover.
- Determinar la validez del modelo evaluando su precisión.

## 1.3 ALCANCE

En este trabajo primero se identificará la ubicación de puntos críticos de la red, estos puntos se localizarán mediante la realización de llamadas de VoIP y se detectará si existen interrupciones o fallos al darse un handover. Una vez detectados los puntos

geográficos se recolectarán los datos del estado de la señal mediante el uso de la aplicación gratuita “NetMonitor Cell Signal Logging” disponible en la appstore de Android utilizando un celular de marca Xiaomi modelo Mi-A3. Esta aplicación permite recolectar datos correspondientes a las redes celulares para la supervisión de la calidad de la red LTE y generaciones anteriores. Los datos recolectados serán procesados previo a su análisis.

Los datos que se utilizarán y que la aplicación entrega son:

- Hora del sistema
- Identificador de Celda
- Cantidad de celdas vecinas LTE
- RSSI más alto de las celdas vecinas
- RSSI de la celda actual
- RSRQ
- RSSNR
- Velocidad del UE

Asimismo es importante considerar al handover como un factor relevante al momento de darse problemas en las llamadas de VoIP; sin embargo, se debe tomar en cuenta que este dato no es proporcionado por la aplicación. Las zonas de handover se detectarán de forma automática mediante un procesamiento previo de los datos, las muestras consecutivas que pertenezcan a celdas distintas estarían representando un handover y de esta manera se podría determinar en cada registro los puntos donde se ha dado un handover en la red.

Además, ya que los identificadores de celdas sirven también para diferenciar entre distintos eNB se los puede utilizar para determinar si este handover se dio entre celdas que pertenecen a un mismo eNB o entre celdas de distintos eNB, un algoritmo de estas características se lo puede implementar en el lenguaje de programación “R”, ya que permite operaciones de clasificación de los datos y tratamiento de bases de datos extensas.

En el tratamiento previo de los datos también se debe agregar una categoría binaria que represente la percepción de la llamada, esta categoría podría compararse con la escala que se utiliza comúnmente para verificar la calidad de llamadas que es la escala MOS (Mean Opinion Score), pero de una manera más sencilla. Por ello, se opta por una categoría binaria de evaluación, en la que existan dos valores de calidad para la llamada, estos valores serían: buena y mala, siendo “buena” la clasificación que indica que la llamada no presenta inconvenientes y “mala” que los presenta aunque sean mínimos y no signifiquen la finalización de la llamada, esta manera de clasificar permite

que los datos recolectados en la fase de adquisición vuelvan al modelo concluyente en resultados.

Los datos recolectados formarán parte de la base de registros que será analizada mediante el lenguaje “R”, este es un lenguaje estadístico que facilita el tratamiento de la información y además cuenta con herramientas que permiten la implementación de Machine Learning.

Luego de obtener una base de datos con una cantidad representativa de muestras de eventos de degradaciones de la calidad de las conexiones de VoIP, se procede a limpiar los datos siguiendo un criterio basado en el siguiente cuadro.

**Tabla 1.1**. Criterio para descartar mediciones

| **Parámetro** | **Criterio de aceptación** | **Justificación** |
|---|---|---|
| Cantidad de celdas vecinas | $X \neq 0$ | Cuando no existen celdas vecinas no puede existir un handover. |
| RSSI | $X > -110$ dBm | Debajo de este valor las interferencias son causadas por una señal pobre y no representan un factor fuera de lo normal. |

De esta manera se puede conformar un conjunto de datos con información relevante para el objetivo de este trabajo, los datos se prepararán mediante un algoritmo elaborado en R Studio que permite seleccionar los registros bajo criterios personalizados, luego de esto se procede a agregar la información de calidad de servicio y de handover, de esta manera se completa el registro de análisis y la toma de muestras.

Cuando se tenga el conjunto de datos final con los elementos requeridos, se procede a la fase de análisis, en esta fase se va a ingresar la información a un algoritmo que segmente las muestras para dividir la base de datos en datos de entrenamiento y datos de prueba.

Los datos finales se procesarán a través de un script de R Studio que utilice las librerías disponibles para utilizar la herramienta de Machine Learning y la aplicación específica del árbol de decisiones, de esta manera se obtiene un diagrama comprensible de variables relevantes para determinar factores que afecten a la calidad de las llamadas.

Para la implementación de un modelo basado en árbol de decisiones existen muchos programas y con diversas aplicaciones pero para este trabajo se utilizará R Studio que es un IDE construido exclusivamente para programar en el lenguaje “R”, este lenguaje de programación se enfoca en el análisis estadístico, por lo que es muy útil para clasificar, filtrar y organizar cantidades enormes de datos y cuenta con licencia GNU GPL, al igual que su IDE R STUDIO, lo que representa una ventaja, ya que no es necesario comprar licencias para su utilización.

Este proyecto de integración curricular no tiene producto final demostrable.

## 1.4 MARCO TEÓRICO

### 1.4.1 LTE

Long-Term Evolution (LTE) define un conjunto de estándares que buscan mejorar las características técnicas de arquitecturas anteriores, este conjunto de estándares fue definido por la organización 3GPP quienes publican cada cierto tiempo nuevos grupos de estándares que deben seguir las arquitecturas móviles para atribuírseles el nombre de una determinada tecnología. A estas tecnologías también se les conoce comercialmente con la denominación de generación, en este contexto LTE se ubica en la cuarta generación o 4G de redes móviles [4].

Dentro de las mejoras mas relevantes que presenta esta nueva generación esta el aumento de velocidad de descarga de datos, mientras que en 3G se tenían velocidades de 1 a 4 Mbps(Mega bit por segundo), en 4G se pueden alcanzar velocidades de hasta 400 Mbps [5] también a las llamadas convencionales que se realizaban entre equipos móviles se les ha realizado mejoras importes en cuanto a calidad, a este nuevo servicio se le conoce como VoLTE que es simplemente un nuevo nombre para las llamadas entre equipos móviles dentro de la estructura LTE. Adicionalmente, la buena estabilidad y calidad de la red de datos 4G ha permitido que los servicios que utilizan aplicaciones como Whatsapp o Messenger para realizar llamadas se popularicen. A los servicios que ofrecen estas aplicaciones para llamar se los conoce como VoIP que define una comunicación por voz mediante la red IP.

#### 1.4.1.1 Elementos de la arquitectura LTE

El estándar que define los elementos de la red LTE es el release 8, definido por 3GPP, dentro de este estándar se definen dos estructuras importantes para el funcionamiento de la red, estos son E-UTRAN (Evolved UMTS Terrestrial Radio Access Network) y el EPC (Evolved Packet Core).

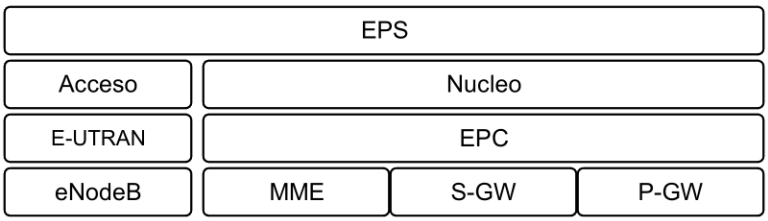


**Figura 1.1** Elementos de la arquitectura LTE

E-UTRAN es la parte de acceso de la arquitectura, es decir, es la parte que se encarga de establecer el enlace de radio entre la red de núcleo y el equipo móvil o UE. Esta parte de la arquitectura solo se conforma por el eNodeB que es el nombre que recibe la antena celular en la red LTE y este es el único elemento de la red de acceso, aquí también se

toman decisiones que involucran los parámetros de la señal que recibe el UE. Además para garantizar la continuidad de una llamada, el eNodeB puede decidir si hacer o no un handover [1].

El Evolved Packet Core (EPC) se compone de tres entidades: el Mobility Management Entity (MME), el Serving Gateway (S-GW) y el Packet data network Gateway (P-GW). El MME funciona en el plano de control de la red que es donde existen protocolos para los procesos de control de los elementos de red, mientras que el S-GW y el P-GW trabajan con el plano de usuario que es el plano donde funcionan protocolos que transportan el tráfico de usuario.

Dentro de una red LTE, un EPC puede controlar a varios eNodeB, de esta manera se tienen estructuras donde existen varios EPC controlando a varios eNodeB sin importar si estos ya tienen un EPC. Los eNodeB se comunican entre ellos mediante la interfaz X2 y se comunican con los EPC mediante la interfaz S1.

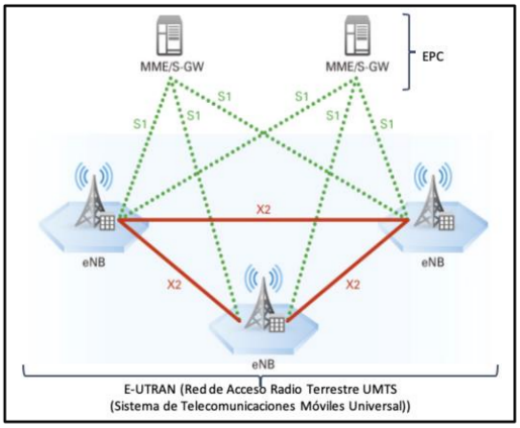


**Figura 1.2.** Interfaces de conexión LTE [1]

### 1.4.1.2 Handover en LTE

Las estaciones base, eNodeB para el caso de LTE, tienen una cobertura limitada en cuanto a distancia, debido a esto existen varias estaciones colocadas estratégicamente para conseguir cubrir una zona amplia y de esta manera mantener el servicio. Handover es el proceso mediante el cual un UE deja de estar conectado a una estación base específica y se conecta a otra. El desafío que hay en este proceso es que el handover se ejecute sin que el usuario lo perciba y esto se traduce en cambios entre estaciones base cada vez mas rápidos. Conforme las generaciones avanzan el proceso de handover se vuelve menos perceptible y conlleva menos efectos que el usuario puede percibir, logrando que un UE pueda moverse grandes distancias y conectarse a varias antenas sin que el usuario lo note; sin embargo, aun en 4G se presentan eventos que causan efectos perceptibles para el usuario durante un proceso de handover [6].

Existen dos tipos principales de handover el soft-handover y el hard-handover, la diferencia es que el soft mantiene una conexión con la estación base de la que parte, pero requiere que existan funciones extra en el UE, lo que vuelve más complejo. En el hard handover se produce una pequeña interrupción de conexión entre el eNodeB y el UE y es un proceso más simple. En las redes LTE solo se utiliza este último tipo de handover, y puede ser entre estaciones base que tengan la misma tecnología (intra-LTE) o distinta tecnología (inter-RAT) y el intra-LTE que puede ser intra-frecuencia o inter-frecuencia.

El proceso de handover entre celdas LTE puede complicarse dependiendo de si hay que cambiar de antena en el mismo eNodeB o si hay que cambiar de eNodeB y también al mismo tiempo de EPC.

Para que un handover se lleve a cabo se pueden dar 4 posibilidades en el algoritmo de ejecución que son nombradas como A1, A2, A3, A4 [7].

**Tabla 1.2** Casos de handover

| | |
|---|---|
| A1 | La señal de la celda en servicio supera un umbral configurado. |
| A2 | La señal de la celda en servicio cae bajo un umbral configurado. |
| A3 | La señal de la celda vecina supera a la celda en servicio más un offset. |
| A4 | La señal de la celda de servicio cae bajo un umbral 1 y la señal de la celda vecina supera un umbral 2. |

En los casos de A1 y A2 los umbrales son configurados en el equipo, mientras que en el caso de A3 el handover va a depender de la fuerza de la señal entre dos celdas vecinas. La potencia de la señal puede medirse mediante el rssi o rsrp, existe un momento en el que el rssi de la celda en servicio cae y el rssi de una celda vecina aumenta, cuando la diferencia entre estos valores es suficiente se produce un handover como se muestra en la figura 1.3.

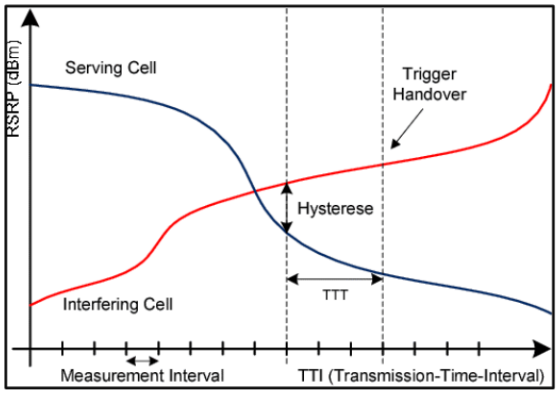


**Figura 1.3** Handover por diferencia de señal [7]

Sin embargo; la señal de cambio no se envía de manera inmediata, se espera un tiempo para intercambiar la señal; si no se hiciera esto, las señales podrían variar inmediata y aleatoriamente y habría un efecto de rebote, cambiando varias veces entre ambas estaciones base y reduciendo la calidad de la transferencia de datos.

### 1.4.2 VOIP

Se denomina VoIP a un conjunto de métodos que permiten el transporte de voz a través de una red que utilice Internet Protocol (IP), este servicio en cuanto a procedimientos es similar al servicio de telefonía tradicional, la diferencia radica en que la información no viaja sobre una red conmutada de circuitos, sino que la información es empaquetada y enviada a través de una red conmutada de paquetes [8].

Para hacer uso del servicio VoIP, es necesario que exista una aplicación en los terminales que gestione los protocolos de comunicación que se van a utilizar y una red conmutada de paquetes por la que se puedan transportar los datos que los terminales envían, en el caso de LTE se tiene una red que es totalmente IP por lo que se puede utilizar para el envío de estos paquetes; sin embargo, el proceso difiere de un servicio VoIP que trabaja sobre una Local Area Network (LAN) [2].

#### 1.4.2.1 VoIP sobre LTE

Cuando se utiliza VoIP más allá de una LAN y se transportan los paquetes a través del internet se denomina OTT VoIP, ya que la arquitectura se encuentra en el internet y no en una red local administrable, esta denominación encaja con el proceso que se lleva a cabo cuando se realiza una llamada VoIP mediante la red LTE, ya que lo que hace la red LTE con los paquetes es transportarlos a través de su infraestructura al internet, por lo que más allá de ella no existe una manera de supervisar la adecuada entrega de los paquetes [2].

Un aspecto importante cuando se tiene una red de transporte de paquetes es la calidad de servicio(QoS) que permite priorizar la entrega de paquetes para que de esta forma no existan problemas dependiendo del servicio que se utilice. QoS no es más que una manera de etiquetar a los paquetes con una prioridad especifica. En las redes de paquetes los servicios de mayor prioridad son los que requieren que sus paquetes lleguen con el menor retardo posible y este es el caso de las comunicaciones en tiempo real como lo es VoIP; sin embargo, este aspecto es controlado por la red por la que se transportan los paquetes [8].

En el caso de la red LTE por la cual se transportan los paquetes VoIP de las distintas aplicaciones no se puede brindar QoS, ya que para una arquitectura LTE todos los paquetes que un UE envía son tratados de la misma manera, es por esto que puede

existir una degradación del servicio debido a que la red no prioriza los paquetes de servicios que son más importantes [8].

#### 1.4.2.2 VoIP vs VoLTE

Existe una alternativa similar a VoIP nativa para las redes LTE, este método se llama VoLTE y es similar a VoIP, ya que utiliza paquetes para transportar los datos por voz; no obstante, ya que se trata de un servicio nativo de la red LTE se pueden aplicar prioridades a estos paquetes sobre otros, es decir QoS, por lo que la calidad de las llamadas será superior a una llamada VoIP; sin embargo, la implementación de este servicio se ha visto retrasada, ya que la arquitectura requerida para implementarlo es más complicada que la que se necesita para ofrecer llamadas VoIP [9].

Para que una red LTE pueda ofrecer VoIP mediante aplicaciones solo es necesario una red que permita transportar datos y esto es algo totalmente integrado en LTE, por lo que vuelve mucho mas sencillo el uso de esta tecnología para el usuario. Además, se debe tomar en cuenta que un requisito necesario es que ambos dispositivos cuenten con la conexión de datos, esto sería inútil en un escenario donde dos UE no cuentan con el servicio de datos y quieren comunicarse mediante una llamada por la red celular, en estas situaciones es donde actúa VoLTE, ya que si bien es similar en cuanto al transporte de datos que las llamadas VoIP no requiere de un contrato adicional con la compañía de telefonía [9].

### 1.4.3 MEDICIONES EN REDES LTE MEDIANTE UE

Los dispositivos móviles actuales tienen la capacidad de mostrar una cantidad de información que es de mucha utilidad al momento de verificar el estado de la estación base a la que se encuentra conectado el dispositivo. Los dos principales sistemas operativos en el mercado son Android y iOS, ambos ofrecen a los usuarios avanzadas características que permiten visualizar el estado y servicios ofrecidos por la estación base a la que el dispositivo se encuentra conectado. Android posee en su código métodos y clases que permite acceder a información importante sobre el estado de la conexión celular. Además se debe mencionar que es sencillo utilizar aplicaciones de terceros para obtener esta información e incluso registrarla y almacenarla [8].

#### 1.4.3.1 Parámetros de medición desde UE

Existen varios parámetros que se pueden obtener desde el dispositivo móvil, estos van a variar dependiendo de la aplicación que utilicemos, en la mayoría de casos los datos que se pueden recolectar se obtienen a partir de funciones que el sistema operativo ofrece y de esta manera están limitados a los datos que el sistema operativo provee a cada aplicación. Esto causa que dependiendo del modelo de dispositivo existan valores para ciertas variables y en otros casos no se pueden obtener esos valores.

Los parámetros que más se utilizan para evaluar el estado de una celda son [10]:

**RSSI (Received Signal Strength Indicator)**

Corresponde a un valor de la potencia de una señal recibida por un dispositivo, es la potencia que tiene el dispositivo respecto a 1 [mW], debido a esto sus unidades son los dBm. Comúnmente se lo utiliza para evaluar la potencia de las celdas en una zona determinada; sin embargo, resulta insuficiente para explicar todos los problemas que se pueden dar en una red celular, ya que su valor puede variar si la cantidad de datos transmitidos varía aunque físicamente la potencia sea la misma.

**RSRQ (Reference Signal Received Quality)**

Este es un indicador que se utiliza para describir la calidad de la señal en una zona determinada, para obtener su valor se utiliza la siguiente ecuación:

$$RSRQ = \frac{N * RSRP}{RSSI} \tag{1.1}$$

Donde N es el número de Resource Block transmitidos (este bloque es la unidad más pequeña de información que se transmite); RSRP corresponde a un valor de potencia más cercano a la potencia real recibida comparado con el RSSI.

**RSSNR (Reference Signal Signal to Noise Ratio)**

Es una medida de la potencia de la señal deseada respecto al ruido, esto permite evaluar el rendimiento de la red en función de las condiciones del medio.

### 1.4.3.2 Aplicaciones de almacenamiento de parámetros

**NetMonitor Cell Signal Logging**

Esta es una herramienta para dispositivos Android que permite la toma de muestras de los datos que almacena el dispositivo con una amplia variedad de opciones y un gran rango de medidas en comparación con el resto de herramientas disponibles en la tienda oficial de aplicaciones para dispositivos Android.

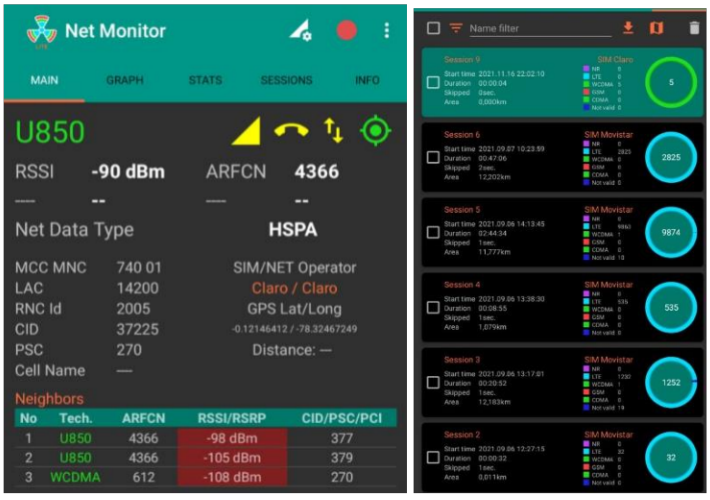


**Figura 1.4** Pantalla de inicio y menú de sesiones.

**Menú principal**

En el menú principal se presentan varios elementos y opciones que ayudarán a tomar muestras en tiempo real; sin embargo, estas muestras no son almacenadas hasta que el usuario lo decida, una ventaja de esta aplicación es que permite controlar la toma de muestras y separarlas por sesiones lo que facilita la manipulación de datos para cada sesión.

Una ventaja de esta aplicación es que luego de grabar varias sesiones se pueden cargar los datos de dichas sesiones para explorar los distintos apartados como el menú principal o el de gráficos para revisar los datos de cada instante de la medición. Se debe mencionar que también permite controlar el tiempo de la toma de muestras y explorar cada instante según se necesite.

**Menú sesiones**

En este menú se puede acceder a las sesiones creadas, desde aquí también se pueden cargar las sesiones grabadas para ser revisadas en la aplicación o también exportar las sesiones en formato csv o kmls.

Para poder exportar una sesión simplemente se mantiene el dedo presionado en la sesión que se desee exportar hasta que aparezca un menú emergente que ofrece la posibilidad de elegir el formato del archivo que se va a exportar, este archivo se guarda en el dispositivo. En el caso del presente trabajo de integración curricular se requiere de un archivo csv con las distintas mediciones que se realizan mediante la aplicación.

**Tabla 1.3** Variables de Network Monitor

| **Variable** | **Descripción** | **Variable** | **Descripción** |
|---|---|---|---|
| Report | Contador de mediciones | node_id | Identificador del eNodeB |
| sys_time | Hora del sistema | rssi | Potencia de la celda en dBm |
| sim_state | Estado de la SIM | rsrq | Indicador de calidad en dB |
| net_op_name | Nombre de operadora | rssnr | Potencia de la señal en función del ruido |
| net_op_code | Codigo de la operadora | arfcn | Código de las frecuencias de los carriers para tx y rx |
| Roaming | Estado de roaming | gps | GPS activo o inactivo |
| Net_type | Tecnología de la red | accuracy | Precisión de GPS |
| call_state | Posibilidad de llamada | lat | Latitud |

| | | | |
|---|---|---|---|
| data_state | Estado de los datos | long | Longitud |
| height | Altura a la que se encuentra el UE | band | Banda de la celda |
| data_rx | Datos recibidos | lte_neighbors | Número de vecinos LTE |
| data_tx | Datos transmitidos | rssi_strongest | Potencia de celda vecina |
| gsm_neighbors | Número de vecino gsm | cid | Identificador de la celda |
| umts_neighbors | Número de veinos umts | mcc | Código de país |
| lte_neighbors | Número de vecinos LTE | lac_tac | Codigo de area |
| rssi_strongest | Potencia de celda vecina | | |

### CellMapper

Esta herramienta permite almacenar los datos de medidas que se realicen en una zona específica, estos datos son procesados por un algoritmo que localiza las posiciones de las torres celulares a las que el dispositivo se encuentra conectado o se ha conectado durante las mediciones. Este tipo de información es útil para tener una idea de la probable localización de las antenas de las estaciones base que se desean estudiar. Luego de identificar una posible ubicación de una antena definida se puede comprobar físicamente su localización y corregirla si es necesario.

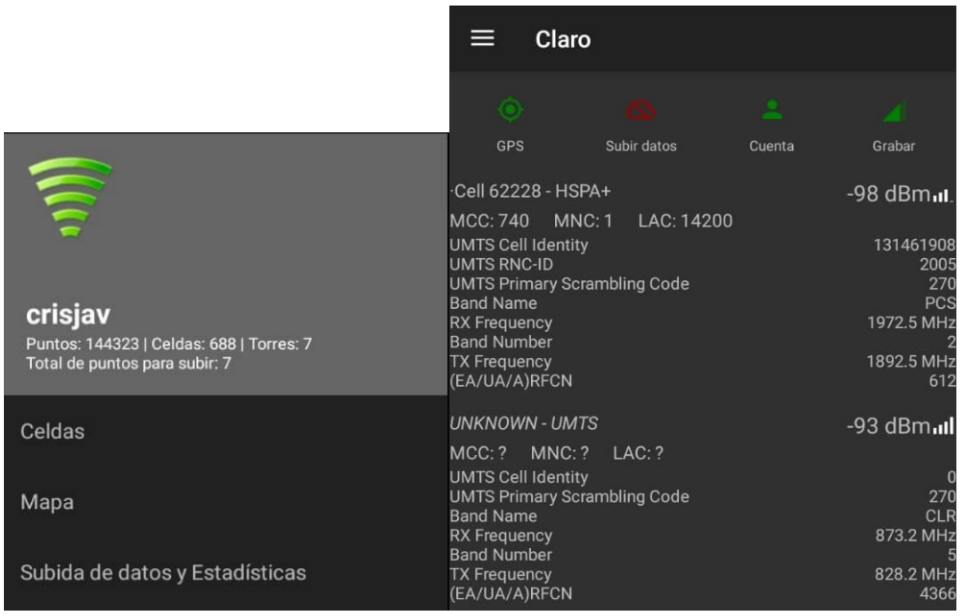


**Figura 1.5.** Menú inicial de cellmapper

En esta sección de la aplicación se encuentra la información general de la red. Se requiere crear una cuenta en cellmapper para poder subir la información a la base de datos colaborativa que permite compartir ubicaciones de antenas en todo el mundo. Se

puede ver en la Figura 1.5 los botones indicadores del estado de la aplicación, el primer símbolo describe el estado del sensor GPS del dispositivo, el segundo indica si es posible subir datos a la nube, el tercero advierte si hemos ingresado a nuestra cuenta y el cuarto anuncia si estamos recolectando medidas. Estos dos últimos se pueden presionar para activar o desactivar sus funciones.

Los datos se pueden subir permanentemente al sistema en línea de la aplicación; para esto, se debe activar la subida automática, sin embargo, se debe mencionar que esto conlleva un mayor consumo de batería, por lo que se recomienda subirlos manualmente cada vez que se hayan recolectado varios puntos de medidas. Para subir los datos se ingresa al menú y desde ahí se selecciona la opción Subida de datos y Estadísticas; esto solo se puede realizar si se dispone de una cuenta en la herramienta, y los datos subidos se pueden observar desde la página web. Los datos también pueden exportarse como csv; sin embargo, no cuentan con cabecera por lo que es difícil su procesamiento. En el presente trabajo de integración curricular se utilizará esta herramienta para registrar las torres con las que el dispositivo se conectará durante las pruebas de campo, así como sus posiciones [11].

**GPS Logger**

Esta aplicación se va a utilizar para registrar la posición donde se den eventos concretos, en el caso de este trabajo de integración curricular se utilizará para registrar los puntos donde existen fallos en la comunicación entre el dispositivo y la estación base. Una desventaja de la aplicación es que no permite exportar los datos como un archivo csv; los datos son guardados como txt; sin embargo, se debe notar que las características con las que los datos son almacenados son similares a las de un archivo csv, por lo que basta con cambiar la extensión del archivo txt a csv para que pueda ser procesado por RStudio, esto es posible debido a que aunque la aplicación no exporta los archivos explícitamente como csv si se guardan con las características de un csv, pero en archivos con extensión txt.

**Force LTE**

Esta herramienta facilita el acceso a las configuraciones de red celular en dispositivos Android, por defecto para ingresar a estas configuraciones se deben ingresar códigos específicos a cada modelo; pero esta aplicación facilita este proceso ofreciendo varios métodos para ingresar a la configuración y así configurar el dispositivo para que se conecte solamente a las redes que sean importantes para el estudio. En el presente proyecto se utilizará esta herramienta para que el terminal móvil se mantenga conectado solamente a la red 4G [10].

### 1.4.4 MACHINE LEARNING

#### 1.4.4.1 Aprendizaje Supervisado

El aprendizaje supervisado es parte del Machine Learning y consiste en conectar un conjunto de datos de entrada con un conjunto de datos de salida utilizando muchos ejemplos de entradas y salidas. El aprendizaje se realiza por medio de iteraciones de los datos, lo que permite al sistema que procesa la información, llegar a una función que logre predecir la salida de nuevas entradas para las que no se tengan datos. En estos modelos es común dividir los datos de entrada en dos partes una para el entrenamiento y otro para probar el modelo [12].

**Matriz de confusión**

Dentro del aprendizaje automatizado existen herramientas que facilitan la lectura del éxito que pudo tener un modelo al predecir eventos, la matriz de confusión grafica en una tabla las salidas esperadas con las salidas reales y se espera que la mayoría de resultados caigan en la diagonal principal.

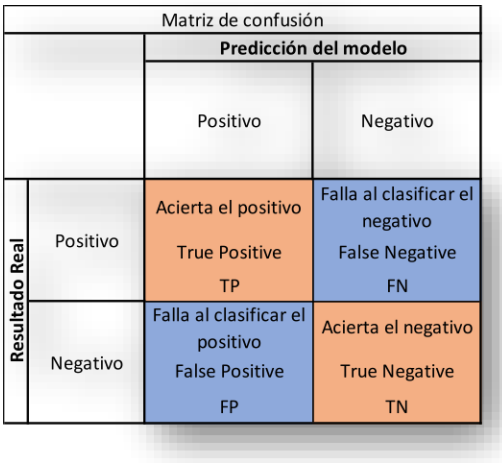


**Figura 1.6.** Matriz de confusión

De esta manera se puede evaluar un modelo solo observando la cantidad de fallos que pudo tener en las salidas. La primera casilla y la última representan los casos de éxito del modelo, mientras que las casillas restantes representan los casos de fracaso al clasificar los datos, para obtener esta matriz es necesario que la base de datos del modelo sea dividida y una pequeña porción de los datos se utilicen para poner a prueba el modelo y obtener los datos de la predicción realizada.

Dentro del aprendizaje supervisado existen varias clasificaciones que permiten dividir los tipos de variables con los que se está trabajando, existen dos principales clasificaciones para estos modelos: clasificación y regresión [12].

**Clasificación**

Este tipo de entrenamiento supervisado consiste en algoritmos que clasifican datos de entrada en categorías o variables discretas, estos algoritmos son útiles cuando se busca reconocer patrones en observaciones [12].

**Regresión**

En los modelos de regresión se tiene una salida que no es discreta, puede tomar un valor que dependerá de las entradas, ya que la salida va a tener una relación funcional con la entrada. En estos modelos se pueden tener valores reales continuos en la salida; para medir la exactitud del modelo se calcula la desviación entre la salida obtenida y la salida esperada [12].

### 1.4.4.2 Árboles de decisión

Los árboles de decisión se usan como modelos predictivos, entran en la categoría de aprendizaje supervisado, en los árboles de clasificación las entradas se clasifican en determinadas categorías, en estas estructuras las hojas de los árboles son las clases y las ramas son las condiciones que conducen a cada una de las hojas [13].

**Estructura**

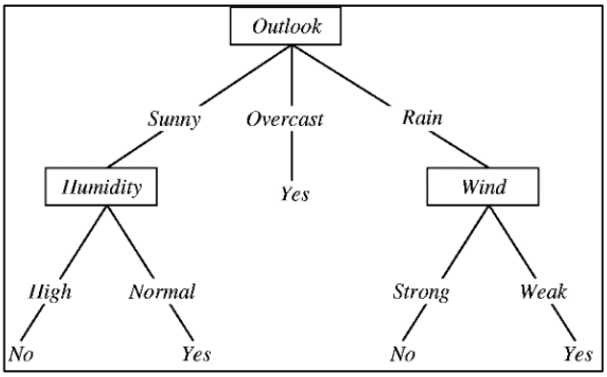


**Figura 1.7** Ejemplo de un árbol de decisión [14]

La estructura de un árbol de decisión es fácil de leer y comprender, ya que grafica las condiciones que llevan a las variables a clasificarse en las categorías que se buscan. En este caso se tiene un gráfico de un árbol que permite modelar un pronóstico de lluvia, de esta manera si se tiene una lectura de las condiciones ambientales es posible obtener una predicción en base a mediciones anteriores.

Cuando se utiliza Machine Learning para construir los modelos, el algoritmo utiliza una base de datos conformada por muchas muestras que contienen observaciones de humedad y viento, estas variables poseen distintos valores categóricos y dependiendo de la recurrencia de esos valores se va a crear el modelo.

**Métricas para validar el modelo**

Para validar un modelo no existe una métrica única que permita comprobar su validez, va a depender de las variables con las que se trabaje y de la construcción del set de datos, es por esto que es importante utilizar las métricas correctas para estudiar la validez de un árbol de decisión. La estimación de certeza de positivos es la capacidad de el modelo para identificar positivos verdaderos, esto se puede obtener a partir de la matriz de confusión del modelo, se debe indicar que se puede hacer lo mismo con los negativos clasificados [15].

Existen muchas métricas que pueden describir lo bien que un modelo consigue realizar predicciones basándose en la matriz de confusión, entre las principales están la exactitud, la sensibilidad y la especificidad.

$$EXACTITUD = \frac{TP+TN}{TP+FP+TN+FN} \quad (1.2)$$

Esta métrica relaciona los aciertos del modelo respecto al total de resultados, es una buena métrica cuando se tiene un conjunto de datos que tiene una cantidad similar de positivos y negativos.

$$SENSIBILIDAD = \frac{TP}{TP+FN} \quad (1.3)$$

$$ESPECIFICIDAD = \frac{TN}{TN+FN} \quad (1.4)$$

Estas métricas ayudan a identificar la cantidad de datos positivos o negativos correctamente clasificados respecto a los clasificados erróneamente, es una métrica que describe que tan bueno es el modelo clasificado los casos negativos o positivos.

### 1.4.5 ÁRBOL DE DECISIÓN EN R

Para poder implementar árboles de decisión en R se utiliza la librería rpart, esta librería crea árboles de regresión o clasificación según las necesidades y rpart contiene varias funciones relacionadas a la implementación de los árboles de decisiones. La principal función que se utiliza es rpart que lleva el mismo nombre que la librería. A continuación se describe en más detalle el uso de esta librería [3].

- **rpart**

Permite crear un modelo de árbol de decisión basado en los datos entregados, esta función devuelve un objeto que contiene todas las características que describe al modelo de árbol de decisión, con este objeto se pueden realizar varias operaciones o extraer características especificas del árbol como las probabilidades, la cantidad de nodos, etc.

**Sintaxis**

rpart(formula, data, method)

*formula*._ Aquí se colocan las variables de salida y de entrada del modelo, (si se desea crear un árbol de clasificación las variables de salida deben ser valores discretos y no continuos).

*data*._ se selecciona el set de datos con el que se va a trabajar.

*method*._ es el método de predicción que se va a aplicar, aquí se define si el árbol será de clasificación o de regresión; para clasificación se usa el valor class [16].

- **Rpart.plot**

Permite crear un grafico a partir de un modelo generado por rpart, este grafico puede ser de modelos de regresión o de clasificación, posee varias configuraciones estéticas que permiten agregar o quitar la información que se muestra por default [17].

**Sintaxis**

rpart.plot(fit, digits = n)

*fit* ._ es el resultado de la función rpart, el objeto que devuelve la función ingresa aquí y es graficado para poder observar sus reglas.

*Digits* ._ permite establecer el numero de decimales que van a tener las probabilidades que se muestren en el gráfico.

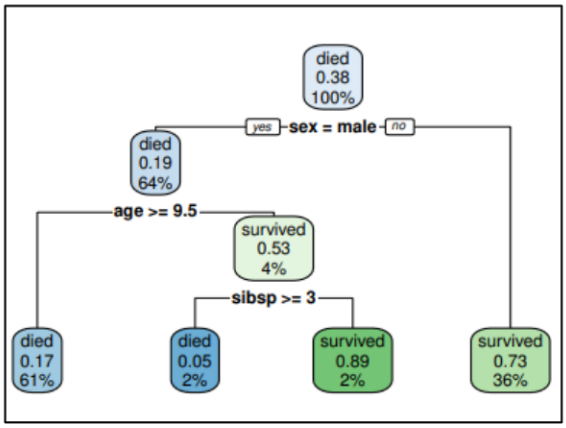


**Figura 1.8.** Ejemplo de la función rpart.plot [17]

En la figura 1.8 se puede observar cómo se grafica un modelo de clasificación binario en donde las probabilidades utilizan 2 cifras significativas esto equivale al parámetro digits = 2 .

- **Rpart.rules**

Esta función permite visualizar las reglas que conllevan a las predicciones del modelo, esto quiere decir que se van mostrar las probabilidades de ocurrencia de los eventos a predecir y las condiciones de las variables para que se de esa probabilidad de

ocurrencia, esta función simplemente recibe el modelo del árbol generado por rpart y muestra las reglas correspondientes [17].

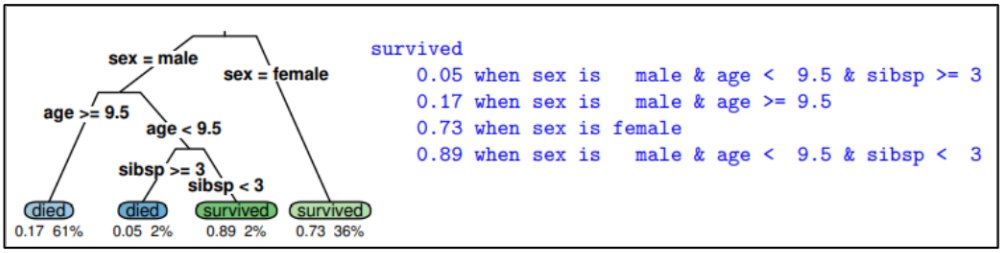


**Figura 1.9** Reglas de árbol de decisión [17].

En la figura 1.9 se puede observar un modelo que predice si los pasajeros del titanic van a sobrevivir utilizando el sexo, edad y el numero de familiares que esa persona tenia en el bote(sibsp), y se puede observar a la derecha las probabilidades que tiene para sobrevivir, de ello se obtiene que para que exista el 89% de probabilidades de supervivencia se requiere que se cumpla que es hombre, menor de 9.5 años y con menos de 3 familiares en el bote.

# 2 METODOLOGÍA

## 2.1 ETAPA DE ADQUISICIÓN DE DATOS

### 2.1.1 OBJETIVOS DE LA ETAPA DE ADQUISICIÓN DE DATOS

1. Obtener parámetros de la red celular que ayuden a identificar situaciones problemáticas durante el proceso de handover
2. Obtener un registro característico de la cobertura de la zona
3. Obtener una base de datos que pueda servir para futuros estudios en redes celulares

### 2.1.2 PARÁMETROS

Para identificar los parámetros que se van a estudiar se realizan mediciones rápidas de prueba en cualquier punto para ver como almacena los datos la aplicación, debido a que las aplicaciones de este tipo deben funcionar en todos los dispositivos Android hay características que no siempre funcionan en todos los dispositivos y llevan a tener valores nulos en algunas variables.

Para esto se realizan mediciones de prueba para identificar los parámetros que este dispositivo concretamente puede almacenar, luego de discriminar las variables que no se pueden estudiar debido a limitaciones del dispositivo queda discriminar variables que no cambian durante las mediciones, estas variables como el nombre de el operador,

estado de la sim, estado del servicio, código del operador, roaming, tecnología de la red, vecinos de tecnologías ajenas a LTE.

Además hay que tener en cuenta que la aplicación cuando no puede almacenar un valor o no lo encuentra en el dispositivo este no se almacena como un valor nulo directamente sino que se guarda con una constante que representa este nulo, el valor de esta constante es : 2147483647 como se puede observar en la figura 2.1.

| nrpci | nrarfcn | rssi | rsrq | rssi_ev | ecio_ev | rssnr | nrssrsrp | nrssrsrq |
|---|---|---|---|---|---|---|---|---|
| 2147483647 | 2147483647 | -103 | -11 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -106 | -10 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -106 | -10 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -106 | -11 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -106 | -11 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -106 | -11 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -103 | -9 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -103 | -9 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -103 | -9 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -102 | -9 | 2147483647 | 2147483647 | 14 | 2147483647 | 2147483647 |
| 2147483647 | 2147483647 | -102 | -9 | 2147483647 | 2147483647 | 17 | 2147483647 | 2147483647 |

**Figura 2.1** Registros de la aplicación

Este valor se va a tomar en cuenta como un nulo que aparece ciertas ocasiones cuando surge un problema en el dispositivo y también permite descartar variables ya que si una variable tiene en su mayoría este valor no puede considerarse para el procesamiento ya que no aporta con información relevante para el estudio.

De esta manera, teniendo en cuenta los parámetros que más varían y las variables que participan en un handover se pueden seleccionar las correctas, en el caso de la hora del sistema se descarta debido que para que la hora del día sea relevante en el análisis se debe tener una base de datos que contenga varias muestras a todas horas del día y la metodología planteada no permite realizar un análisis a todas horas del día, debido a la poca cantidad de información que aporta la hora del día ya que las mediciones se hacen alrededor de la misma hora se descarta esta variable. Finalmente se tiene una lista de parámetros relevantes para el análisis de el fenómeno.

Estos parámetros son:

- sys_time (Corresponde a la hora y fecha del sistema en el momento de tomar las muestras)
- lte_neighbors (Número de vecinos LTE en el punto)
- rssi_strongest (Potencia del vecino más alta)
- rssi (Potencia de la celda actual)
- lat y long (Posicionamiento del dispositivo)
- speed (Velocidad del dispositivo)

Los parámetros seleccionados son variables en cada punto de medición y representan un posible factor para analizar los problemas de las llamadas en la zona de estudio.
Para identificar una zona de estudio es necesario identificar las características que debe cumplir esta zona, una de ellas es que sea una zona donde sean frecuentes las interferencias o problemas durante las llamadas VoIP, teniendo en cuenta que las fallas en las llamadas pueden darse debido a una señal débil deben descartarse los puntos donde a pesar de tener problemas en la llamada VoIP la señal sea débil ya que lo que se busca es encontrar los fallos en la llamada debido a un proceso de handover, para conseguir esto la segunda característica de la zona debe ser que tenga una señal suficientemente alta (mayor a 100 [dbm]).

Para esta parte del estudio solo será necesario realizar una llamada VoIP en distintos puntos de las cercanías para encontrar puntos donde se presenten interferencias y para esto es necesario el uso de una aplicación que ofrezca el servicio de VoIP. La aplicación seleccionada para este estudio es whatsapp ya que es una de las mas utilizadas para realizar llamadas VoIP, la otra aplicación que se va a utilizar es Network Monitor, no se van a almacenar datos de los recorridos únicamente se va a visualizar el nivel de la señal durante la llamada y así corroborar que a pesar de un buen nivel de señal den interferencias en la comunicación.
Es importante tener en cuenta que existen dos dispositivos durante la comunicación VoIP en este caso uno conectado a la red celular y otro conectado a la red doméstica, el dispositivo conectado a la red doméstica va a contar con una mayor estabilidad lo que reduce los errores de la llamada debido a este dispositivo.

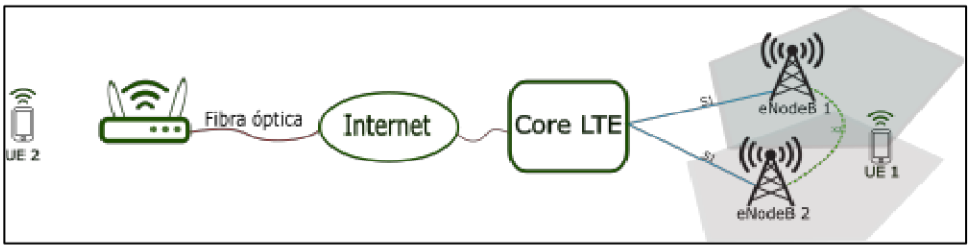


**Figura 2.2.** Diagrama de llamada VoIP

Como se puede ver en la figura 2.2 el UE 2 se encuentra conectado a una red domestica donde es el único dispositivo conectado y se encuentra siempre a una distancia de 1 [m] del punto de acceso, esto permite reducir mucho las interferencias producidas por este dispositivo durante la llamada y reducir errores en la base de datos.
El UE 2 debe tener la llamada de whatsapp en primer plano y ninguna aplicación ejecutándose en segundo plano para evitar que estas afecten con trafico innecesario durante la llamada, es importante que el dispositivo se encuentre conectado todo el

tiempo y no utilice la batería para que de esta manera se evite que las configuraciones de ahorro de energía interfiera con la comunicación entre el UE 2 y el punto de acceso.

### 2.1.3 ZONA DE ESTUDIO

La zona donde se realizará la recolección de datos corresponde a la parroquia de Tababela en un sector de flujo vehicular de alta velocidad.

La zona se seleccionó, ya que es un lugar donde se presentan casos de handover fallido. En esta zona se evaluó la calidad de una llamada usando la red de datos mediante una conexión de VoIP. Se tomaron muestras durante la llamada y si se daban cortes en la comunicación se almacenaba la posición para luego correlacionarla con la base de datos almacenada en el dispositivo.

Luego de varias pruebas en distintas zonas de las cercanías del punto escogido, se encontró un sector en el que se daban estos errores con mucha más frecuencia que otros. Se trata de los alrededores del redondel de Tababela (Figura 2.3), en donde a pesar de existir cobertura se dan eventos de fallos en las llamadas.

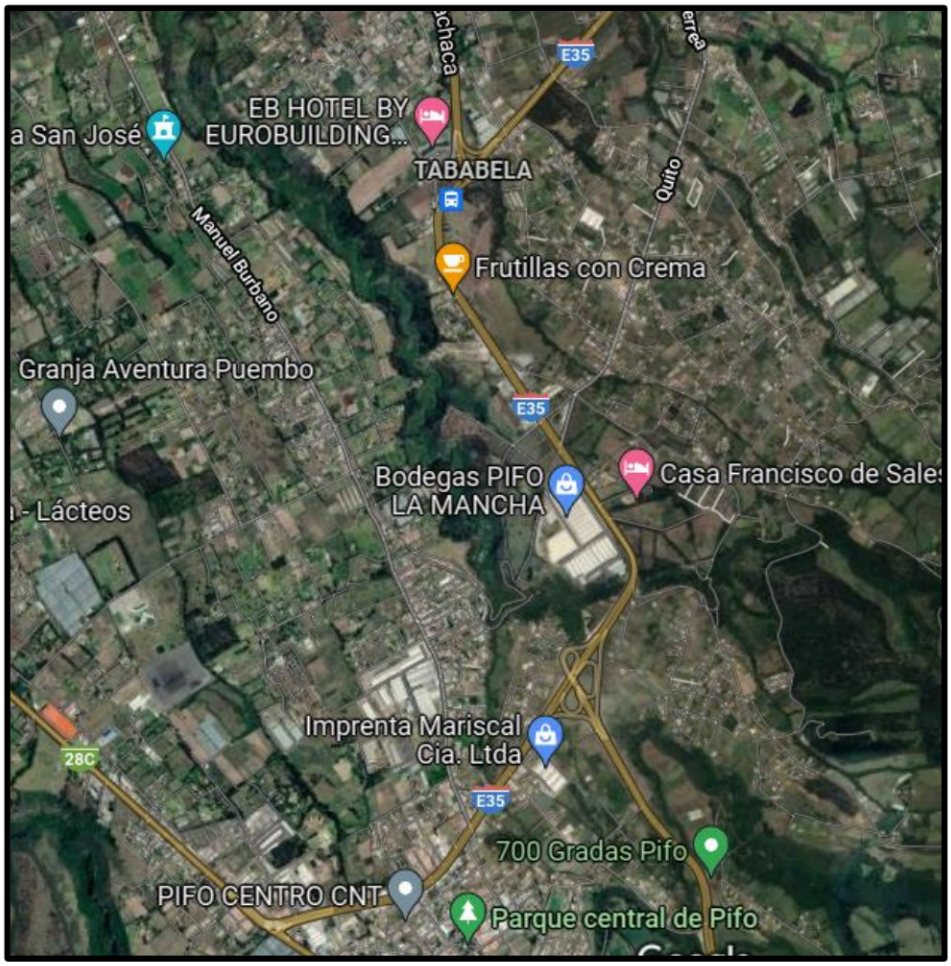


**Figura 2.3.** Zona de estudio

Mediante los datos almacenados por el dispositivo es posible conocer las celdas que cubren esta zona, que son las siguientes:

**Tabla 2.1** Celdas presentes en la zona de estudio.

| Long Cell ID | e_NodeB | Cell ID |
|---|---|---|
| 7233726 | 28256 | 190 |
| 7233728 | 28256 | 192 |
| 7212991 | 28175 | 191 |
| 7212992 | 28175 | 192 |
| 7391422 | 28872 | 190 |
| 7284414 | 28454 | 190 |
| 7284415 | 28454 | 191 |
| 7391422 | 28872 | 190 |
| 7185599 | 28068 | 191 |
| 7333312 | 28645 | 192 |
| 7333310 | 28645 | 190 |

En la zona de estudio en total el dispositivo tiene conexión con 11 celdas para mantener la llamada. El conjunto de datos de donde se extraen las celdas, ya han sido filtrados con un criterio de potencia mínima, por lo que las celdas que se muestran en la tabla anteriormente, solo son las que efectivamente presentan un nivel de señal suficiente (mayor a 100 [dbm]).

También se puede observar que existen 6 e-NodeB a los que el dispositivo se conecta, es de gran utilidad conocer la ubicación de estas antenas, y para esto, se utiliza la información disponible en cellmapper, que permite ubicar y/o conocer las antenas en función de medidas o del reporte de usuarios.

Con la información de la ubicación de los nodos importantes se puede proponer un mapa de rutas para tomar medidas de manera sistemática, y así, adquirir información variada y que permita contar con los datos necesarios y suficientes a ser analizados.

En la figura 2.4 se proponen las rutas a considerar.

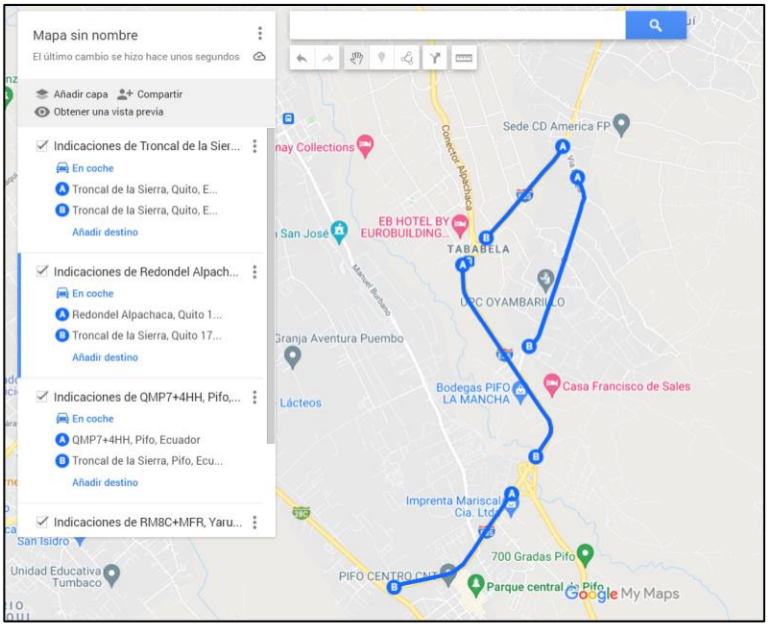


**Figura 2.4.** Rutas consideradas para la ejecución de las mediciones

Tabla 2.1 Descripción de las rutas

| **Ruta** | **Coordenada Inicio** | **Coordenada final** | **Distancia recorrida [km]** |
|---|---|---|---|
| Ruta 1 | -0.17965<br>-78.32982 | -0.18924<br>-78.33799 | 1,39 |
| Ruta 2 | -0.19153<br>-78.33992 | -0.212<br>-78.33297 | 2,48 |
| Ruta 3 | -0.21511<br>-78.33501 | -0.22477<br>-78.34735 | 1,86 |
| Ruta 4 | -0.18241<br>-78.32868 | -0.20054<br>-78.33379 | 2,23 |

Las rutas van a ser exploradas en ambos sentidos, lo que permite duplicar los datos en un mismo día; pero se almacenarán en sesiones separadas, ya que eso permite mantener a las zonas distinguibles a nivel de data set.

Con la ayuda de cellmapper es posible encontrar las antenas y las zonas de cobertura de sus celdas para encontrar los puntos en los que existen solapamientos de celdas que servirá en el análisis de los procesos de handover.

Este procedimiento requiere de una fase previa de exploración en donde se recorran las rutas tomando medidas con la aplicación cellmapper pero únicamente con el propósito de que se genere un mapa de celdas en la zona de estudio.

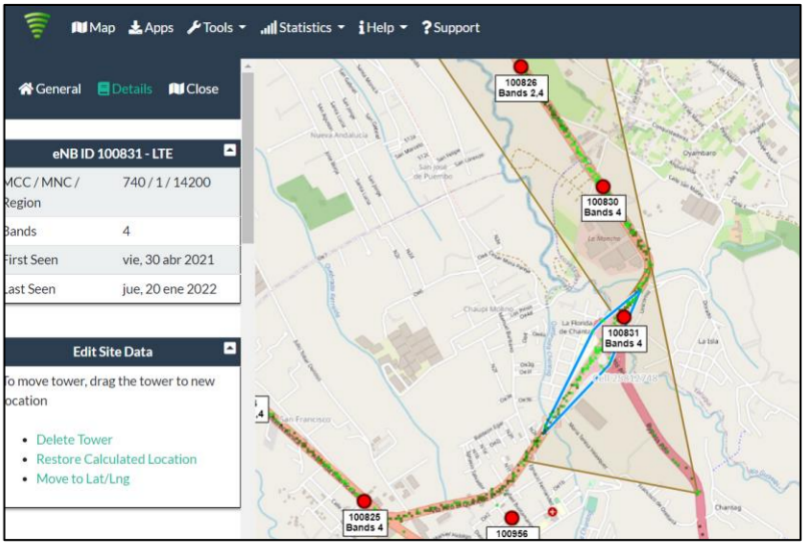


**Figura 2.5.** Mapa de celdas de la zona

Como parte de la etapa de recolección de datos, primero es necesario otorgar una información a la aplicación cellmapper para que se puedan ubicar de mejor manera las antenas y sus celdas, para esto hay que realizar mediciones de cobertura en las rutas seleccionadas sin separarlas en sesiones y subir esta información a la pagina de cellmapper.

En la figura 2.5 se puede observar que dentro de un eNodeB existen 3 celdas cubriendo la ruta de estudio y se pueden apreciar los puntos donde se solapan y existe mayor probabilidad de handover, entre mas mediciones se hagan en la ruta mas detallados van a ser estos mapas de celdas sin embargo no es necesario crear un mapa de gran detalle ya que solo se requiere la referencia de los puntos donde se solapan las celdas. Los puntos rojos que se pueden apreciar en la figura 2.5 son los eNodeB que cubren la zona de estudio y esta información es útil al momento de hacer las mediciones de campo ya que permite tener en cuenta los puntos críticos donde dos eNodeB se encuentran en sus limites.

Para crear una base de datos completa se van a utilizar otras 2 bases de datos menores, estas se van a tomar en cada ruta las dos corresponden a las bases de datos que van a crecer con cada sesión de adquisición de datos, una es el posicionamiento de los puntos donde existen fallas y otra los datos ofrecidos por Network monitor en los que se incluyen los datos que se utilizarán en el presente análisis y que se mencionaron al inicio de esta sección.

**Toma de muestras en las rutas**

Para la toma de muestras en las zonas elegidas es importante comprender el manejo de las herramientas digitales que se van a utilizar, hay que tener en cuenta que el UE se puede enlazar a la tecnología que más le convenga para llevar a cabo las llamadas y esto produce que exista un handover entre distintas tecnologías, sobre todo que el UE

cambie de LTE a HSPA o HSPA+, los datos de este tipo de handover no son de interés para el presente trabajo por lo que deben descartarse estos eventos y una manera de eliminar esta posibilidad es configurar al dispositivo de manera que solamente se conecte a celdas LTE y de esta manera si se presenta un handover será entre dos celdas LTE y no se presentarán otras tecnologías.

Para conseguir esto lo primero que se hace es configurar al dispositivo para conectarse solo a celdas LTE, como se dijo anteriormente el dispositivo cuenta con un sistema operativo Android, en dispositivos con este sistema operativo existen códigos alfanuméricos que se introducen desde el panel de llamada y permite acceder a un menú particular que corresponde a configuraciones muy especificas que normalmente no se encuentran disponibles para el usuario a través de la interfaz básica.

Estos códigos suelen ser genéricos y no siempre dependen del fabricante sin embargo, existen diversas posibilidades y es mejor utilizar aplicaciones móviles capaces de ingresar estos códigos dependiendo del modelo del dispositivo, en la tienda de aplicaciones existen varias opciones que permiten esto y en este caso se va a utilizar “Force LTE 4G” que realiza este proceso internamente y con ello ya no es necesario ingresar códigos manualmente, esto también es útil ya que el dispositivo debe regresar a su configuración inicial luego de finalizar cada sesión por lo que resulta mas ágil utilizar una aplicación que permita configurar y restaurar los parámetros del dispositivo.

Se va a utilizar Network Monitor para registrar los datos de el estado de la señal en diversos puntos, es necesario comenzar una sesión nueva luego de haber recorrido una ruta de ida y regreso, esto duplica la posibilidad de encontrar fallos en la misma ruta y separa los datos medidos de manera que sea fácil clasificarlos posteriormente.

Luego que la llamada se realiza y Network Monitor ya esta registrando los puntos se puede utilizar GPS Logger para registrar la posición cada que se tenga una interferencia en la llamada, de esta forma existen 3 aplicaciones corriendo simultáneamente, whatsapp, Network Monitor y GPS Logger, con la primera se mantiene la llamada, la segunda registra los puntos en segundo plano y la tercera se encuentra en primer plano y registra los puntos geográficos donde ocurren las interferencias.

El uso de dos aplicaciones separadas para este estudio se debe a que Network Monitor no permite marcar puntos para señalar particularidades en ellos, no hay forma de indicarle a la aplicación que un punto determinado presenta un inconveniente sin embargo, la aplicación registra latitud y longitud de cada punto medido y de esta forma resulta útil almacenar la latitud y longitud de los puntos de interferencia con otra aplicación y luego utilizar estos puntos para correlacionarlos con la base de datos de mediciones.

Las llamadas VoIP en whatsapp están optimizadas principalmente para transmisión de voz es por esto que se debe establecer un método que permita reducir los silencios en las llamadas y tener un punto de referencia constante durante la llamada, ya que se van a hacer varias sesiones en días distintos es importante escoger una fuente de sonido que se va a reproducir en el UE 2 que se encuentra en la red doméstica para que sea escuchado en el UE 1 que se encuentra en movimiento.

Como punto de referencia se toma un audio de una entrevista ya que durante las entrevistas normalmente los silencios son cortos y solamente se tiene voz, es importante señalar que la fuente de sonido no puede pertenecer al UE2 que se encuentra en la red doméstica sino que debe ser de una fuente externa, esto para evitar que el dispositivo lo pueda considerar como ruido y lo elimine al originarse en el mismo. Esta fuente de sonido reproduce el audio a un volumen constante y a una distancia de 30 [cm] del UE.

Las medidas que toma la aplicación Network Monitor se toman cada segundo y es el valor más bajo que se puede configurar por lo que se debe tener en cuenta que a velocidades muy altas puede perderse información por lo que es importante controlar la velocidad a la que el dispositivo viaja, en este caso se plantea un límite de 23 [m/s] que equivale a 82.8 [km/h] cuando se viaja a esta velocidad la aplicación va a tomar registros cada 23 metros lo que resulta ser una distancia grande sin embargo a nivel de cobertura celular no es tan significativa por lo que representa un limite adecuado, las velocidades muy bajas implican sistemas más estables lo que reduce la probabilidad de que se presente una interferencia.

**Tabla 2.2** Cronograma de la etapa de adquisición y análisis de datos

| Actividad | Diciembre 2021 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 06 | 07 | 08 | 09 | 10 |
| Adquisición de datos para cellmapper en todas las rutas | ■ | ■ | ■ | ■ | ■ | | | | |
| Comprobación de las ubicaciones de las torres | | | | | | ■ | ■ | | |
| Sesión 1 de todas las rutas con Network Monitor y GPS logger | | | | | | | | ■ | |
| Sesión 2 de todas las rutas con Network Monitor y GPS logger | | | | | | | | | ■ |

| Actividad | Diciembre 2021 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 11 | 13 | 14 | 15 | 16 | 17 | 18 | 20 | 21 |
| Sesión 3 de todas las rutas con Network Monitor y GPS logger | ■ | | | | | | | | |
| Sesión 4 de todas las rutas con Network Monitor y GPS logger | | ■ | | | | | | | |
| Sesión 5 de todas las rutas con Network Monitor y GPS logger | | | ■ | | | | | | |
| Sesión 6 de todas las rutas con Network Monitor y GPS logger | | | | ■ | | | | | |
| Sesión 7 de todas las rutas con Network Monitor y GPS logger | | | | | ■ | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Sesión 8 de todas las rutas con Network Monitor y GPS logger | | | | | | ■ | | | |
| Sesión 9 de todas las rutas con Network Monitor y GPS logger | | | | | | | ■ | | |
| Sesión 10 de todas las rutas con Network Monitor y GPS logger | | | | | | | | ■ | |
| Creación de la base de datos principal en base a las mediciones recolectadas en las sesiones. | | | | | | | | | ■ |

| Actividad | Enero 2022 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | |
| Tratamiento de los datos obtenidos | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |

## 2.2 TRATAMIENTO DE LOS DATOS

Para conseguir el objetivo de este trabajo se plantea analizar la base de datos recolectada mediante el lenguaje “R” de programación que tiene diversas herramientas estadísticas que facilitan el estudio con los datos. Estos datos van a ser procesados y analizados utilizando R Studio que es un IDE para el lenguaje R.

Para procesar los datos, primero se deben definir las variables de entrada y salida que va a tener el modelo; esto se debe realizar ya que es necesario establecer una relación entre las mediciones y los efectos producidos, en este proyecto la variable de salida es la interferencia (o interrupción de la conexión durante un handover) que podrá tomar dos valores “Si” y “No”.

Como datos de entrada se deben considerar todos los datos que ofrece la aplicación con la que se hizo la recolección y luego determinar los que pueden tener un efecto causal sobre las interferencias durante handover. Con esta base se toman las siguientes variables de entrada: lte_neighbors, rssi_strongest, rssi, rsrq, rssnr [18].

En la etapa de recolección de datos se obtuvieron dos tipos de datasets, por un lado se tienen las mediciones de la señal y por el otro el registro de los puntos geográficos donde se presentaron interrupciones en la llamada mediante VoIP que en nuestro caso las llamamos interferencias , con esto en cuenta, lo que se busca es integrar la información de interferencia y las medidas de campo en una sola base de datos que contenga toda la información de las pruebas realizadas.

Es necesario aclarar que la variable de salida no viene incluida inicialmente en los datos medidos, es una variable que se obtiene de acuerdo al registro de los datos de la base de datos de interferencias. Debido a que la base de datos de señal contiene datos de latitud y longitud y la base de datos de interferencia también, se pueden usar estos datos para correlacionar ambas bases de datos y generar una columna de la variable interferencia (o interrupción) en una base de datos total, el método para correlacionarlos

es por comparación de longitud, latitud y hora del sistema, si estos tres parámetros coinciden entonces el valor de la variable interferencia de esa observación será: “si”. Este proceso de correlación se puede ejecutar mediante R comparando longitud, latitud y hora del sistema en ambas bases de datos y generando una base de datos total con la variable interferencia, a esta base de datos total se le deben agregar las variables de entrada desde la base de datos de señal y así se obtiene la base de datos total con la que se va a trabajar. En la siguiente figura se describe el proceso de obtención de la base de datos a analizar.

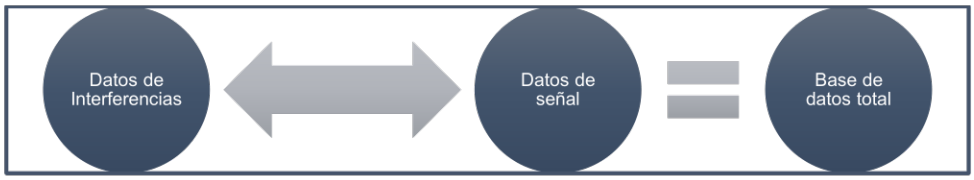


**Figura 2.6.** Correlación de bases de datos

Esta base de datos total puede contener datos que las aplicaciones comúnmente almacenan como “NA” , en el caso de la aplicación celular utilizada en el presente trabajo se genera un valor entero igual a “2147483647” cuando existen problemas con la medición, es por esto que se deben filtrar todos los datos que contengan este valor eliminando esas mediciones.

Las interferencias que están contenidas en este dataset pueden deberse a un proceso de handover fallido o a una señal pobre en el punto donde se tomó la medición. Para reducir el error que corresponde a esta última situación se van a eliminar los datos que contengan un rssi muy bajo para solamente mantener aquellos que correspondan a puntos con una buena calidad de señal, siendo así la única posible causa de fallo el handover. Todos los puntos de medición con un rssi menor a 100 dBm son eliminados para descartar las interferencias (o interrupciones) por señal pobre [18].

Se puede observar que en los datos recolectados se tiene un porcentaje bajo de puntos de handover respecto a puntos donde no se está ejecutando dicho proceso y esto se debe a que para preservar la calidad de la llamada el sistema celular evita en lo posible realizar un handover a menos que sea estrictamente necesario; es por eso que los datos en su mayoría contienen información de procesos donde no se está efectuando un handover. Los datos que no están relacionados con el handover no ayudan a predecir lo que sucede durante este proceso, por lo que deben ser descartados y solo aportan al modelo aquellas observaciones en donde se está realizando un handover entre dos celdas. Para discriminar estos dos tipos de datos es necesario identificar en que puntos se realiza un handover durante las mediciones, lo cual se presenta en la figura 2.7.

Para determinar si se produjo un handover se puede hacer uso de un dato especifico que es recolectado por la aplicación, este dato es el identificador de la celda; esta

consideración va a consistir en lo siguiente: si se presenta que en observaciones consecutivas, el identificador de la celda cambia, esto quiere decir que se acaba de producir un handover, este proceso se puede automatizar en R para discriminar los datos útiles para el estudio y seleccionarlos para el modelo.

| | X | Y | Z | AA | AB | AC | AD |
|---|---|---|---|---|---|---|---|
| 1 | mnc | mnc_master | lac_tac_sid | long_cid | node_id_nid | cid_bid | psc_pci |
| 2 | 1 | 1 | 14200 | 25811979 | 100828 | 11 | 3 |
| 3 | 1 | 1 | 14200 | 25811979 | 100828 | 11 | 3 |
| 4 | 1 | 1 | 14200 | 25811979 | 100828 | 11 | 3 |
| 5 | 1 | 1 | 14200 | 25811980 | 100828 | 12 | 4 |
| 6 | 1 | 1 | 14200 | 25811980 | 100828 | 12 | 4 |

**Figura 2.7.** Datos de evidencia de la ejecución de un handover

Luego de todos los procesos realizados la base de datos final tiene únicamente los datos de las variables de entrada y la de salida de los puntos donde se ha realizado un handover. Una parte de los datos finales se muestra en la figura 2.8.

| lte_neighbors | rssi_strongest | rssi | rsrq | rssnr | speed | interf |
|---|---|---|---|---|---|---|
| 1 | -93 | -92 | -15 | 12 | 18.42 | no |
| 1 | -97 | -91 | -11 | -1 | 18.54 | no |
| 3 | -88 | -92 | -18 | 6 | 16.82 | si |
| 3 | -88 | -87 | -14 | 6 | 16.86 | si |
| 3 | -88 | -87 | -14 | 7 | 17.36 | no |
| 2 | -93 | -96 | -16 | -1 | 17.18 | no |

**Figura 2.8.** Matriz con variables y datos finales

Antes de la implementación en R es necesario aislar los datos medidos y clasificarlos en carpetas de manera ordenada, ya que esto será útil al momento de cargar los datos al programa. Para cumplir esto, se crea una carpeta para almacenar las rutas y dentro de ellas se crean 4 subcarpetas correspondientes a cada una de las rutas con el formato "Ruta#", donde # representa el número de ruta que corresponda.

Dentro de las carpetas de cada ruta se van a cargar las mediciones, tanto de señal como de interferencia, ya que estas mediciones se han realizado en distintas sesiones para cada ruta; además, es necesario etiquetarlas con el siguiente formato: R#S#S o R#S#P, correspondiendo la primera letra a la inicial de Ruta, la segunda a la inicial de sesión y la última es un identificador del tipo de dato que guarda el archivo sea señal o posición. Dentro de los archivos de señal se encuentran las mediciones de Network Monitor y dentro de los archivos de posición se encuentran las localizaciones guardadas con GPSLogger.

### 2.2.1 IMPLEMENTACIÓN EN R

Para implementar el modelo en R, lo primero que se realiza es establecer un directorio de trabajo, donde se almacenan las mediciones realizadas. Se tiene una carpeta para cada ruta y en cada ruta hay 10 archivos csv que contienen las mediciones y otros 10 que contienen los puntos de interferencia para cada ruta, en total se tienen 80 archivos que se deben importar a RStudio, para importarlos se va a realizar una función recursiva.

```
setwd("/Users/crist/Documents/Tesis/datos_medidos/Net_monitor/Rutas")
datos <- lapply(Sys.glob("Ruta?/R?S*S.csv"), read.csv)
pos <- lapply(Sys.glob("Ruta?/R?S*P.csv"), read.csv)
```

**Figura 2.9.** Importación de datos a R

La función setwd permite establecer el directorio de trabajo donde se encuentran los datos de las mediciones, hay que aclarar que normalmente las rutas en Windows se escriben separando directorios con el símbolo "\"(backslash); sin embargo, para referenciar la ruta en R se utiliza slash normal "/". Luego se utiliza la function lapply que va a crear una lista de matrices, esta es una estructura de datos que permite anidar varias matrices dentro de una lista, de esta forma se tienen todas las mediciones de señal en la variable "datos", mientras que en la variable "pos" se van a almacenar todas las matrices que representan las mediciones de interferencia (o interrupción).

El método para cargar las matrices utiliza expresiones regulares, las cuales permiten realizar búsquedas basadas en símbolos especiales que representan condiciones específicas. En el argumento "Ruta?" se utiliza ? para que se tomen todas las carpetas que tienen como nombre la palabra Ruta y seguido de este un carácter cualquiera, ya que en la carpeta de trabajo se tienen las rutas nombradas como Ruta1, Ruta2, Ruta3, Ruta4, estas carpetas forman parte del conjunto sobre las que va a actuar la función; es debido a esto, que estas carpetas deben estar aisladas, ya que si existe otro archivo que pueda coincidir con la búsqueda el programa lo va a intentar cargar y puede causar un error. La segunda parte del argumento corresponde a "R?S*S.csv" que corresponde al archivo que se va a cargar al programa, esta expresión va a coincidir solamente con los datos almacenados en las carpetas del directorio de trabajo.

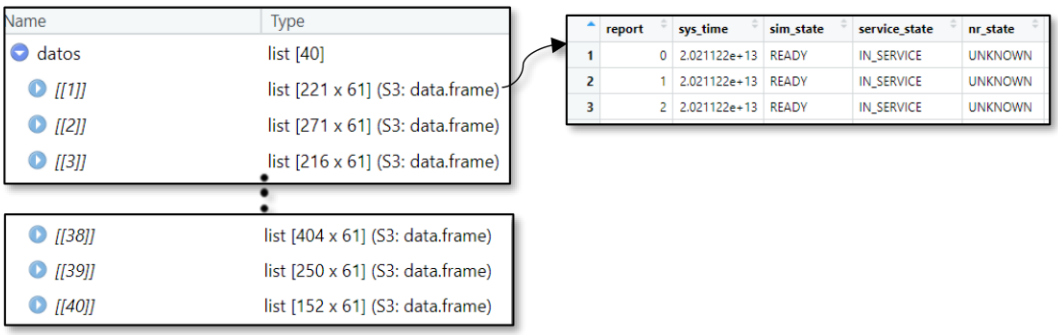

**Figura 2.10.** Estructura de datos importados

De esta manera se crean dos listas de matrices que contienen los datos de señal y los datos de posición; sin embargo, para crear el modelo que se requiere se debe tener una única matriz con todos los datos recolectados y que contenga las variables de entrada y salida que se necesitan.

A este set de datos se les debe agregar dos columnas, la columna de interferencia (o interrupción) y una de handover, ambas pueden tomar dos valores “si” y “no”, el valor que tome la columna de interferencia va a depender del proceso de correlación con la base de datos de posición. Este proceso, como se mencionó anteriormente, se lleva a cabo buscando coincidencias entre los datos de la base de datos de señales y la de posición (de la interferencia o interrupción), siempre que haya una coincidencia el valor de la columna “interferencia” cambiará a “si”, ya que por defecto todas las columnas tienen el valor “no”. En cuanto al handover se realiza un lazo for para realizar la comparación descrita anteriormente. El código utilizado para correlacionar las bases de datos de interferencia con la de señales y para establecer la ejecución del proceso de handover se muestra en la siguiente figura.

```
j<-length(datos)
for (i in 1:j) {
  datos[[i]]$interf <- "no"
  datos[[i]]$handover <- "no"
  for (h in 1:nrow(pos[[i]])) {
    for (m in 1:nrow(datos[[i]])) {
      a <- datos[[i]]$lat[m]
      b <- pos[[i]]$latitude[h]
      if(m>1){
        c <- datos[[i]]$long_cid[m]
        d <- datos[[i]]$long_cid[m-1]
      }
      else{
        c=1
        d=1
      }
      if(a == b)
      {
        if(m>4){
          datos[[i]]$interf[m-2] <- "si"
          datos[[i]]$interf[m-1] <- "si"
          datos[[i]]$interf[m] <- "si"
        }
        else{
          datos[[i]]$interf[m] <- "si"
        }
      }
      if (c != d) {
        datos[[i]]$handover[m]<-"si"
        datos[[i]]$handover[m-1]<-"si"
```

**Figura 2.11.** Correlación de las bases de datos y handover

Luego de que los datos de interferencia y handover ya se disponen se pueden unir todas las matrices contenidas en la lista para crear una única matriz (figura 2.12) con la que se va a trabajar para limpiar los datos y seleccionarlos bajo los criterios descritos en la sección anterior. A continuación se describe este código.

```
total <- do.call("rbind",datos)
total <- filter(total, handover == "si")
total <- select(total,c(lte_neighbors, rssi_strongest, rssi, rsrq, rssnr, speed, interf))
total[total == 2147483647] <- NA
total <- filter(total, rssi >= -110)
na.omit(total)
```

**Figura 2.12.** Creación de la base de datos final

El comando do.call permite repetir una función a lo largo de varios datos y se utiliza en este caso con la función rbind que permite unir las filas de varios archivos en una misma matriz, por lo que se aplica a todas las matrices del conjunto seleccionado de datos. En la siguiente línea de código se eliminan todos los datos que no corresponden a un handover, ya que son filtrados para disponer solamente de los datos donde existe un handover. Después se usa el comando select [19] que permite seleccionar únicamente las columnas que son de interés, aquí deben estar las variables de entrada y la variable de salida. Para eliminar el valor nulo igual a 2147483647 se los convierte en NA, esto para luego usar la función que suprime los NA. Por último se filtran las observaciones que no tengan un rssi mayor que -110, esto para evitar que las interferencias por señal baja aparezcan. Una parte de la matriz final se muestra en la siguiente figura.

| lte_neighbors | rssi_strongest | rssi | rsrq | rssnr | speed | interf |
|---|---|---|---|---|---|---|
| 4 | -86 | -72 | -9 | 19 | 16.66 | no |
| 2 | -96 | -89 | -9 | 9 | 13.42 | no |
| 4 | -104 | -102 | -13 | 1 | 18.04 | no |
| 3 | -95 | -97 | -20 | -5 | 19.70 | no |

**Figura 2.13.** Matriz con valores filtrados

Se tiene como resultado una matriz que contiene las variables que se necesitan y la que se puede procesar con la función que crea los árboles de decisión, para esto se va a utilizar rpart. Antes de enviar la matriz a la función es necesario realizar una división de los datos, esto para contar con dos datasets, uno para entrenar al modelo y otro para ponerlo a prueba, para esta situación se van a tomar el 80% de los datos para el entrenamiento y el restante 20% va a ser para probar el modelo.

```
fit <- rpart(interf ~ ., data = train, method = "class", minsplit = 10, minbucket = 10)
fancyRpartPlot(fit, caption = NULL)
prueba <-predict(fit, test, type = 'class')
matriz <- table(test$interf, prueba)
sensib <- (matriz[4]/(matriz[2]+matriz[4]))
```

**Figura 2.14.** Creación del modelo

Al modelo que va a almacenar la estructura del árbol se le nombra como fit, esta es la salida de la función rpart que en este caso cuenta con “interf” como la variable a predecir, esto hará que el árbol trate de clasificar todos los datos para determinar si en diversas circunstancias se podría producir una interferencia o interrupción durante un handover. El siguiente parámetro que se define es la matriz de entrenamiento que es igual al 80% de los datos de la matriz total, el método que se va a utilizar es “class”, ya que se trata de un árbol de clasificación. Minsplit es la cantidad mínima de datos que se necesitan para crear una rama, y minbucket la cantidad mínima que debe haber en un nodo.

Manipulando estos datos se puede encontrar una relación entre la complejidad del árbol resultante y la sensibilidad al momento de predecir.

A continuación en la figura 2.15 se crea un lazo for para que el modelo se ejecute con distintos valores de estos parámetros, y se obtiene de ellos el tamaño del árbol y la sensibilidad que produce.

```
k=1
for (j in 1:10) {
  for (i in 1:10) {
    fit <- rpart(interf ~ ., data = train, method = "class", minsplit = i, minbucket = j)
    prueba <-predict(fit, test, type = 'class')
    matriz <- table(test$interf, prueba)
    sensib <- (matriz[4]/(matriz[2]+matriz[4]))
    valores[k]<-sensib
    tam[k]<-length(fit$frame$var)
    k=k+1
```

**Figura 2.15** Bucle de evaluación de los parámetros del árbol

De aquí se puede obtener una relación que permita elegir el mejor valor, y con esto la mejor versión que se puede obtener del árbol.

Se obtienen dos matrices, una que representa la sensibilidad del modelo y otra que representa el tamaño del árbol.

```
> valores
  [1] 0.12 0.12 0.12 0.22 0.22 0.16 0.16 0.06 0.06 0.06 0.22 0.22 0.22 0.22 0.22 0.16 0.16 0.06 0.06 0.06 0.16 0.16 0.16 0.16 0.16
 [26] 0.16 0.16 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06
 [51] 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.06 0.25 0.25 0.25 0.25 0.25
 [76] 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25

> tam
  [1] 379 379 295 233 225 215 177 167 159 143 159 159 159 159 151 135 127 125 119 105 139 139 139 139 139 139 131 127 115 103 101
 [32] 101 101 101 101 101 101 101 101  89  93  93  93  93  93  93  93  93  93  61  61  61  61  61  61  61  61  61  61  61  61  61
 [63]  61  61  61  61  61  61  61  61  21  21  21  21  21  21  21  21  21  21  19  19  19  19  19  19  19  19  19  19  19  19  19
 [94]  19  19  19  19  19  19  19
```

**Figura 2.16**. Matriz con valores de sensibilidad y tamaño del arbol

En la matriz se puede observar que a partir de la iteración 71 se tiene la mayor sensibilidad con un valor de 0.25 que representa a un 25% de positivos correctos, este es el mayor valor que alcanza el modelo y existen varios tamaños que permiten alcanzar este valor, en la segunda matriz se puede ver que a partir de la iteración 71 en adelante los valores de tamaño se reducen.

Para relacionar el número de iteración con el valor correcto de minsplit y minbucket se debe tomar el primer digito del número de iteración y sumarle 1, este valor representa el minbucket, y el segundo dígito del número de iteración representa el valor de minsplit, en este caso se toma el número 71, al primer dígito se le suma 1 y se tiene que el valor de minbucket es 8, el segundo dígito es 1 por lo que este es el valor de minsplit, esto lleva a un árbol que tiene la mayor sensibilidad que se puede obtener del modelo que es del 25%.

Luego de identificar los valores adecuados de los parámetros minsplit y minbucket, ya se los puede reemplazar en la función principal y obtener el modelo de árbol que conforma la predicción de los parámetros de interferencia en zonas donde existe handover.

# 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

## 3.1 RESULTADOS

La función utilizada para crear el modelo devuelve un objeto de función, que es como tal el árbol(figura 3.1), de este objeto se pueden obtener los nodos y ramas en forma de texto con información más explícita.

```
## n= 2659
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 2659 119 no (0.95524633 0.04475367)
##    2) rssi_strongest>=-107.5 2475  94 no (0.96202020 0.03797980) *
##    3) rssi_strongest< -107.5 184  25 no (0.86413043 0.13586957)
##      6) rssi< -108.5 83   1 no (0.98795181 0.01204819) *
##      7) rssi>=-108.5 101  24 no (0.76237624 0.23762376)
##       14) rsrq>=-17.5 86  14 no (0.83720930 0.16279070)
##         28) speed< 17.17 69   5 no (0.92753623 0.07246377) *
##         29) speed>=17.17 17   8 si (0.47058824 0.52941176)
##           58) rssi>=-104 9   3 no (0.66666667 0.33333333) *
##           59) rssi< -104 8   2 si (0.25000000 0.75000000) *
##       15) rsrq< -17.5 15   5 si (0.33333333 0.66666667) *
```

**Figura 3.1.**Árbol de decisión resultante.

Este modelo por si mismo es capaz de explicar el comportamiento de los datos en el experimento; sin embargo, es poco intuitivo y difícil de interpretar inmediatamente; es por este motivo que se acude a la función rpart.plot que permite construir un gráfico a partir de este modelo para que sea más fácil de leer y entender.

En la figura 3.1 se puede observar que comienza con el numero de datos que han sido evaluados, estos datos corresponden al dataframe de entrenamiento, es necesario tener en cuenta que el dataframe inicial antes de clasificar los datos era de 17199 observaciones; y luego de sólo tomar los puntos donde existe handover, los datos se redujeron a 3324 observaciones, esto equivale a un 19,32% de los datos iniciales, tomando este valor de referencia se puede concluir que ese es el porcentaje de ocurrencia de un handover en la zona de estudio.

De aquí se puede entender que el handover es un evento poco frecuente en la zona de estudio durante llamadas de VoIP; y es comprensible, ya que lo que se busca es minimizar eventos inestables que alteren la calidad de la llamada, teniendo en cuenta que el handover es un proceso que aumenta las probabilidades de fallo y por tanto se evita que se produzca con mucha frecuencia.

Del conjunto de datos que sólo contienen eventos de handover se pueden obtener dos histogramas importantes, uno de la variable “rssi_strongest” y el otro de la variable “rssi”, con esto se pueden observar las condiciones en cuanto a nivel de señal que son más frecuentes que sucedan durante un handover en la zona estudiada.

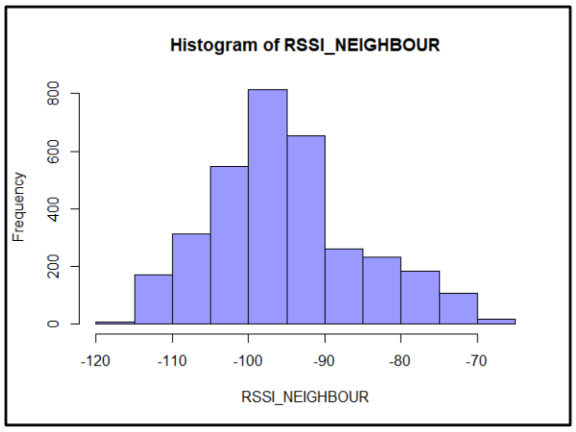


**Figura 3.2.**Histograma de potencia de celda vecina (rssi_strongest) durante el handover

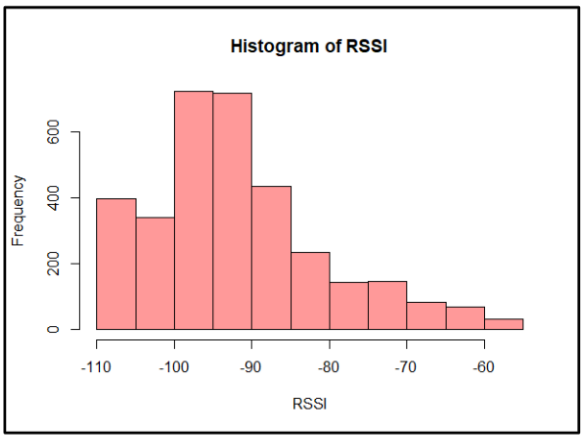


**Figura 3.3.**Histograma de la potencia de la celda servidora (rssi)

Se puede notar un evento importante que corresponde a la situación en la que el handover se debe a que la potencia de la celda vecina es mayor a la potencia de la celda que sirve en ese momento, si en cada observación se resta la potencia de la celda vecina de la potencia de la celda actual se puede obtener un resultado positivo o negativo. Cuando el resultado sea positivo implicará que la potencia de la celda vecina es mayor a la potencia de la celda actual y cuando sea negativo implicará que existe un handover a pesar que la potencia de la celda actual es mayor que la vecina y aun así se da el intercambio de celdas.

Este último caso puede corresponder a otras causas de handover como sobrecarga de la celda actual o calidad de servicio de la celda actual respecto a la vecina; sin embargo, esta información no se recolecta en las muestras, ya que requiere de otros métodos de adquisición.

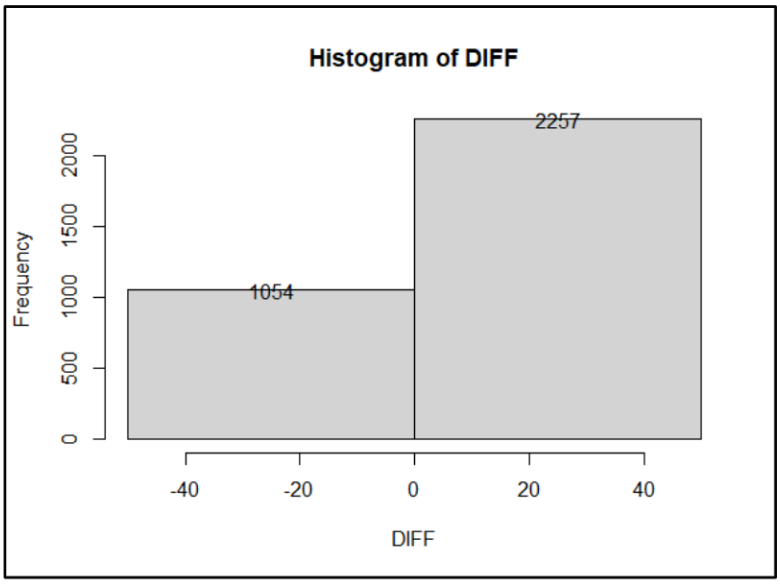


**Figura 3.3.** Resta entre rssi de celda actual y celda vecina

De esta manera si el resultado de la resta es positivo significa que la celda vecina tiene mayor potencia que la celda actual, lo que corresponde a lo que se espera de un handover normal; y si el resultado de la resta es negativo, ya no existe una mejor potencia en la antena vecina; sin embargo, se da el handover. Se puede ver por la cantidad de resultados negativos, en la figura 3.3, que aproximadamente el 32% de las veces el handover se da por motivos distintos al de una potencia mejor en la celda vecina.

### 3.1.1 ANÁLISIS DEL MODELO OBTENIDO

Del árbol de decisión obtenido se pueden extraer varios datos muy relevantes, en la figura 3.4 se puede observar al nodo principal o “raíz” del que parten todas las ramas que tiene el árbol y se puede usar para explicar la información que ofrece el modelo.

```
1) root 2659 119 no (0.95524633 0.04475367)
  2) rssi strongest>=-107.5 2475  94 no (0.96202020 0.03797980) *
```

**Figura 3.4.** Primeros Nodos del árbol

Lo primero que se puede observar es un identificador numérico del nodo y luego la condición que causa una rama, en el caso del nodo raíz no se tiene una condición; pero en el resto si existe condición. La condición va a dividir el conjunto de mediciones en dos grupos y luego de la condición se encuentra la cantidad de datos que pertenecen al nodo actual, y posteriormente, la cantidad de datos donde existe una interferencia (interrupción) durante el handover.

En el nodo raíz se inicia con 2659 observaciones de las cuales sólo 119 corresponden a interferencias durante el handover, se puede ver el porcentaje correspondiente entre paréntesis y es el 4.47% de los datos que ingresaron al modelo.

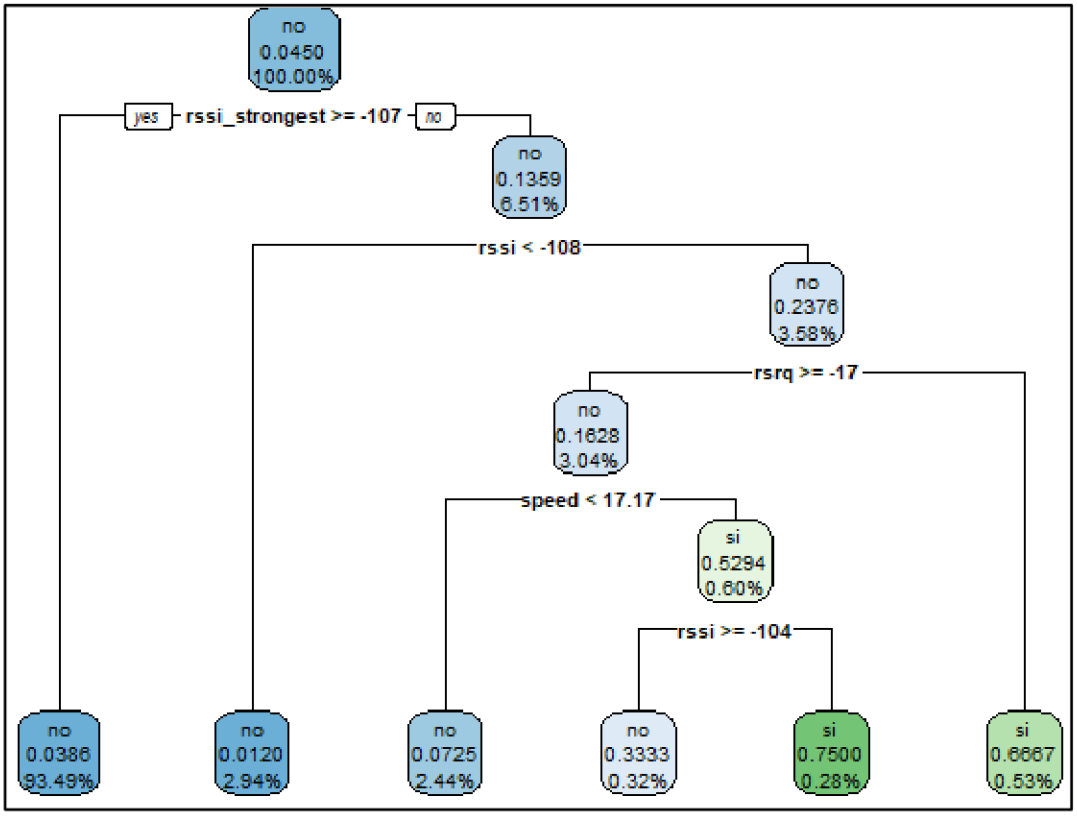


**Figura 3.5.** Gráfico del modelo

En la versión gráfica del modelo se puede apreciar de mejor manera las ramas que se generaron a partir de las mediciones realizadas, se puede observar que el árbol generado tiene 11 nodos en los que se reparten los datos.

Una manera de estudiar estos datos es viendo las primeras ramas, la primera división se realiza en base al parámetro rssi_strongest, por lo que se lo puede tomar como uno de los factores principales al momento de definir los parámetros que causan la interferencia, en la primera división se tiene que hay dos grupos principales, los que tienen un rssi_strongest mayor a –107 (rama izquierda) y los que son menores (rama derecha), este parámetro determina que tan fuerte es la potencia de la celda vecina. Lo anterior tiene sentido, ya que si la potencia de la antena vecina durante el handover es mayor a -107 exista un 4% de probabilidad de interferencia (interrupción); mientras que si el parámetro es menor a dicho valor existe un 14% de probabilidad. En el gráfico anterior se puede ver que la cantidad de observaciones de interferencia es de 94, lo que sigue siendo significativo.

El siguiente parámetro de discriminación para determinar la interferencia es el rssi, en este parámetro el valor que divide el nodo es -108, en este caso si es menor existe un

7% y si es mayor existe un 24% de probabilidad, esto puede resultar contraintuitivo; sin embargo, se debe tener en cuenta que en este nodo solo existe una observación por lo que puede considerarse un dato aislado.

El rsrq divide 86 muestras en dos grupos, cuando se tiene un valor de rsrq mayor a -17 en donde hay un 13% de probabilidad de que exista interferencia; mientras que si es menor, existe un 67% de probabilidad, esto tiene sentido, ya que esta variable representa la calidad de la llamada y se puede observar que es crucial durante el handover, ya que si es muy baja puede causar mayor probabilidad de interferencia.

La velocidad también es una variable que discrimina los datos y se tiene que cuando su valor es mayor a 17 mps que equivale a 61 kph, existe un 53% de probabilidades y 8 observaciones caen en este grupo por lo que sí se lo puede tomar en cuenta como un factor relevante.

El objetivo de este trabajo es estudiar los parámetros que se manifiestan principalmente al momento de darse un handover entre dos celdas, con los datos obtenidos se puede observar un orden de prioridad en los factores que conducen al fallo del handover: rssi del vecino, rssi en servicio, rsrq y velocidad. En el orden anterior están las prioridades del fallo del handover, solamente la velocidad va a depender del UE, el resto son parámetros principalmente controlados por el operador telefónico.

```
##  interf
##    0.01 when rssi_strongest <  -107 & rssi <  -108
##    0.04 when rssi_strongest >= -107
##    0.07 when rssi_strongest <  -107 & rssi >=         -108 & rsrq >= -17 & speed <  17
##    0.33 when rssi_strongest <  -107 & rssi >=         -104 & rsrq >= -17 & speed >= 17
##    0.67 when rssi_strongest <  -107 & rssi >=         -108 & rsrq <  -17
##    0.75 when rssi_strongest <  -107 & rssi is -108 to -104 & rsrq >= -17 & speed >= 17
```

**Figura 3.6.** Reglas de variables

En la figura 3.6 se pueden observar las reglas para obtener distintas probabilidades de interferencia (interrupción de la conexión), se ve que la máxima probabilidad de interferencia es del 75%, lo que implica que no hay una serie de reglas que aseguren la interferencia en el handover, lo que vuelve más difícil el predecir los factores de su ocurrencia.

| | No | Si |
|---|---|---|
| no | 446 | 21 |
| si | 24 | 8 |

**Tabla 3.1** Matriz de confusión

Se puede observar que existen muchos casos de éxito al predecir cuando no se va a dar una interferencia; sin embargo, al momento de predecir la interferencia solo se logran 8 aciertos de las 32 veces que se prueba el modelo.

Este valor corresponde a un 25% de sensibilidad; por lo que el modelo no consigue correlacionar las variables de entrada con la posible causa de una interferencia durante un handover.

## 3.2 CONCLUSIONES

Se encontró una correlación entre las potencias de la celda vecina y servidora que como se muestra en el modelo son variables de gran peso para predecir la interferencia (interrupción) durante el handover, esto está ligado al porcentaje de datos de la figura 3.3; donde se observa que en la mayoría de los casos, el handover se da por una diferencia de las potencias de una celda vecina y la servidora, esto corresponde a un proceso normal de handover, por lo que guarda mucho sentido que sean los factores más relevantes para que se dé una interferencia.

Se obtuvo que el 32% de los handovers que se tienen no se dan cuando la potencia de la celda vecina es mayor que la celda actual ya que se da cuando la potencia de la celda vecina es menor y esto no cumple con la regla de handover que normalmente se aplica lo que implicaría que estos handovers se dan por otros motivos no relacionados directamente a la diferencia de potencia de las celdas.

La velocidad de movimiento es determinante para que el error en el handover sea más frecuente, en especial cuando se tienen velocidades superiores a 62 [km/h]. Se debe mencionar que la velocidad máxima de las muestras es de 84 [km/h], este podría ser el rango en el que la probabilidad de error aumenta significativamente. Hay que tener en cuenta que las observaciones que se ven afectadas por este parámetro son el 0.62% del total, lo que equivale a un conjunto de 25 observaciones de las cuales en el 53% se producen interferencias, el cual es un porcentaje significativo de los datos en este nodo y se puede notar que dentro de todo el modelo es donde se encuentra la mayor probabilidad con un número significativo de observaciones.

El modelo crea una correlación entre las variables medidas y la interferencia durante el proceso de handover; sin embargo, por lo que se puede apreciar en la matriz de confusión esta relación no permite predecir correctamente más del 25% de los casos y esto puede deberse a que como se puede ver en la figura 3.3 hay un porcentaje de casos que corresponden a situaciones en las que la carga de la celda puede influir en el handover; sin embargo, la carga de las celdas consideradas no son parte de este

trabajo, ya que implican un estudio adicional extenso que sale de los objetivos de este modelo.

## 3.3 RECOMENDACIONES

El árbol, debido a su naturaleza, siempre tendrá un conjunto reducido de datos en los nodos que vienen de muchas ramas, ya que cada ramificación implica una división del grupo de datos iniciales. Debido a esto, es importante tener en cuenta que entre más ramificaciones tenga un nodo en los niveles anteriores, este nodo va a tener un número más pequeño de muestras, lo que puede llevar a una correlación poco concluyente en nodos que vengan de muchas ramas sobre todo cuando el grupo de datos que se estudia es de un número reducido. Por esto, se sugiere en un estudio futuro generar árboles con un menor número de variables de entrada para observar el comportamiento y de igual manera realizar un análisis para la selección de dichas variables de entrada.

Para mejorar la precisión del árbol pueden ser de gran utilidad los datos de la carga de las celdas involucradas en el handover. Una manera de recolectar información de la carga podría ser colocando un UE en un punto o zona que se encuentre bajo la cobertura de las celdas involucradas y tomar mediciones periódicas durante varias horas y días para obtener un perfil de carga de las celdas, este dato de la carga de la celda se puede agregar a la base de datos obtenida en el presente trabajo para aumentar las variables involucradas y con esta información aumentar la precisión del árbol.

En un trabajo futuro se podría aumentar la precisión del árbol desestimando los datos que correspondan a handovers que no vengan de una diferencia de potencia entre las celdas servidoras y las vecinas, tomando en cuenta la figura 3.3, esto implica eliminar todas las observaciones cuya diferencia es negativa; sin embargo, la cantidad de muestras para conseguir resultados debe ser mucho mayor que las obtenidas en este trabajo, ya que se descartó cerca del 32% de los datos obtenidos.

Debido a el comportamiento de los arboles de decisión es fácil que con pequeñas variaciones en los valores de los datos la estructura cambie completamente es por esto que no presentan un comportamiento muy robusto, se recomiendan técnicas alternativas de clasificación mas robustas como los son las redes neuronales.

# 4 REFERENCIAS BIBLIOGRAFICAS


[1] C. . A. Serra Jimenez y F. . R. Marante Rizo, «Arquitectura general del sistema LTE,» Revista Telemática, vol. 12, nº 2, pp. 81-90, 2013.

[2] A. Hassan Mohammed and K. Hamid Bilal, "Voice over IP over LTE Network: A Review," International Journal of Engineering, 2015.

[3] R. Foundation, «r-project,» 01 11 2021. [En línea]. Available: https://cran.r-project.org/doc/manuals/r-release/R-intro.html.

[4] 3GPP, «3GPP,» December 2008. [En línea]. Available: https://www.3gpp.org/technologies/keywords-acronyms/98-lte.

[5] S. Patel, V. Shah y M. Kansara, «Comparative Study of 2G, 3G and 4G,» International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, nº 3, pp. 1962-1964, 2018.

[6] M. Anas y F. Calabrese, «Performance Evaluation of Received Signal Strength Based Hard Handover for UTRAN LTE,» Department of Electronic Systems, pp. 1046-1050, 2007.

[7] E. A. Trejo Narváez y C. M. Hernandez Bonilla, «Handover Algorithms in LTE Networks for Massive Means of Transport,» Sistemas & Telemática, vol. 16, nº 46, pp. 21 - 36, 2018.

[8] K. Andersson y S. A. M. Mostafa, «Mobile VoIP User Experience in LTE,» de *IEEE Workshop On User MObility and VEhicular Networks*, Bonn, 2011.

[9] ericsson, «Voice and video calling over LTE,» ericsson, 2012.

[10] S. G. Mompó, Medición y Análisis de las Redes de Comunicaciones Móviles 4G LTE en Cullera, GANDIA, 2019.

[11] Cellmapper, «Soporte,» [En línea]. Available: https://cellmapper.freshdesk.com/support/solutions. [Último acceso: 17 11 2021].

[12] N. T. a. A. S. A. Singh, «A review of supervised machine learning algorithms,» *International Conference on Computing for Sustainable Global Development (INDIACom),* pp. 1310-1315, 2016.

[13] A. J. Myles, R. N. Feudale, Y. Liu y N. A. Woody, «An introduction to decision tree modeling,» *JOURNAL OF CHEMOMETRICS,* p. 275–285, 2004.

[14] C. D. Altay A., «Fuzzy Decision Trees,» de *Fuzzy Statistical Decision-Making*, Springer, Cham, 2016, pp. 221-261.

[15] S. B. Z. I. &. P. Kotsiantis, «Supervised machine learning: A review of classification techniques,» *Emerging artificial intelligence applications in computer engineering,* pp. 3-24, 2007.

[16] RDocumentation, «RDocumentation,» 8 Marzo 2020. [En línea].

[17] S. Milborrow, Plotting rpart trees with the rpart.plot package, 2021.

[18] A. Ghasemi, «Predictive modeling of LTE user throughput via crowd-sourced mobile spectrum data,» *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN),* pp. 1-5, 2018.

[19] H. Wickham, «Rdocumentation,» 10 Noviembre 2018. [En línea]. Available: https://www.rdocumentation.org/packages/dplyr/versions/0.7.8.

# 5 ANEXOS

Anexo I. Código en R utilizado para la creación del modelo.

Anexo II . Conjunto de mediciones realizadas en la zona de estudio.

# Anexo I

```
#CODIGO PRINCIPAL
#CREACION DE MODELO PREDICTIVO CON ARBOL DE DESICIONES
#Autor: Cristian Rocha
#Paquetes: dplyr,rpart

# Librerias ---------------------------------------------------------
library(dplyr)
library(rpart)
library(rpart.plot)
library(rattle)
library(RColorBrewer)
# Cargar Datos ------------------------------------------------------
setwd("/Users/crist/Documents/Universidad/Tesis/datos_medidos/Net_monitor/Rutas")
datos <- lapply(Sys.glob("Ruta?/R?S*S.csv"), read.csv)
pos <- lapply(Sys.glob("Ruta?/R?S*P.csv"), read.csv)
datos <- do.call("rbind",datos)
pos<- do.call("rbind",pos)
datos$interf <- "no"
datos$handover <- "no"
# Tratamiento de datos ----------------------------------------------
a <- nrow(pos)
b <- nrow(datos)
#Agregar interferencia
for (i in 1:a) {
  for (j in 1:b) {
    if((j>2)&(pos$latitude[i]) == (datos$lat[j])){
      datos$interf[j-2] <- "si"
      datos$interf[j-1] <- "si"
      datos$interf[j] <- "si"
    }
  }
}
for (i in 1:b) {
  if (i>4) {
    x <- datos$long_cid[i-2]
    y <- datos$long_cid[i-3]
```

```
    if((x)!=(y)){
      datos$handover[i-5] <- "si"
      datos$handover[i-4] <- "si"
      datos$handover[i-3] <- "si"
      datos$handover[i-2] <- "si"
      datos$handover[i-1] <- "si"
      datos$handover[i] <- "si"
    }
  }
}
prop.table(table(datos$interf))
prop.table(table(datos$handover))
total <- filter(datos, handover == "si")
total <- select(total,c(lte_neighbors, rssi_strongest, rssi, rsrq, rssnr, speed, interf))
total[total == 2147483647] <- NA
total <- filter(total, rssi >= -110)
na.omit(total)

# Muestreo ------------------------------------------------------------
create_train_test <- function(data, size =0.85, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
train <- create_train_test(total, 0.85, train = TRUE)
test <- create_train_test(total, 0.85, train = FALSE)
#total <- total[sample(1:nrow(total)), ]
cantidad <- count(total,interf)
cantidad
# Pruebas de valores ------------------------------------------------------
valores<-vector()
tam<-vector()
```

```
ideal<-vector()

k=1
for (r in 1:10) {

}
for (j in 1:10) {
  for (i in 1:10) {
    fit <- rpart(interf ~ ., data = train, method = "class", minsplit = i, minbucket = j,cp=0)
    prueba <-predict(fit, test, type = 'class')
    matriz <- table(test$interf, prueba)
    sensib <- round((matriz[4]/(matriz[2]+matriz[4])),2)
    valores[k]<-sensib
    tam[k]<-length(fit$frame$var)
    k=k+1
  }
}
valores
tam
max(valores)
# Modelo final ----------------------------------------------------------
fit <- rpart(interf ~ ., data = train, method = "class", minsplit = 1, minbucket = 8)
rpart.plot(fit, digits = 4)
prueba <-predict(fit, test, type = 'class')
matriz <- table(test$interf, prueba)
sensib <- (matriz[4]/(matriz[2]+matriz[4]))
matriz
sensib
length(fit$frame$var)
fit
rpart.rules(fit)

# Analisis ----------------------------------------------------------
RSSI<- total$rssi
RSSI_NEIGHBOUR<- total$rssi_strongest
DIFF <- RSSI - RSSI_NEIGHBOUR
h1<- hist(RSSI_NEIGHBOUR)
```

```
h2<- hist(RSSI)
h3<- hist(DIFF, breaks = 1)
text(h3$mids,h3$counts,labels=h3$counts)
```