DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE ANCHO DE BANDA PARA REDES WLAN BASADO EN APRENDIZAJE DE MÁQUINA

ANDREA JOHANA CHAVES VILLLOTA
OSCAR JAVIER JOSSA BASTIDAS

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2018

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE ANCHO DE BANDA PARA REDES WLAN BASADO EN APRENDIZAJE DE MÁQUINA

ANDREA JOHANA CHAVES VILLLOTA
OSCAR JAVIER JOSSA BASTIDAS

Trabajo de grado presentado como requisito para optar al título de Ingeniero Electrónico

Director
M.Sc. Mario Fernando Jojoa Acosta
Ingeniero Electrónico

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2018

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.”

“Las ideas y conclusiones aportadas en este Trabajo de Grado son Responsabilidad de los autores”.

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN:

_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

San Juan de Pasto, 26 de noviembre de 2018

DEDICATORIA

"A mi madre por ser el pilar que equilibra mi vida, por el gran amor, la devoción, el apoyo ilimitado e incondicional que siempre me ha brindado, por contar con la suficiente fortaleza para salir adelante sin importar las adversidades. A mi padre que eternamente me cuida y me guía desde el cielo.
A mi hermano por su compañía absoluta, por su protección, por sus muestras de cariño y afecto continuamente presentes.
A toda mi familia por sus palabras de aliento y sus gratos deseos, por su compañía y apoyo en todos los momentos."

Andrea Johana Chaves

"A Dios, por guiarme en el camino correcto, cuidarme y bendecirme cada día. A mí mamá Consuelo Bastidas quien es la guía de mi vida y por mucho tiempo tuvo que hacer el papel de padre y madre, a Víctor Guancha quien le otorgo su compañía y me brindó su apoyo en todo momento, de la misma forma a mis dos hermanas Juliana y Catalina. A mis tíos Dayra Bastidas Y Hugo Rosero que me acogieron en su hogar y me brindaron su cariño y apoyo, a mis Abuelos que también fueron como otros padres para mí y a toda mi familia en general por estar siempre pendiente y apoyándome en cada paso que he dado.

A Andrea Johana Chaves por todos los buenos momentos compartidos, sus consejos, enseñanzas y cariño, a su familia quien me dio todo el afecto y apoyo a lo largo de toda la carrera. Así mismo un agradecimiento especial a nuestro director Mario Fernando Jojoa, quien nos dio la confianza y seguridad para sacar este proyecto adelante"

Oscar Javier Jossa Bastidas

AGRADECIMIENTOS

En primera instancia queremos agradecer a la Universidad de Nariño y especialmente al Departamento de Electrónica, por permitirnos hacer parte de su grupo científico e investigativo, así mismo a los diferentes docentes que brindaron su conocimiento y apoyo en las diversas áreas de la carrera.

Agradecemos especialmente a nuestro asesor el M.Sc. Mario Fernando Jojoa Acosta, por brindarnos su confianza, amistad, asesoría y acompañamiento durante la ejecución del trabajo; su motivación y ejemplo nos permite crecer en la formación como investigadores y personas.

A todos los funcionarios y personal del Aula de Informática, particularmente al M.Sc. Ignacio Eraso y al Ing. Pablo Vaca, quienes brindaron todo su apoyo y conocimiento de la estructura de la red de datos de la Universidad de Nariño.

A cada uno de nuestros compañeros y amigos, por cada uno de los consejos recibidos, por los momentos compartidos y su apoyo incondicional en todo momento.

RESUMEN

En la actualidad, las redes inalámbricas de área local (Wireless Local Area Network, WLAN) son estructuras de comunicación muy comunes y ampliamente utilizadas, debido a que permiten mayor flexibilidad, adaptabilidad y comodidad para los usuarios; sin embargo, tienen una gran dificultad en brindar una conexión rápida y segura bajo alta demanda de tráfico. Por lo tanto, otorgar un servicio de calidad en las WLAN es una necesidad latente, de ahí que este problema ha sido objeto de múltiples estudios que involucran diversidad de técnicas y métodos para garantizar conexiones estables y de buena calidad. En la presente investigación se emplearon técnicas de aprendizaje de máquina que contribuyeron a la administración de una WLAN.

En este proyecto se propueso el uso de dos métodos de predicción basados en aprendizaje de máquina: k-medias (kmeans) y máquina de soporte vectorial (Support Vector Machine, SVM) para estudiar el comportamiento de una WLAN de la universidad de Nariño, en conjunto con un método o algoritmo de gestión de ancho de banda para los clientes de dicha red.

El sistema desarrollado fue capaz de administrar el ancho de banda disponible y asignarlo dinámicamente, de acuerdo a la navegación y tiempo de conexión de los clientes. Para lograr este fin, se conformó un repositorio del cual se extrajeron características del tráfico de datos, posteriormente se implementó una SVM, que permitió clasificar el comportamiento de navegación de los usuarios, luego estos resultados fueron sometidos a una etapa de validación cruzada de k-iteraciones (K-fold cross validation). Finalmente, con la información obtenida de los sistemas de aprendizaje de máquina, se establecieron perfiles de navegación y se procede a asignar ancho de banda con la ayuda de la herramienta que incorporada en mikrotik.

ABSTRACT

Currently wireless local area networks (WLAN) are very common communication structures and widely used, because they allow greater flexibility, adaptability and comfort for users; however, they have great difficulty in providing a fast and secure connection under high traffic demand. Therefore, grant a quality service in WLAN was a latent need, hence this problem has been the subject of multiple studies that involve a variety of techniques and methods to ensure stable and good quality connections. In this research, machine learning techniques are used that contribute to the administration of a WLAN network.

In this Project the used of two prediction methods based on machine learning: K-means and Support Vector Machine (SVM) are proposed to study the WLAN behaivor of University of Nariño together with a bandwidth management method or algorithm for the clients of the mentioned network.

The developed system was capable of managing the available bandwidth and assigning it dynamically, according to navigation and connection time of clients.. To achieve this end, first a data repository was shaped from which data traffic characteristics will be extracted, later an SVM was implemented, which allows to classify the user's browsing behavior, then these results were subjected to K-fold cross-validation. Finally, with the information obtained from the machine learning systems, navigation profiles were established and bandwidth was allocated with the help of the queue tool that was incorporated in the mikrotik.

CONTENIDO

Pag

LISTA DE TABLAS

Pag

LISTA DE FIGURAS

Pag

LISTA DE ANEXOS

Pag

GLOSARIO

AGRUPAMIENTO: mejor conocido como *clustering*, es la técnica que permite agrupar objetos que tienen un alto grado de similitud entre ellos en grupos o clusters y que a su vez son muy diferentes a los de otros grupos.

ANCHO DE BANDA: hace referencia a la medida de la velocidad de datos y recurso de comunicación disponible o consumida.

APROVISIONAMIENTO: también conocido como *overprovisioning*, cuando una red cuenta con todos los recursos para brindar una buena calidad del servicio a sus usuarios.

BYTES-IN: en esta investigación hace referencia a los bytes recibidos por el equipo de red provenientes del cliente.

BYTES-OUT: en esta investigación hace referencia a los bytes que el equipo de red entrega al cliente.

CAJA Y BIGOTES: diagrama que muestra visualmente grupos de datos numéricos a través de sus cuartiles, utilizado principalmente para indicar la variabilidad de las muestras.

CLIENTE: en esta investigación hace referencia a un dispositivo electrónico que se conecta a una red y realiza las peticiones de servicio de internet.

Cloud Core ccr1016-12g: es un enrutador en calidad de administrador de la marca MIKROTIK, cuenta con un procesador potente que le permite realizar varias funciones con una velocidad muy superior a otros equipos.

CONMUTADOR: también conocido como *switch,* es un dispositivo de red utilizado para la interconexión de redes informáticas y cuya función principal es interconectar dos o más segmentos de red.

DIMENSIÓN: En este trabajo hace referencia a la cantidad de atributos, características o mediciones que tiene cada observación.

DIRECCIÓN IP: es un número de 32 bits que identifica lógica y jerárquicamente a una interfaz de red de un host o equipo de red y que funciona bajo el protocolo TCP/IP.

ENRUTADOR: es un equipo a nivel de capa de red en el modelo OSI, cuya función principal es encaminar paquetes de una red a otra.

FLUJO: se denomina a un conjunto de paquetes que van de un origen a un destino.

FortiGate 800C: equipo de red creado por la empresa FORTINET, cuenta con funciones de alto rendimiento de red, seguridad y contenido.

FUNCIÓN DE BASE RADIAL: conocido por sus acrónimos en ingles como *Radial Basis Function (RBF)*, es una función kernel usada en varios algoritmos de aprendizaje de máquina, especialmente en la SVM.

GRUPO: en este trabajo hace referencia a un grupo o cluster de perfiles de usuarios.

HOST: es un equipo que envía y recibe tráfico de datos de los usuarios de una red y que se identifica mediante una dirección IP.

IEEE 802.11: es el estándar que especifica las normas de funcionamiento de una red WLAN.

JITTER: variación en el retardo o los tiempos de llegada de los paquetes.

KERNEL: en este trabajo hace referencia a funciones matemáticas que permiten en general la transformación de un espacio a otro con mejor separabilidad.

K-ITERACIONES: técnica empleada para evaluar el desempeño del sistema de clasificación que consiste en dividir el conjunto de datos en *k* grupos y realizar el entrenamiento y prueba en *k* iteraciones.

K-MEANS: técnica de agrupamiento cuyo objetivo es dividir un conjunto de *N* muestras en *K* grupos, en el que cada muestra pertenece al grupo donde su media es más cercana.

LATENCIA: hace referencia al tiempo que se tarda en transmitir un paquete desde un equipo local hacia un equipo remoto.
MAC: es un identificador único asignado por el fabricante de una pieza de hardware de red. Una dirección MAC se compone por seis grupos de dos caracteres, que utiliza numeración hexadecimal.

MÁQUINA DE SOPORTE VECTORIAL: también conocido como *support vector machine (SVM)*, es un método de aprendizaje automático supervisado, se fundamenta en el uso del kernel y su interpretación geométrica y la construcción de un hiperplano de separación óptimo.

MASCARA DE RED: es una combinación de bits que indica qué parte de la dirección IP es el número de red y que parte corresponde al host.

MySQL: sistema de gestión de base de datos.

OSI: es un modelo de referencia formado por siete capas que define las diferentes etapas por las que los paquetes de información deben pasar, para viajar de un dispositivo a otro sobre una red de comunicaciones.

PAQUETE: en informática hace referencia a los bloques en los que es dividida la información para ser enviada en la capa de red del modelo de referencia OSI.

PORTAL CAUTIVO: también conocido como *hotspot*, aplicación empleada generalmente en redes inalámbricas abiertas con el fin de controlar y autenticar el acceso de los usuarios a la red.

PROTOCOLO DE CONFIGURACIÓN DINÁMICA DE HOST: también conocido como *Dynamic Host Configuration Protocol (DHCP)* es un protocolo de red que asigna dinámicamente una dirección IP junto con otros parámetros de red a cada dispositivo para que puedan comunicarse.

PUNTO DE ACCESO: también conocido como *Acces Point*, es un dispositivo de red que interconecta equipos de comunicación para formar una red inalámbrica y permite transportar la información mediante ondas electromagnéticas.

PYTHON: es un lenguaje de programación que permite trabajar de forma rápida e integrar sistemas de manera efectiva. En este trabajo se utiliza para el desarrollo del procesamiento de datos.

RADIUS: es un protocolo de autenticación y autorización para acceso a la red.
RED DE AREA LOCAL: conocida como *local area network (LAN),* son redes en un área local de propiedad privada que permiten el intercambio de información entre los equipos conectados a esta red.

RED DE AREA LOCAL VIRTUAL: también conocida como *Virtual Local Area Network (VLAN)* divide los grupos de usuarios de una red física en segmentos de redes lógicas o virtuales.

RouterOS: Sistema Operativo de los equipos fabricados por la empresa Mikrotik.

SEGMENTO DE RED: división de una red informática en subredes de menor tamaño.

SERVICIO ORIENTADO A CONEXIÓN: Hace referencia a una forma de comunicación de redes donde se debe establecer una conexión antes de que los datos sean transferidos

SISTEMA DE NOMBRES DE DOMINIO: conocido como *Domain Name System (DNS),* su función mas importante es traducir nombres inteligibles para las personas en identificadores binarios entendibles para los equipos de red.

SCIKIT-LEARN: librería que permite desarrollar aprendizaje de maquina en Python, brinda herramientas simples y eficientes para la minería de datos y el análisis de datos.

TCP/IP: hace referencia al protocolo de control de transmisión (TCP) y protocolo de Internet (IP) que hacen posible la transferencia de datos entre redes de ordenadores.

TÉCNICA DE ELBOW: también conocido como metodo del codo, utiliza los valores de las distancias al cuadrado de cada objeto del clúster a su centroide, para diferentes números de *K* clusters de 1 a *N*, dando una representación gráfica que

permite visualizar un punto de inflexión que indica el número de clusters *K* adecuado para la técnica de K-means.

TRADUCCIÓN DE DIRECCIONES DE RED: conocido por sus siglas en ingles *Network Address Translation (NAT)*, cuya función es intercambiar paquetes entre dos redes que asignan mutuamente direcciones IP incompatibles.

UP-TIME: en este trabajo indica la cantidad de tiempo que el usuario ha permanecido conectado.

WLAN: es un tipo de red local donde sus equipos no necesitan estar vinculados a través de un medio físico para conectarse a internet o compartir recursos.

INTRODUCCIÓN

El número de usuarios de internet, según Internet Live Stats, reflejó que ésta se ha convertido en una necesidad y una herramienta indispensable de trabajo. Alrededor del 40% de la población mundial ha tenido conexión a Internet, indicando que el número de usuarios se ha multiplicado diez veces entre los años 1999 y 2013. Según el escalafón de competitividad de los departamentos en Colombia para el año 2015, la velocidad de descarga desde internet residencial, la penetración del internet y la cobertura de servicios públicos domiciliarios, fueron los indicadores con mayor peso en el factor infraestructura del país. De acuerdo con esta información, se apreció la importancia que ha tomado la Internet en el mundo y en el país. En la actualidad, es común encontrar conexiones a internet mediante las redes de área local LAN (*Local Area Network*) y las WLAN, de ahí que es importante brindar un servicio de calidad para los clientes de las redes.

Por otra parte, se dice que el realizar actividades tales como escribir programas de computadores, realizar ejercicios matemáticos, razonar según el sentido común, entender lenguajes e incluso conducir un automóvil, requieren inteligencia. Con el auge de la tecnología moderna, en las últimas décadas se han incorporado sistemas informáticos que pueden realizar algunas de estas tareas con niveles de confiabilidad alta. Se han encontrado sistemas informáticos que, por ejemplo, diagnostican enfermedades, planifican la síntesis de compuestos químicos orgánicos complejos, resuelven ecuaciones diferenciales complejas, analizan circuitos electrónicos o comprenden el habla humana en diversos idiomas; dichos sistemas han sido desarrollados con alguna técnica de inteligencia artificial.

Un tipo de algoritmo de aprendizaje de máquina muy conocido, y que ha tenido gran acogida entre los investigadores de distintas áreas del conocimiento, corresponde a la máquina de soporte vectorial, cuyo uso ha reportado buenos resultados frente al uso de otras técnicas del aprendizaje de máquinas y reconocimiento de patrones. Una SVM mapea los datos que se pretende clasificar a un espacio de características de una dimensión mayor; es decir, si los puntos de entrada pertenecen a $\mathbb{R}^m$, entonces son mapeados por la SVM al espacio $\mathbb{R}^{m+1}$, y encuentra un hiperplano óptimo que los separa, maximizando su margen de separación.

Este trabajo de grado en modalidad de investigación se desarrolló bajo la línea de comunicaciones del departamento de ingeniería electrónica, en donde se hizo uso de la SVM con la capacidad de aprender a través de la dinámica de las conexiones de los clientes de la red. El sistema aprendió de los datos de navegación de los dispositivos conectados a la red, como tiempos de conexión o volúmenes de descarga, con esta información se llevó a cabo una clasificación de los usuarios en perfiles de navegación. El proceso de clasificación se realizó con el fin de asignar el ancho de banda disponible de la red y bajo protocolos de navegación preestablecidos por el administrador, dando prioridad según sea el caso a un cierto grupo de clientes.

Definición del problema

¿Pudo mejorarse la navegación en la red de datos inalámbrica de la Universidad de Nariño, a partir de un sistema de gestión del ancho de banda basado en la clasificación de los perfiles de navegación de los usuarios?

Objetivo General

Proponer un sistema de gestión de ancho de banda para una red WLAN de la Universidad de Nariño basado en un clasificador de tipo SVM.

Objetivos Específicos

- Crear una base de datos (repositorio) del tiempo de conexión y el volumen de descarga de los clientes de una red inalámbrica de prueba.

- Proponer un modelo de gestión de ancho de banda fundamentado en un clasificador de perfiles de navegación tipo SVM y la extracción de características.

- Evaluar el desempeño del clasificador a través del método de validación cruzada de K iteraciones.

Alcance

Los alcances metodológicos, teóricos y prácticos alcanzados en esta investigación se listan a continuación:

- **Sistema de almacenamiento de bases de datos de una red de prueba (WLAN):** Se definió un repositorio para los datos de navegación de los clientes de la red de prueba. Para este fin, se llevó a cabo el desarrollo del script, mediante un lenguaje de programación de alto nivel.

- **Metodología de procesamiento de datos:** Se desarrolló un proceso matemático para la identificación de las características más relevantes en el proceso de separación de clases.

- **Metodología de clasificación de perfiles de navegación:** Se diseñó una estrategia de clasificación basada en SVM para clasificar perfiles de navegación. Para este propósito se utilizó las características extraídas de los datos del repositorio.

- **Metodología de validación del clasificador:** Se evaluó los resultados del clasificador a través de un análisis estadístico, haciendo uso de una técnica que permitió determinar la precisión del modelo propuesto. Se utilizó el método de validación cruzada de K iteraciones.

- **Sistema de gestión de ancho de banda:** Se desarrolló una técnica de gestión, que asigne de manera dinámica el ancho de banda disponible y de acuerdo a los perfiles de navegación encontrados.

Marco Teórico

Hardware de red[1]

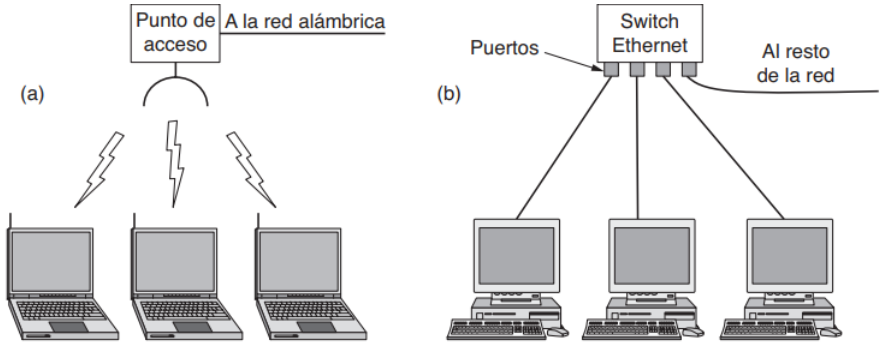
- **Redes de área local:** como su nombre lo indica son redes en un área local de propiedad privada, generalmente se encuentran dentro de un edificio, fábrica, casa u oficina. Este tipo de redes se usan en gran medida para conectar ordenadores, electrodomésticos y de esta manera compartir recursos (impresoras, escáner) e intercambiar información. En esta investigación se trabajó sobre la red LAN de la Universidad de Nariño sede panamericana, cuyos equipos que la conforman fueron utilizados como medios para recolectar y transportar la información de la base de datos.

[1] WETHERALL, Tanenbawm. Redes de Computadoras. México: Pearson Educación, 2012.

Las redes LAN cableadas utilizan distintas tecnologías de transmisión, la mayoría usan cobre aunque también se utiliza la fibra óptica. Por lo general estas redes operan en velocidades de 100 Mbps a 1 Gbps, tienen retardo en microsegundos o nanosegundos y cometen muy pocos errores. En la red LAN de la Universidad de Nariño se utiliza tecnologías de cobre y fibra óptica.

El Estándar IEEE 802.3 comúnmente conocido como Ethernet, es en la actualidad el más utilizado para redes LAN cableadas. En la Figura 1(b) se muestra un ejemplo de topología de ethernet conmutada, donde cada dispositivo se conecta a un equipo de red denominado conmutador (switch) en un enlace punto a punto a través del protocolo Ethernet. El trabajo del conmutador es conectar los dispositivos en red y transmitir los paquetes entre los computadores conectados a él.

Figura 1. Redes inalámbrica y cableada. (a) 802.11. (b) Ethernet conmutada.



Fuente: Wetherall, Tanenbawm. Redes de computadoras.

- **Redes de área local inalámbricas (WLAN):** son cada día más populares, encontrándose en cualquier lugar (universidad, centros comerciales, hogar) con mayor frecuencia. Este tipo de redes se utilizan para conectar computadoras, dispositivos (PDA, personal digital assistant) y teléfonos inteligentes (smartphones) a internet. Este tipo de redes utilizan las ondas de radio para transportar la información de un punto a otro sin necesidad de un medio físico. La red WLAN de la Universidad de Nariño de la sede panamericana está presente en la mayoría de bloques y es difundida mediante puntos de acceso, donde diversos usuarios (estudiantes, docentes, administrativos, invitados) se conectan a la red, esto permitió que la recolección de la base de datos contenga información múltiple.

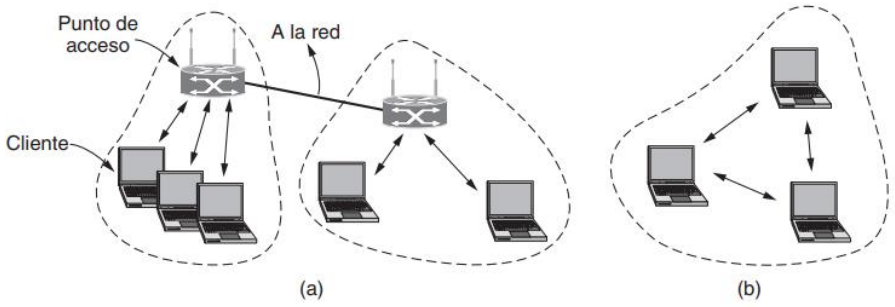
En la Figura 1(a) se muestra un ejemplo una red inalámbrica, cada dispositivo se conecta a un equipo de red denominado punto de acceso (Access Point, AP) para realizar la transmisión de información sin necesidad de un medio físico.

Estándar IEEE 802.11

Arquitectura básica 802.11: el principal estándar para redes WLAN es el IEEE 802.11 que define y especifica las normas y el funcionamiento de la capa física y de la capa de enlace de datos del modelo OSI. Este tipo de redes se puede usar de diferentes formas, la más usual y conocida es conectar un host a otra red, como la intranet de una empresa o internet, esto se muestra en la Figura 2(a), En modo infraestructura, los clientes se conectan a través de un AP que está conectado a la red, de manera que los hosts envían y reciben los paquetes a través del AP, este es el modelo que utilizó la red WLAN de la Universidad de Nariño.

Por otra parte, el modo ad hoc apreciado en la Figura 2(b) es una conexión de computadoras conectadas a una red, así estas se pueden enviar información de unos a otros directamente, en este modo no hay un punto de acceso y debido a esto no es muy popular ya que no existe salida a internet.

Figura 2. Arquitectura 802.11. (a) Modo infraestructura. (b) Modo ad hoc



Fuente: Wetherall, Tanenbawm. Redes de computadoras.

Calidad del servicio

Existen aplicaciones y/o clientes que exigen un desempeño de calidad de la red que de garantía al buen funcionamiento de dichas aplicaciones. Una alternativa rápida y a la vez sencilla para proporcionar una buena calidad del servicio es adquirir y/o construir una red cuyas capacidades soporten cualquier tráfico que esta maneje, a esto se le llama exceso de aprovisionamiento (overprovisioning), obteniendo como resultado una latencia supremamente baja y altas tasas de transferencia. Sin

embargo, adquirir y mantener una red de dichas capacidades supone un alto costo económico.

Con el propósito de dar soluciones alternativas se han desarrollado mecanismos de calidad del servicio, lo cual permite que una red con menos capacidad cumpla con igual eficiencia los requerimientos de aplicaciones y clientes a costos considerablemente más bajos. El propósito de la investigación fue aportar un camino diferente utilizando técnicas de aprendizaje de máquina que permitan agrupar el comportamiento de los usuarios y de esta manera distribuirles los recursos de una red.

Se tienen que tener en cuenta cuatro aspectos para garantizar la calidad del servicio:

- Lo que las aplicaciones necesitan de la red.
- Hacer una regulación de entrada y salida del tráfico de la red.
- Reservación de recursos en los enrutadores.
- Aceptación o no de más tráfico en forma segura a la red.

En una red orientada a conexión, un flujo podría constituir todos los paquetes de una conexión y en una red sin conexión, un flujo seria todos los paquetes enviados de un proceso a otro. Dicho esto, las necesidades de un flujo se pueden caracterizar teniendo en cuenta 4 parámetros principales: ancho de banda, retardo, variación del retardo (jitter) y pérdida, en conjunto estos forman lo que se denomina como calidad del servicio (Quality of Service, QoS). Dentro de esta investigación se trabajó sobre el ancho de banda para otorgar en cierta medida QoS.

En la Tabla 1 se listan ciertas aplicaciones con sus respectivos requerimientos de red. Alto indica una mayor prioridad y bajo la menor prioridad.

Tabla 1 Niveles de requerimientos de calidad de servicio en algunas aplicaciones

Aplicación	Ancho de banda	Retardo	Variación del retardo	Pérdida
Correo electrónico	Bajo	Bajo	Bajo	Media
compartir archivos	Alto	Bajo	Bajo	Media
Acceso a web	Medio	Medio	Bajo	Media
inicio de sesión remoto	Bajo	Medio	Medio	Media
Audio bajo demanda	Bajo	Medio	Medio	Baja
Video bajo demanda	Alto	Bajo	Alto	Baja

Telefonía	Bajo	Alto	Alto	Baja
Videoconferencias	Alto	Alto	Alto	Baja

Fuente: Wetherall, Tanenbawm. Redes de computadoras.

Es posible notar que los servicios orientados a conexión requieren un nivel de exigencia superior a la red que los servicios sin conexión. En cuanto a las necesidades de ancho de banda, el correo electrónico, audio e inicio de sesión remoto no son muy exigentes, caso contrario pasa con los servicios de compartición de archivos y video.

Por otra parte, los requerimientos de retardo no son muy sensibles, en las aplicaciones en las cuales existen transferencia de archivos como lo son video o e-mail, en cambio las aplicaciones interactivas como inicio de sesión remoto y navegación web si lo son.

En cuanto a las aplicaciones en tiempo real, como telefonía y videoconferencias, son muy estrictas en cuanto al retardo. El jitter, no es muy sensible en el correo electrónico, compartir archivos o acceso a la web, pero para los casos de inicio de sesión y en especial para una videoconferencia si lo es.

Cabe resaltar que los servicios y aplicaciones en la red no necesitan tener un retardo ideal de 0 ms para reproducir audio y video o una pérdida de cero paquetes para realizar una transferencia de archivos. Las pérdidas se pueden remediar con las retransmisiones y las variaciones del retardo si se colocan paquetes en el buffer del receptor o más aún la creación de aplicaciones que apliquen calidad de servicio a la red, que es en cierto modo lo que se desarrolló en este trabajo de grado; sin embargo, si la red cuenta con recursos muy limitados de ancho de banda o existe demasiado retardo no hay nada que se puede hacer.

Modelos de predicción estadísticos[2,3]

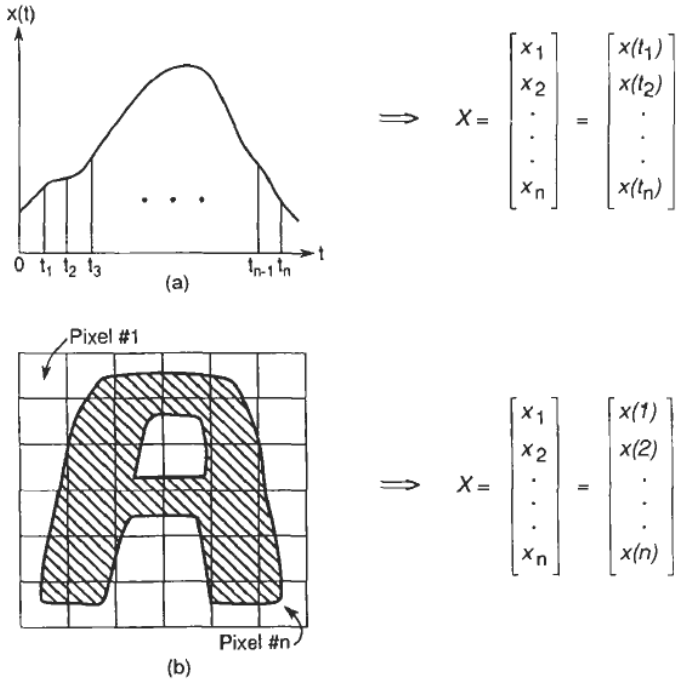
Unas de las aplicaciones más utilizadas en el reconocimiento de patrones suelen ser la clasificación de forma de onda o clasificación de figuras geométricas. Una forma de extraer toda la información contenida en la muestra es medir los *n* valores muestreados en el tiempo para una forma de onda, $x(t_1), \ldots, x(t_n)$, y los *n* niveles

[2] FUKUNAGA, Keinosuke. Statistical Pattern Recognition. World Scientific, 1999.
[3] GUISANDE, Gonzáles. *et. al*. Tratamiento de Datos con R, Statistica y SPSS. Diaz de Santos, 2013.

grises de píxeles de una figura, *x(1), . . . , x(n)*, formando un vector *X*, como se indica en la Figura 3.

Cada vez que se realiza la observación, las muestras de la forma de onda son distintas, por lo tanto, $x(t_i)$ es una variable aleatoria y el vector cuyas componentes son variables aleatorias se denomina vector aleatorio, expresado como ***X***.

Figura 3**.** Vector de características de dos patrones. (a) Forma de Onda; (b) Carácter.



Fuente: Fukunaga, Keinosuke. Statistical Pattern Recognition.

Funciones de densidad y distribución

Sea ***X*** un vector aleatorio con *n* variables aleatorias definido como:

$$\boldsymbol{X} = [x_1 \;\; x_2 \ldots x_n \,]^T \tag{1}$$

Donde *T* indica la transpuesta del vector.

- **Función de distribución:** un vector aleatorio se puede caracterizar por una función de distribución de probabilidad definida en (2).

$$P(x_1, \ldots, x_n) = P\,r\{\boldsymbol{x_1} \leq x_1, \ldots, \boldsymbol{x_n} \leq x_n\} \tag{2}$$

Donde *Pr{A}*, es la probabilidad de un evento A. Por conveniencia (2) se escribe como:

$$P(X) = \Pr\{\boldsymbol{X} \leq X\} \tag{3}$$

- **Función de densidad:** un vector aleatorio también se puede caracterizar por la función de densidad definida en (4)

$$p(\boldsymbol{X}) = \lim_{\substack{\Delta x_1 \to 0 \\ \vdots \\ \Delta x_n \to 0}} \frac{\Pr\{x_1 < \boldsymbol{x_1} \leq x_1 + \Delta x_1, \ldots, x_n < \boldsymbol{x_n} \leq x_n + \Delta x_n\}}{\Delta x_1 \ldots \Delta x_n} = \frac{\partial^n P(X)}{\partial x_1 \ldots \partial x_n} \tag{4}$$

En el reconocimiento de patrones, los vectores aleatorios trabajados son extraídos de diferentes clases (o categorías), cada una es caracterizada por su propia función de densidad. Esta función de densidad es denominada densidad de la clase *i* y se expresa en (5).

$$p(X \mid \omega_i) = p_i(X) \quad (i = 1, \ldots, L) \tag{5}$$

Donde ω_i indica el rotulo de la clase *i* y *L* es el número de clases.

Parámetros de las distribuciones

Como se mostró anteriormente un vector aleatorio ***X*** es caracterizado por sus funciones de densidad o distribución. Sin embargo, para la práctica dichas funciones no pueden determinarse fácilmente debido a su complejidad matemática, por esta razón se adopta una caracterización más computable.

- **Valor esperado:** también conocido como media de un vector aleatorio ***X***, se define por:

$$M = E\{\boldsymbol{X}\} = \int \boldsymbol{X}\, p(\boldsymbol{X})\, dX \tag{6}$$

La integración es tomada en todo el espacio ***X***, a menos de que se especifique lo contrario. El í-ésimo componente de *M*, m_i puede ser calculado por:

$$m_i = \int x_i p(\boldsymbol{X})\, dX = \int_{-\infty}^{\infty} x_i\, p(x_i) dx_i \tag{7}$$

Donde $p(x_i)$ es la densidad marginal del i-ésimo componente de $\boldsymbol{X}$. De ahí que cada componente m_i de M es calculado realmente por el valor esperado de una variable individual con la densidad marginal unidimensional.

El valor esperado del vector aleatorio $\boldsymbol{X}$ para una clase ω_i, se define por:

$$M_i = E\{\, \boldsymbol{X} \mid \omega_i\} = \int \boldsymbol{X}\, p_i(\boldsymbol{X})\, dX \tag{8}$$

- **Matriz de covarianza:** permite evaluar conjuntos de parámetros que indican la dispersión de la distribución. La matriz de covarianza Σ de $\boldsymbol{X}$ se estima con (9).

$$\begin{aligned}\Sigma = E\{(\boldsymbol{X} - M)(\boldsymbol{X} - M)^T\} &= E\left\{ \begin{bmatrix} x_1 - m_1 \\ \vdots \\ x_n - m_n \end{bmatrix} \quad [x_1 - m_1 \ \dots \ x_n - m_n] \right\} \\ &= \begin{bmatrix} E\{(x_1 - m_1)(x_1 - m_1)\} & \dots & E\{(x_1 - m_1)(x_n - m_n)\} \\ \vdots & \ddots & \vdots \\ E\{(x_n - m_n)(x_1 - m_1)\} & \dots & E\{(x_n - m_n)(x_n - m_n)\} \end{bmatrix} \\ &= \begin{bmatrix} c_{11} \dots c_{1n} \\ \vdots \\ c_{n1} \dots c_{nn} \end{bmatrix}\end{aligned} \tag{9}$$

Los componentes c_{ij} de la matriz son:

$$c_{ij} = E\big\{(x_i - m_i)\big(x_j - m_j\big)\big\} \quad (i, j = 1, \dots, n) \tag{10}$$

Por lo tanto, las componentes de la diagonal de la matriz son las varianzas de las variables aleatorias individuales, y las componentes fuera de la diagonal son las covarianzas de dos variables aleatorias x_i y x_j. Por conveniencia los índices c_{ij} son expresados como:

$$c_{ii} = \sigma_i^2 \quad c_{ij} = \rho_{ij}\sigma_i\sigma_j \tag{11}$$

Donde σ_i^2 es la varianza de x_i o σ_i es la desviación estándar de x_i y ρ_{ij} es el coeficiente de correlación entre x_i y x_j.

Como $M = E\{\boldsymbol{X}\}$ la matriz de covarianza se puede escribir:

$$\Sigma = E\{\boldsymbol{XX^T}\} - E\{\boldsymbol{X}\}M^T - ME\{\boldsymbol{X^T}\} + MM^T = S - MM^T \tag{12}$$

Donde:

$$\boldsymbol{S} = E\{\boldsymbol{XX^T}\} = \begin{bmatrix} E\{x_1x_1\} & \dots & E\{x_1x_n\} \\ \vdots & \ddots & \vdots \\ E\{x_nx_1\} & \dots & E\{x_nx_n\} \end{bmatrix} \tag{13}$$

La matriz ***S*** se conoce como matriz de autocorrelación de ***X***. A pesar de que el vector esperado y la matriz de autocorrelación son parámetros importantes para caracterizar una distribución, en la práctica son desconocidos y deben estimarse a partir de un conjunto de muestras disponibles (generalmente se hace utilizando la técnica de estimación de muestra).

- **Estimación de muestras:** sea *y* una función de $x_1, \dots, x_n$ dada por $y = f(x_1, \dots, x_n)$ con valor esperado $m_y = E\{y\}$ y varianza $\sigma_y^2 = Var\{y\}$.

  Como se dijo anteriormente la función de densidad de *y* es desconocida o compleja de calcular. De ahí que es un ejercicio común remplazar el valor esperado de *y* por el promedio de las muestras disponibles:

$$\widehat{m_y} = \frac{1}{N}\sum_{k=1}^{N} E\{\boldsymbol{y_k}\} \tag{14}$$

  Donde $\boldsymbol{y_k}$ es calculada a partir de la k-ésima muestra x_k de la función *y*. Este cálculo es conocido como estimación de la muestra.

- **Momentos de estimaciones:** como la estimación de $\widehat{m_y}$ es la suma de *N* variables aleatorias, también es una variable aleatoria, caracterizada por un valor esperado y una varianza. El valor esperado de $\widehat{m_y}$ es:

$$E\{\widehat{m_y}\} = \frac{1}{N}\sum_{k=1}^{N} E\{\boldsymbol{y_k}\} = \frac{1}{N}\sum_{k=1}^{N} m_y = m_y \tag{15}$$

  Es decir, el valor esperado de la estimación es el mismo que el valor esperado de *y*. Del mismo modo la varianza de la estimación puede ser calculada como:

$$Var\{\widehat{m_y}\} = E\left\{\left(\widehat{m_y} - m_y\right)^2\right\} = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{k=1}^{N} E\{(\boldsymbol{y_k} - m_y)(\boldsymbol{y_k} - m_y)\}$$

$$Var\{\widehat{m_y}\} = \frac{1}{N^2}\sum_{k=1}^{N} E\{(\boldsymbol{y_k} - m_y)^2\} = \frac{1}{N}\sigma_y^2 \qquad (16)$$

La varianza de la estimación es *1/N* veces la varianza de *y*, así puede ser reducida a cero cuando *N* tiende a infinito.

Medidas de localización

Es necesario, antes de realizar un tratamiento complejo de datos, calcular ciertas medidas que brindan ideas preliminares de la naturaleza de los mismos. De esta manera se busca encontrar algún tipo de medida que permita caracterizar, diferenciar y distinguir los datos. En la investigación se hará uso de una medida de posición central (media) así como también una medida de dispersión (varianza), que permitan evaluar la variabilidad de las muestras

- **Medidas de posición central:** algunas de las medidas de posición central más conocidas se listan a continuación:

  - Media aritmética (μ): también conocida simplemente como media o promedio, se define por la siguiente ecuación:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (17)$$

    Donde x_i es cada observación i-ésima de la variable y *n* el número de datos.

  - Moda: se define como el valor con mayor frecuencia que se presenta en el conjunto de observaciones.

  - Mediana: Es el valor para el cual, cuando todas las observaciones se ordenan de manera creciente, la mitad de éstas son menores que este valor y la otra mitad son mayores. Sean las observaciones x_1, x_2, . . . , x_n una muestra aleatoria organizada en orden de magnitud creciente, la mediana de la muestra está dada por:

$$\tilde{x} = \begin{cases} x_{\frac{n+1}{2}} & si\ n\ es\ impar \\ \frac{1}{2}x_{\frac{n}{2}} + x_{\frac{n}{2}+1} & si\ n\ es\ par \end{cases} \quad (18)$$

- **Otras medidas de posición:** se definen los cuantiles de orden *k* como los valores de la variable ordenada de menor a mayor, que la dividen en *k* partes con la misma frecuencia de observaciones. Por lo tanto, existirán *k-1* cuantiles de orden *k*.

  El r-ésimo cuantil de orden *k* deja a su izquierda la fracción *r/k* de frecuencia de observaciones. Por ejemplo, el cuantil 15 de orden 100 deja por debajo el 15% de los valores del total de las observaciones.

  - Percentiles: son los 99 puntos que dividen la distribución en 100 partes, de tal modo que en cada parte, está incluido el 1% de los valores de la distribución.
  - Cuartiles: son los tres valores que dividen la distribución en 4 partes iguales, es decir en 4 intervalos, dentro de cada cual está incluido el 25% de los valores de la distribución.

  Cabe resaltar que la mediana es equivalente al valor del cuartil 2 (Q2), e igualmente al percentil 50 (P50).

Medidas de dispersión

El objetivo de estas medidas es determinar hasta qué punto las medidas de posición representan bien el conjunto de datos de la distribución. De esta manera, para complementar la información que se obtiene a partir de la media, es necesario otro tipo de parámetros que midan la dispersión o variabilidad de los datos.

- **Amplitud:** también conocida como rango, es la diferencia entre el valor máximo y el mínimo de la serie de datos.

- **Varianza:** compara cada dato de una distribución, con la media de la serie de datos. En los casos en los que se dispone de toda la población, la varianza es calculada mediante (19) y para aproximar la estimación de la varianza cuando se conoce de manera parcial la población se utiliza (20):

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad (19)$$

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad (20)$$

- **Desviación típica:** también llamada desviación estándar, permite obtener la medida de dispersión en las mismas unidades que la media, se calcula como la raíz cuadrada de la varianza. De la misma forma que para las dos expresiones anteriores, se calcula la desviación típica y cuasidesviación típica en (21) y (22) respectivamente:

$$\sigma = \sqrt{\sigma^2} \quad (21)$$

$$s = \sqrt{s^2} \quad (22)$$

Gráfica de caja y bigotes[4]

Como se explicó anteriormente, existen medidas de localización que dividen la distribución en cuantiles. La división de los datos en cuatro partes se hace mediante cuartiles, donde el tercer cuartil separa el cuarto (25%) superior del resto de los datos, el segundo cuartil es la mediana y el primer cuartil separa el cuarto (25%) inferior del resto de los datos (en la Figura 4 se muestra el diagrama de caja y bigotes). La caja central encierra el 50% de los datos en el rango intercuartil (R.I.) conteniendo la mediana y algunas veces la media. El rango intercuartil tiene como extremos el primer y tercer cuartil. Partiendo del centro de cada lado vertical de la caja se dibujan los bigotes (Walpole, Myers y Myers 1999):

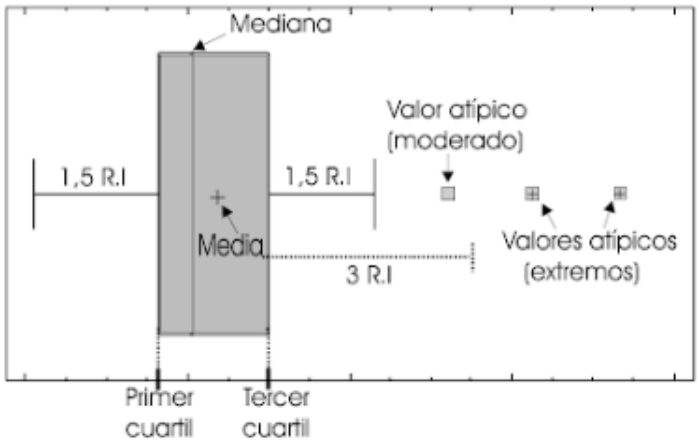
- El bigote de la izquierda tiene un extremo en el primer cuartil Q_1 y el otro en $Q_1 - 1.5\ R.I.$
- El bigote de la derecha tiene un extremo en el tercer cuartil Q_3 y el otro en $Q_3 + 1.5\ R.I.$

Los datos que se encuentran por fuera de los bigotes son denominados *valores atípicos*, por lo general son elementos para los cuales se han anotado sus valores

[4] WALPOLE, Ronald E. *et. al*. Probabilidad y estadística para ingenieros. México: Pearson. 1999.

en forma errónea o que por error fueron incluidos en el conjunto de datos; como también puede ser un elemento poco común que se haya anotado de forma correcta y que sí pertenece al conjunto de datos. Dentro de la investigación se usa un algoritmo que permite eliminar los datos que son considerados como atípicos (volumenes de carga y descarga demasiado altos), sí se encuentra fuera de los bigotes como se acabo de mencionar.

Figura 4. Diagrama de caja y de bigotes.



Fuente: Solano, Humberto Llinás; Álvarez, Carlos Rojas. Estadística Descriptiva y Distribuciones de Probabilidad.

Concepto de agrupamiento[5,6]

El objetivo del agrupamiento (clustering) es encontrar grupos de objetos en un conjunto de datos, de tal manera que en un grupo dichos objetos sean similares (o relacionados) y diferentes (o no relacionados) con los objetos de otros grupos. Entre mayor sea la homogeneidad dentro de un grupo y mayor sea la diferencia entre los grupos, mejor o más distinta es la agrupación.

A la hora de realizar la tarea de clasificación se suele asumir la existencia de etiquetas en los datos pertenecientes al conjunto de entrenamiento, de esta manera se realiza un aprendizaje supervisado. Sin embargo, existen también muestras que no cuentan con etiquetas, para lo cual es necesario desarrollar ciertas técnicas de

[5] JAMES, Gareth. et. al. An Introduction to Statistical Learning. New York: Springer. 2013.
[6] TAN, Pang Ning. *et. al*. Introduction to Data Mining. Pearson, 2013.

agrupamiento que permitan crear un etiquetado de objetos con etiquetas de clase (cluster), este aprendizaje es conocido como no supervisado.

Clasificación de agrupamiento

A continuación, se describirá brevemente la clasificación de algunos tipos de cluster según sea el conjunto de datos:

- **Bien-separado:** la distancia entre dos muestras cualesquiera en diferentes grupos es mayor que la distancia entre dos muestras dentro de un grupo. Los clusters bien separados pueden tener cualquier forma. Véase Figura 5(a).

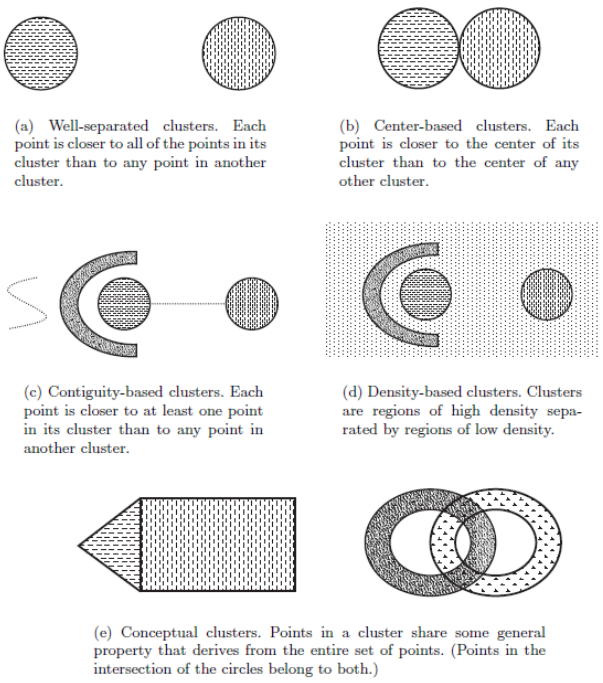
- **Basado en prototipos:** el prototipo de un cluster suele ser el centroide o promedio de todos los puntos del grupo, como también en datos categóricos, el prototipo es el medoide. En tales casos es común referirse a clusters basados en prototipos o basados en centros. Frecuentemente tales agrupaciones tienden a ser globulares. Véase Figura 5(b). Este tipo de cluster, es el que emplea el algoritmo k-means desarrollado en la presente investigación

- **Basado en gráficos:** hace referencia a un grupo de objetos que están conectados entre sí, pero que no tienen conexión con objetos fuera del grupo. Un ejemplo importante de clusters basados en gráficos son los basados en contigüidad, donde dos objetos están conectados solo si están dentro de una distancia especificada entre sí. Véase Figura 5(c).

- **Basado en densidad:** un cluster es formado por una región densa de objetos que está rodeada por otra de baja densidad. Una definición basada en la densidad se utiliza cuando los clusters son irregulares o entrelazados, y cuando el ruido y los valores atípicos están presentes. Véase Figura 5(d).

- **Propiedad compartida:** el cluster en este tipo es definido como un conjunto de objetos que cumplen alguna propiedad, así esta definición abarca todas las definiciones previas de un cluster como también incluye nuevos tipos de clusters. Véase figura 5(e).

Figura 5. Diferentes tipos de clusters ilustrados por conjuntos de puntos bidimensionales.



(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Fuente: Tan, Pang Ning. *et. al*. Introduction to Data Mining.

Técnicas de agrupamiento

Existe una gran cantidad de algoritmos de agrupamiento; según la literatura, entre las técnicas simples e importantes se encuentran:

- **K-medias (K-means):** es una técnica que se basa en prototipos, intenta encontrar una cantidad de *k* clusters, donde k debe ser especificada a priori, los clusters encontrados están representados por sus medias.

- **Agrupamiento jerárquico aglomerativo (Agglomerative Hierarchical Clustering, AHC):** hace referencia a la colección de técnicas de clustering que

realizan un agrupamiento jerárquico, comenzando con cada punto como un grupo y luego fusionando repetidamente los dos clusters más cercanos hasta que quede un solo grupo que abarque todas las muestras.

- **Agrupamiento espacial basado en densidad de aplicaciones con ruido (Density-based spatial clustering of applications with noise, DBSCAN):** esta técnica determina el número de clusters a partir de la distribución de densidad. Si existen regiones de baja densidad se clasifican como ruido y son omitidos.

K-means

Es uno de los algoritmos más antiguos y es ampliamente utilizado. Esta técnica está basada en prototipo según un centroide, que es usualmente la media del grupo de puntos. Este algoritmo requiere que el número de clusters *k* sea definido a priori, luego el algoritmo asignará a cada observación uno de los *k* clusters establecidos. Esta técnica se emplea en este trabajo con el fin de realizar un etiquetado de datos que permita agruparlos teniendo en cuenta las medias de las características de navegación.

En este algoritmo se considera una buena agrupación cuando la variación dentro del cluster es lo más pequeña posible. Sean $C_1, \ldots, C_k$ los conjuntos que contienen las observaciones de cada cluster. La variación dentro del cluster C_k es una medida $W(C_k)$ de la diferencia entre sí de las observaciones de un grupo. Así el problema se convierte en un ejercicio de optimización definida de la siguiente manera:

$$\min_{C_1 \ldots C_k} \left\{ \sum_{i=1}^{k} W(C_k) \right\} \tag{23}$$

En otras palabras, se quiere dividir las observaciones en *k* grupos de manera que la variación total dentro del grupo sumada a todos los *k* grupos, sea lo más pequeña posible. Sin embargo, para aplicar esta solución es necesario definir la variación dentro del cluster. En la literatura hay muchas formas posibles de definir este concepto, pero la opción más común implica una distancia euclidiana cuadrada dada por:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{24}$$

Donde $|C_K|$ denota el número de observaciones en el k-ésimo cluster. Es decir, la variación dentro del k-ésimo cluster es la suma del cuadrado de todas las distancias euclidianas emparejadas entre las observaciones pertenecientes al k-ésimo cluster, dividido por el número total de observaciones. De esta manera el problema de optimización que define la técnica de clustering es:

$$\underset{C_1 \ldots C_k}{minimize} \left\{ \sum_{i=1}^{k} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\} \tag{25}$$

Ahora, es necesario encontrar un método para dividir las observaciones en *k* grupos solucionando el problema de optimización. Es un problema muy difícil de resolver con precisión, ya que existen casi k_n maneras de dividir *n* observaciones en *k* clusters. Un algoritmo simple que generalmente proporciona un óptimo local es una buena solución para el problema de optimización, dicho algoritmo se describe a continuación:

1. Asignar de manera aleatoria un número de 1 a *k* a cada una de las observaciones. Estos sirven como asignaciones iniciales de cluster.
2. Iterar hasta que las asignaciones de cluster dejen de cambiar.
   a. Para cada uno de los *k* clusters, estimar su centroide. El centroide del k-ésimo cluster es el vector de las *p* medias características de las observaciones pertenecientes a ese cluster.
   b. Asignar cada observación al cluster cuyo centroide es más cercano.

Cuando el resultado ya no cambia se ha alcanzado un óptimo local. El nombre de k-means se debe a que en el paso 2 (a), los centroides se calculan como la media de las observaciones asignadas a cada grupo.

Máquina de soporte vectorial

Las máquinas de soporte vectorial aparecieron a principios de los años 90 como un clasificador de margen óptimo en el contexto de aprendizaje estadístico de Vapniks Desde entonces, este método de aprendizaje automático se ha convertido en una metodología estándar en las ciencias de la computación y la comunidad de la ingeniería, siendo aplicado satisfactoriamente a problemas de análisis de datos reales, frecuentemente otorgando mejores resultados si se le es comparado con otras técnicas.

Las SVM ofrecen grandes ventajas; entre ellas, se pueden lograr resultados en términos de clasificación y regresión que no involucran demasiadas muestras; este factor facilita la aplicación de las SVM en problemas de gran cantidad de datos, tales como procesamiento de texto y tareas bioinformáticas. Otra facilidad es que está técnica tiene un compromiso entre problemas que tienen un enfoque paramétrico y también para enfoques no paramétricos.

SVM para clasificación binaria de datos linealmente separables[7]

Dado un conjunto de datos separable

$$S = \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_i}, y_i)\}, \tag{26}$$

donde $\boldsymbol{x_i} \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$, d indica la dimensionalidad del espacio de entrada.

Es posible definir un hiperplano de separación como una función lineal que es capaz de separar dicho conjunto sin error, es decir,

$$g(x) = (x_1, y_1 + \dots + x_i, y_i) + b = < \boldsymbol{w}, \boldsymbol{x_i} > + b, \tag{27}$$

donde $\boldsymbol{w}$ y $b \in \mathbb{R}$. El hiperplano de separación cumplirá con las siguientes restricciones para todo $\boldsymbol{x_i}$ del conjunto de muestras:

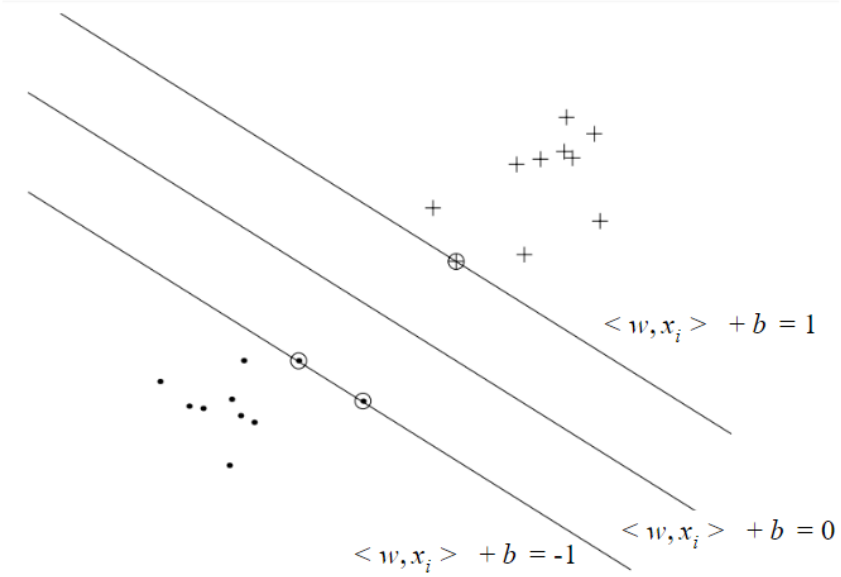
$$< \boldsymbol{w}, \boldsymbol{x_{i+}} > +b \geq 1 \text{ sí } y = +1 \tag{28}$$

$$< \boldsymbol{w}, \boldsymbol{x_{i-}} > +b \leq -1 \text{ sí } y = -1 \tag{29}$$

Esto significa que sí es una muestra positiva, la función de decisión en (28) le dará un valor de 1 o más (puntos ubicados en la parte superior derecha de la Figura 6) y por el contrario, sí es una muestra negativa la función de decisión en (29) le dará un valor de -1 o menos (puntos ubicados en la parte inferior izquierda de la Figura 6)

[7] CARMONA, Enrique. Tutorial sobre Máquinas de Vectores Soporte (SVM). Madrid,2014.

Figura 6. Ejemplo de hiperplano de mayor margen con vectores de soporte en los círculos.



Fuente: Cristianini, Nello; Taylor, John Shawe. An Introduction to Support Vector Machines.

Sin embargo, por facilidad matemática es conveniente unificar las anteriores ecuaciones en una, como sigue:

$$y_i < \boldsymbol{w}, \boldsymbol{x_i} > + b \geq 1 \tag{30}$$

De tal manera que se siguen cumpliendo las restricciones impuestas en (28) y (29); continuando con las operaciones:

$$y_i < \boldsymbol{w}, \boldsymbol{x_i} > + b - 1 \geq 0 \tag{31}$$

De la anterior ecuación se observa que todas las muestras tienen que ser iguales o mayores que cero y se añade la siguiente restricción que establece que todos los puntos que se encuentren sobre las rectas cumplen con

$$y_i < \boldsymbol{w}, \boldsymbol{x_i} > + b - 1 = 0 \tag{32}$$

A estos puntos se los conoce como los vectores de soporte.

El hiperplano que permite separar las dos clases no es único es decir, existen infinitos hiperplanos separables representados por aquellas rectas que cumplan por la restricción indicada en (31). Pero como se mencionó anteriormente, una de las características de la SVM es encontrar el hiperplano de separación óptimo. Para ello se define el concepto de margen de un hiperplano de separación como la mínima distancia entre dicho hiperplano y la muestra más cercana de cualquiera de las dos clases; por lo tanto, un hiperplano de separación se denominará óptimo sí su margen es máxima. Cabe resaltar que un hiperplano de separación óptimo tiene como propiedad equidistar las muestras más cercanas de cada clase.

Ahora bien, se conoce que la distancia entre una recta $L = A_x + B_x + C = 0$ y un punto $P(x_o, y_0)$ está dada por

$$d(LP) = \frac{|A_{x_0} + B_{y_0} + C|}{\sqrt{A^2 + B^2}}, \tag{33}$$

entonces se encuentra que la distancia entre el hiperplano intermedio y un punto ubicado en la recta del lado derecho, en donde se encuentran los vectores de soporte, es:

$$\frac{< \boldsymbol{w}, \boldsymbol{x_i} > + b}{\sqrt{w^2}}, \tag{34}$$

y de (32), $< \boldsymbol{w}, \boldsymbol{x_i} > + b = 1$ para un $y = +1$, por lo tanto:

$$\frac{< \boldsymbol{w}, \boldsymbol{x_i} > + b}{\sqrt{w^2}} = \frac{1}{||\boldsymbol{w}||} \tag{35}$$

Así la búsqueda del hiperplano óptimo se puede formalizar como el problema de optimización que busca encontrar el valor óptimo de $\boldsymbol{w}^*$ y b^* que minimiza $||\boldsymbol{w}||$, sujeto a la restricción (31), que es equivalente a:

$$\begin{aligned} \min \rightarrow \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 &= \frac{1}{2} < \boldsymbol{w}, \boldsymbol{w} > \\ s.a. \rightarrow y_i &< \boldsymbol{w}, \boldsymbol{x_i} > + b - 1 \geq 0 \end{aligned} \tag{36}$$

Lo anterior corresponde a un problema de programación cuadrático, que puede ser resuelto mediante la teoría de optimización. Debido a que tenemos que minimizar

una función sujeta a restricciones, es conveniente el uso de multiplicadores de Lagrange, así:

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 - \sum_{i=1}^{n} \alpha_i [y_i(< \boldsymbol{w}, \boldsymbol{x_i} > + b) - 1] \tag{37}$$

Donde los $\alpha_i \geq 0$ son los multiplicadores de Lagrange. El siguiente paso consiste en aplicar las condiciones de Karush-Kuhn-Tucker (KKT). Según estas condiciones, el problema de optimización primal tiene una forma dual si las funciones a optimizar y de restricciones son estrictamente convexas. De ahí que resolver el problema dual permite obtener la solución del problema primal, de esta manera se tiene:

$$\frac{\partial L(\boldsymbol{w}^*, b^*, \alpha)}{\partial \boldsymbol{w}} = \boldsymbol{w}^* - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = 0, \;\; i = 1, \ldots, n \tag{38}$$

$$\frac{\partial L(\boldsymbol{w}^*, b^*, \alpha)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0, \;\; i = 1, \ldots, n \tag{39}$$

$$\alpha_i [1 - y_i(< \boldsymbol{w}^*, \boldsymbol{x_i} > + b^*) = 0 \tag{40}$$

Al aplicar la primera condición de KKT se obtienen (38) y (39), y (40) se obtiene de aplicar la segunda condición de KKT, llamada condición complementaria. Despejando $\boldsymbol{w}^*$ de 38 se tiene

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i, \;\; i = 1, \ldots, n \tag{41}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0\,, \;\; i = 1, \ldots, n \tag{42}$$

Con las relaciones obtenidas, se procede a construir el problema dual. Reemplazando las condiciones 41 y 42 en 37, se llega a:

$$L(\alpha) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i\right)\left(\sum_{j=1}^{n} \alpha_j y_j \boldsymbol{x}_j\right) - \left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i\right)\left(\sum_{j=1}^{n} \alpha_j y_j \boldsymbol{x}_j\right) - \mathrm{b} \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \tag{43}$$

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < \boldsymbol{x_i}, \boldsymbol{x_j} > \right)_i \tag{44}$$

De ahí que el problema de optimización primal de (36), se ha trasformado en el problema de optimización dual descrito en (44), sujeto a las restricciones de 42 y las de los multiplicadores de lagrange. Reescribiendo el problema de optimización resulta:

$$\max \rightarrow L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < \boldsymbol{x_i}, \boldsymbol{x_j} > \right)$$

$$s.a \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0\,, \alpha_i \geq 0, i = 1, \ldots, n \tag{45}$$

De la misma manera que el problema de optimización primal descrito en (36), el problema dual de (45) se puede resolver mediante técnicas estándar de programación cuadrática. No obstante, es importante señalar que mientras el problema de optimización dual escala con un número de muestras *n*, el problema primal lo hace con dimensionalidad *d*. Aquí radica la ventaja del problema dual, reflejándose en el coste computacional.

Como se mencionó anteriormente, la solución del problema dual, α^*, nos lleva a resolver el problema primal. Para ello, solo basta con sustituir (41) en (27):

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^* y_i < \boldsymbol{x}, \boldsymbol{x_i} > + b^* \tag{46}$$

Nuevamente retomando las restricciones de (40), que se obtuvieron aplicando las condiciones complementarias de KKT, es posible decir que sí $\alpha_i > 0$, el segundo término de la izquierda de dicha ecuación tiene que ser cero, así que:

$$y_i < \boldsymbol{w}^*, \boldsymbol{x_i} > + b^* = 1 \tag{47}$$

De (47) se observa que, la muestra $(\boldsymbol{x_i}, y_i)$ satisface la restricción de (31) del problema primal, sin embargo se puede obtener otra conclusión y es la siguiente, considerando el caso “igual que” expresado en (32) correspondiente a los vectores de soporte, solo las muestras que tienen un $\alpha_i > 0$ cumplen con las restricciones de (32), de ahí que aquellas muestras serán vectores de soporte. Sin embargo, para que se pueda definir completamente el plano de (46), hace falta calcular el valor de *b** y este se obtiene de (47)

$$b^* = \frac{1}{y_{vs}} - < \boldsymbol{w}^*, \boldsymbol{x}_{vs} > \quad (48)$$

Donde $(\boldsymbol{x}_{vs}, y_{vs})$ representan la tupla de cualquier vector de soporte, en la práctica es más robusto obtener este valor promediando todos los vectores de soporte; es decir:

$$b^* = \frac{1}{N_{vs}} \sum_{1}^{N_{vs}} \frac{1}{y_{vs}} - < \boldsymbol{w}^*, \boldsymbol{x}_{vs} > \quad (49)$$

Finalmente se reemplaza el valor de (41) en (49) para calcular b^* en función de la solución del problema dual.

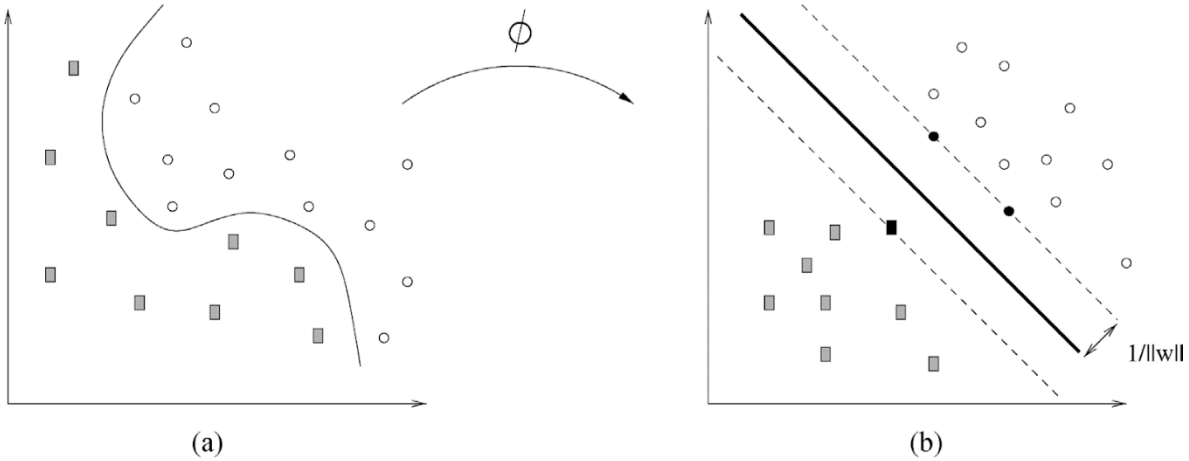
SVM para clasificación binaria de datos no separables linealmente

En la sección anterior se mostró que la SVM construye unos hiperplanos que son buenos clasificadores cuando los conjuntos de muestras son linealmente separables. Sin embargo, para el caso cuando las muestras no son separables, dichos hiperplanos lineales ya no funcionan. Por lo tanto, en esta sección se explica cómo conjuntos de funciones base no lineales, se pueden usar para definir espacios transformados de alta dimensionalidad y cómo encontrar hiperplanos óptimos en dichos espacios transformados que son denominados como espacio de características.

Sea $\varphi: X \rightarrow F$ la función de transformación que hace corresponder cada vector de entrada con un punto en el espacio de características F.

Donde $\varphi(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \dots, \varphi_m(\boldsymbol{x})]$ y $\exists\ \varphi_i(\boldsymbol{x}), i = 1, \dots, m\ \ tal\ que\ \varphi_i(\boldsymbol{x})$ es una función no lineal. Entonces el objetivo es construir un nuevo hiperplano de separación lineal óptimo en este nuevo espacio. La frontera de decisión lineal obtenida en el espacio de características se transformará en una frontera de decisión no lineal en el espacio original de entrada como se observa en la Figura 7.

Figura 7. Transformación de datos bajo la función kernel (a) Datos originales en el espacio de entrada (b) Datos mapeados en el espacio de características.



Fuente: Moguerza, Javier M; Muñoz, Alberto. Support vector machines with applications.

Teniendo en cuenta lo anterior, la función de decisión inyectiva en el espacio de características vendrá dada por

$$g(\boldsymbol{x}) = \left(w_1\varphi_1(\boldsymbol{x}) + \ \ldots + w_m\varphi_m(\boldsymbol{x})\right) = < \boldsymbol{w}, \boldsymbol{\varphi}(\boldsymbol{x}) > \tag{50}$$

Y de la misma forma, como se hizo para el caso de las muestras linealmente separables, su forma dual se obtiene transformando la expresión de frontera de (46) en:

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^* y_i \, K < \boldsymbol{x}, \boldsymbol{x_i} > \tag{51}$$

Donde $K < \boldsymbol{x}, \boldsymbol{x}' >$ corresponde a la famosa función kernel, que se define como una función $K: \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, que asigna a cada par de elementos del espacio de entrada $\mathbb{X}$, un valor real correspondiente al producto escalar de las imágenes de dichos elementos en un nuevo espacio de características así:

$$K(\boldsymbol{x}, \boldsymbol{x}') =< \varphi(\boldsymbol{x}), \varphi(x') > = (\varphi_1(\boldsymbol{x})\varphi_1(x') + \cdots + \varphi_m(\boldsymbol{x})\varphi_m(x')) \tag{52}$$

Donde $\varphi: X \to F$

De lo anterior, es posible decir que el producto escalar en la ecuación 46, puede ser sustituido por una función kernel. Así dado el conjunto de funciones base, $\varphi = \{\varphi_1(\boldsymbol{x}), \dots, \varphi_m(\boldsymbol{x})\}$ El problema de optimización de la ecuación 51 continúa siendo encontrar los valores de $\alpha_i^*, \quad i = 1, \dots, n$, que resuelven el problema dual de la ecuación 45, pero ahora expresado como:

$$\max \rightarrow L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j})\right)$$

$$s.a \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0\,, 0 \leq \alpha_i \leq C, i = 1, \dots, n \tag{53}$$

De lo anterior la función de decisión vendrá dada por (51), donde el valor de los parámetros $\alpha_i, i = 1, \dots, n$, se obtendrán como solución al problema de optimización cuadrático dado por (53), conocidos el conjunto de ejemplos de entrenamiento (x_i, y_i), i= $1, \dots, n$, el kernel K y el parámetro de regularización C. En cuanto al parámetro, no hay una base teórica que diga cómo encontrar el más óptimo, sin embargo, existe la heurística de usar un valor de C grande.

Funciones Kernel

Las funciones kernel en las máquinas de soporte vectorial son usadas para convertir un problema de clasificación no lineal en el espacio dimensional de entrada, a un problema de clasificación lineal en un espacio dimensional mayor. Como se mencionó anteriormente el problema de optimización usando una función kernel se estima según (51). Sin embargo, es importante mencionar que estas funciones pueden transformar el producto escalar en un espacio de características de mayor dimensión siempre y cuando se satisfagan las condiciones de Mercer.
Existen dos tipos principales de funciones kernel: las locales y las globales. En las funciones globales aquellas muestras que estén muy alejadas entre sí tienen impacto en el valor de la función kernel; sin embargo, en las funciones locales solo las muestras cercanas entre sí tienen impacto en el valor de la función. Un ejemplo de una global es el kernel polinomial, y un ejemplo de una local es el kernel de función de base radial (Radial Basis Function, RBF), de los cuales se va a hablar en seguida.

- **RBF kernel:** Es de las funciones kernel más usadas debido a su gran capacidad de aprendizaje. Una de las funciones más comunes de este tipo se indica en la siguiente expresión:

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = e^{-\frac{||\overrightarrow{x_i} - \overrightarrow{x_j}||^2}{2\sigma^2}} \quad (54)$$

El kernel RBF, tiene la capacidad de adaptarse bajo varias condiciones, entre estas: baja y alta dimensión, pocas y muchas muestras, etc. Además, cuenta con la ventaja de tener pocos parámetros. La capacidad del RBF es inversamente proporcional al parámetro σ, donde σ determina el área de influencia sobre el espacio de datos, de ahí que un σ más grande proporciona una superficie de decisión más uniforme y un límite de decisión más regular. Esto se debe a que un RBF con un σ grande hace que los vectores de soporte tengan gran influencia sobre un área grande. Por otra parte, si σ es pequeño, hace que aquellas muestras cuyas distancias son cercanas sean afectadas; por lo tanto, esto afectará a las muestras vecinas al punto de prueba, de ahí que entra en la categoría de funciones kernel locales.

- **Kernel polinomial:** Una función kernel polinomial se define según la ecuación 55.

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i}.\overrightarrow{x_j} + 1)^d \quad (55)$$

Donde $d \geq 2$, que indica el grado del kernel. Para el primer kernel $\binom{n+d-1}{d}$, las características distintivas son todos los monomios de grado d. Para el segundo kernel $\binom{n+d}{d}$, las características distintivas, son todos los monomios hasta el grado d. Por lo tanto, el límite de decisión en el espacio de entrada corresponde al hiperplano en el espacio de características que es una curva polinomial de grado d. A diferencia del anterior tipo de kernel esta función tiene gran capacidad de generalización, afectando así al valor del kernel global. Sin embargo, la capacidad de aprendizaje no es tan efectiva como RBF.

Estas funciones kernel fueron utilizadas para la ejecución del clasificador SVM, en donde se tiene en cuenta σ, el parámetro de regularización C y el grado del polinomio d. Posteriormente se realizó una comparación de resultados de precisión con cada uno de estos.

Método de validación cruzada de k-iteraciones

En ocasiones el conjunto de datos no es lo suficientemente extenso para dividirlo en un conjunto de entrenamiento y en otro de prueba. De modo que se pensaría en realizar el entrenamiento y la prueba con el conjunto completo. Sin embargo, este

enfoque es inadecuado, ya que el modelo tiende a funcionar con mayor precisión con aquellos datos con los que ha aprendido, pero falla cuando intenta clasificar datos desconocidos. Para solucionar este tipo de problemas se suele optar por la validación cruzada, un método que aplica el procedimiento prueba- entrenamiento, sobre diferentes particiones del conjunto de datos aproximadamente de igual tamaño de forma iterativa.

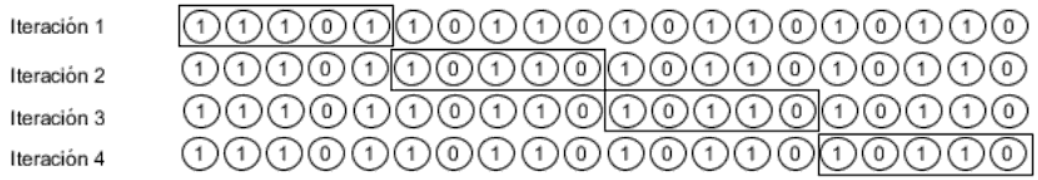
El método de validación cruzada de k iteraciones (K-fold Cross Validation, K-fold CV) para un conjunto de M datos, toma un parámetro de entrada $\alpha \in (0,1)$, correspondiente al $\alpha M\%$, de datos que se van a utilizar para realizar la prueba, en cada una de las k iteraciones. En la primera iteración se selecciona de forma aleatoria u ordenada $(1-\alpha)M\%$ de los datos como conjunto de entrenamiento; luego se ejecuta el correspondiente método de clasificación y se evalúa sobre el conjunto de prueba que se había seleccionado $\alpha M\%$. Este procedimiento se repite durante k iteraciones y da como resultado k estimaciones del error de prueba $MSE1, MSE2, \ldots, MSEk$. Al final se obtiene la media de aciertos calculada en porcentaje, este valor representa la bondad del modelo de clasificación.

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i \tag{56}$$

En la práctica un valor típico de rendimiento para el k-fold CV es $k = 5$ o $k = 10$. La diferencia o ventaja del valor de k radica en el costo computacional.

El proceso anteriormente descrito se muestra en la Figura 8. Se observa que hay un conjunto de entrenamiento de un tamaño de 20 piezas que pertenecen a dos clases 0 y 1. A partir de la primera iteración se toma el 25% del conjunto para la prueba y el 75% restante para el entrenamiento, así sucesivamente hasta completar todas las iteraciones.

Figura 8. Validación cruzada para *k=4*.



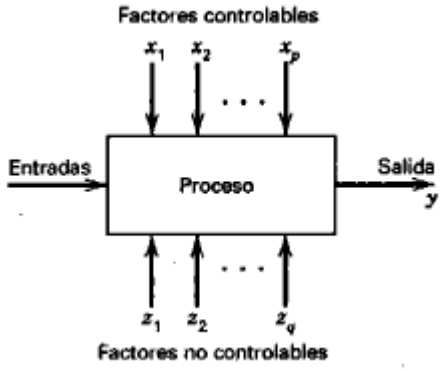
Fuente: Cestero, Eloy Vicente; Caballero, Alfonso Mateos. Data Science y redes complejas: métodos y aplicaciones.

Dentro de la investigación se realizó una validación 5-fold con $\alpha = 20$.

DISEÑO DE EXPERIMENTOS

Con el objetivo de desarrollar una investigación experimentalmente bien estructurada y que logre obtener buenos resultados, se consideró necesario realizar un diseño de experimento. Un experimento se considera un conjunto de pruebas en las que las variables de entrada de un proceso o sistema son modificadas, con el fin de identificar las razones de los cambios que presenta la salida. Las conclusiones y resultados que se pueden obtener dependen en gran medida de la manera en que los datos fueron obtenidos. Un proceso o sistema se puede representar como se indica en la Figura 9.

Figura 9. Modelo general de un proceso o sistema.



Fuente: Montgomery, Douglas C. Diseño y análisis de experimentos.

El proceso suele explicarse como una combinación de máquinas, métodos, personas u otros recursos que transforman la entrada en una salida que tiene respuestas observables. Las variables $x_1, x_2, \ldots, x_p$ son controlables, mientras que $z_1, z_2, \ldots, z_q$ no lo son.

Uno de los principios básicos del diseño experimental es la aleatorización, ya que si es formulada de manera correcta ayuda a “sacar del promedio” los efectos de factores extraños que posiblemente estén presentes. Cabe además resaltar que la mayoría de experimentos deberán ser iterativos, de esta manera se brindará una oportunidad de reconocer durante el desarrollo del experimento los factores importantes, su rango de variación, los métodos y unidades de medición adecuados para cada respuesta.

A continuación, se muestran los aspectos generales para diseñar experimentos:

- **Variable de respuesta:** Proporciona información útil acerca del proceso bajo estudio, en la mayoría de los casos suelen ser utilizados el promedio y/o la desviación estándar.

- **Factores:** Lo que el investigador cambia o manipula.

- **Niveles:** Los posibles valores que toma cada factor.

- **Parámetros:** Lo que se mantiene constante en la realización del experimento.

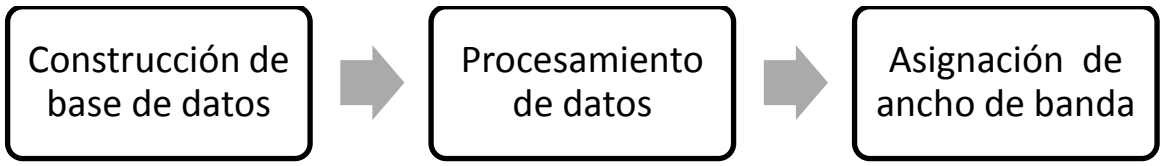
- **Repetición:** Hace referencia a una corrida del experimento para obtener un valor de la variable de respuesta.

En la presente investigación, este método fue empleado para llevar a cabo la consignación y elección de la precisión del clasificador de una manera más clara y ordenada utilizando los diferentes modelos y funciones kernel propuestos a lo largo de la metodología realizada.

1. METODOLOGÍA Y RESULTADOS

Se siguió una metodología basada en el diagrama de bloques mostrado en la Figura 10, de tal forma que se ilustra los procesos seguidos en la realización del proyecto.

Figura 10. Etapas del problema de investigación.



Fuente: Diagrama de procesos elaborado por los autores

A continuación, se muestran todos los procedimientos que se siguieron para lograr los objetivos de la investigación.

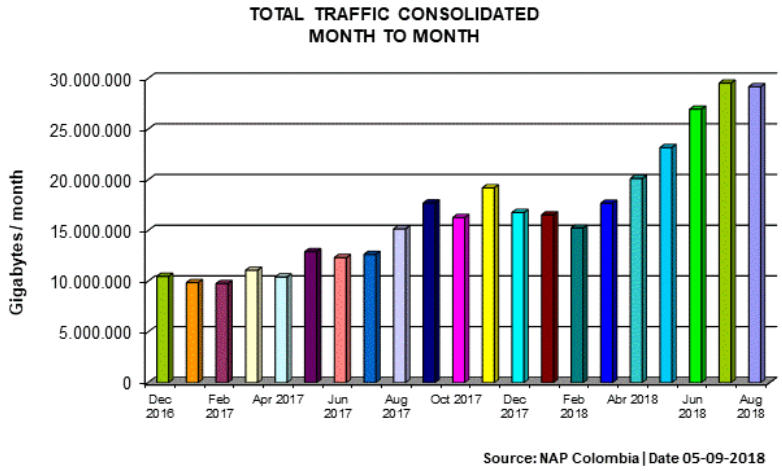
1.1. Construcción de la base de datos

1.1.1. Selección ventana de observación

Para realizar una conveniente caracterización del comportamiento de los usuarios fue necesario estimar las variables en periodos donde haya una mayor actividad por parte de ellos. NAP Colombia posibilitó acceder a estadísticas acerca de la distribución del tráfico de los usuarios colombianos. En la Figura 11, se observó la distribución por meses a partir de diciembre del 2016 hasta agosto del 2018; se apreció que el tráfico aumento en este periodo de tiempo. Sin embargo, se debió tener en cuenta que el experimento se desarrolló en una universidad, donde las actividades académico-administrativas se suspenden en vacaciones. Por tal motivo las medidas se realizaron durante el periodo académico. Se tuvo en cuenta la misma razón para escoger los días de la semana, se toma 3 días aleatorios de lunes a viernes durante estos meses.

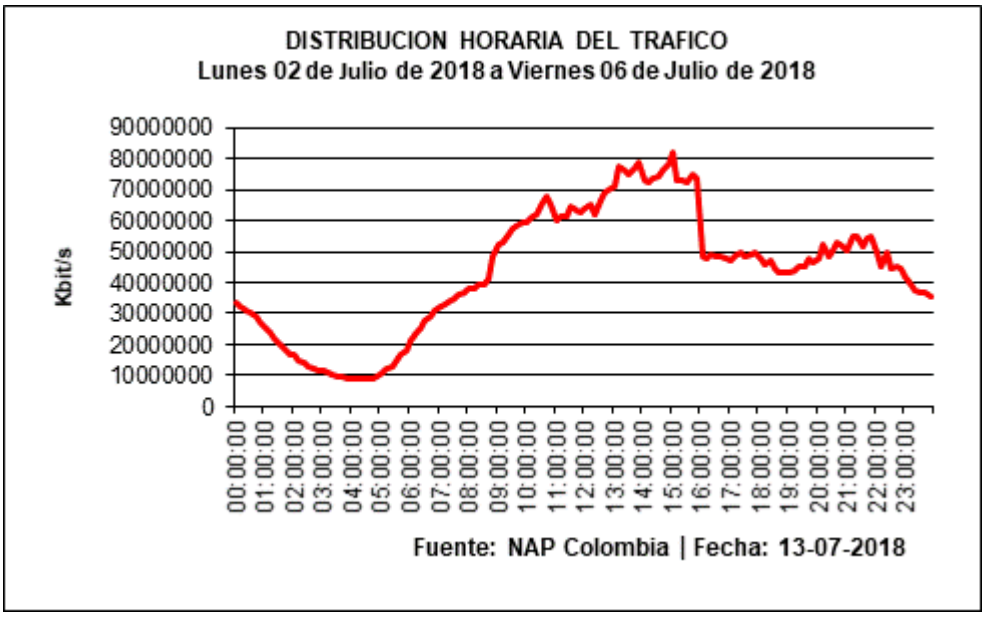
En la Figura 12 se indica el promedio de la distribución del tráfico horaria durante una semana, se apreció que el periodo donde hubo un crecimiento en el consumo se encuentra entre las 9 y las 16 horas.

Figura 11. Tráfico total consolidado por meses.



Fuente: NAP Colombia.

Figura 12. Distribución horaria de tráfico.



Fuente: NAP Colombia.

Según lo anterior se decidió obtener los datos de consumo de los usuarios de la red en los siguientes periodos de tiempo:

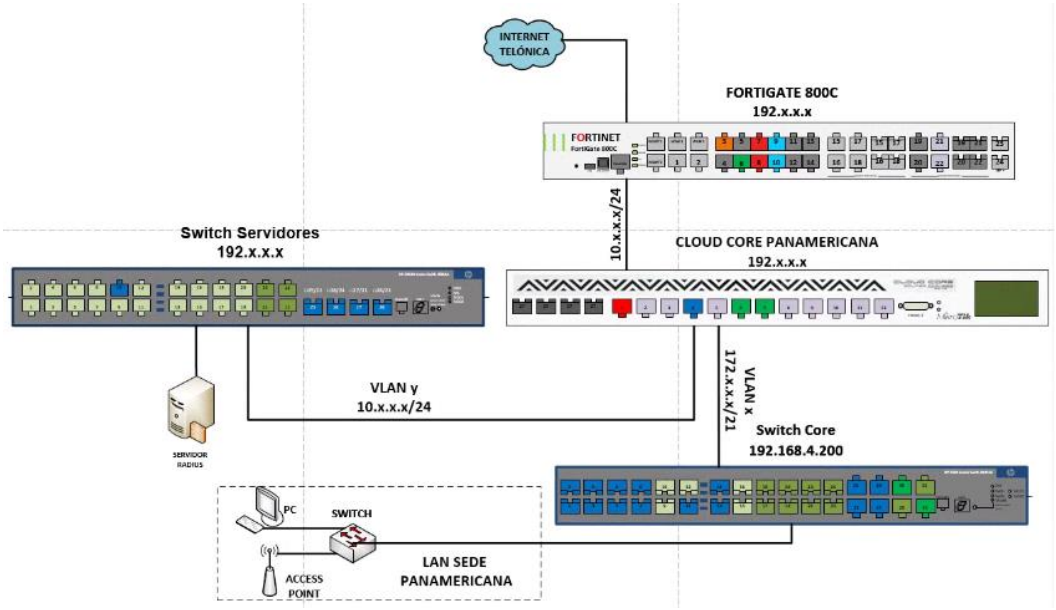
- **Meses:** marzo, abril, mayo (2018)
- **Días:** 3 días aleatorios de lunes a viernes
- **Horas:** 9 am a 4 pm

1.1.2. Topología red LAN Universidad de Nariño

Una vez se seleccionó la ventana de observación, se procedió a realizar la recolección de la información general de la red LAN y WLAN de la Universidad de Nariño, esta información fue adquirida con colaboración del aula de informática y la administración de redes de la universidad de Nariño. Esta etapa nos ayudó a comprender el funcionamiento básico de la red para proceder a tomar datos.

El esquema principal de la red de la sede panamericana Universidad de Nariño se muestra en la Figura 13. El servicio de internet dedicado se proporcionó a través de un equipo del proveedor, que se conectó al equipo FortiGate 800C que corresponde al equipo de red principal que administra las reglas de seguridad, la entrada y salida de internet, entre otras funciones. Este se conectó al Cloud Core ccr1016-12g de la sede panamericana que se encargó de administrar toda la red inalámbrica de ésta, a través de una VLAN x (x corresponde a un ID de la VLAN), que se conectó al conmutador principal que repartió la conexión al resto de bloques que tienen sus correspondientes conmutadores y puntos de acceso, para que los usuarios lograran conectarse. En cuanto al servidor RADIUS, también estableció la conexión con el Cloud Core con el fin de realizar funciones de autenticación de los clientes.

Figura 13. Topología red LAN sede panamericana Universidad de Nariño.



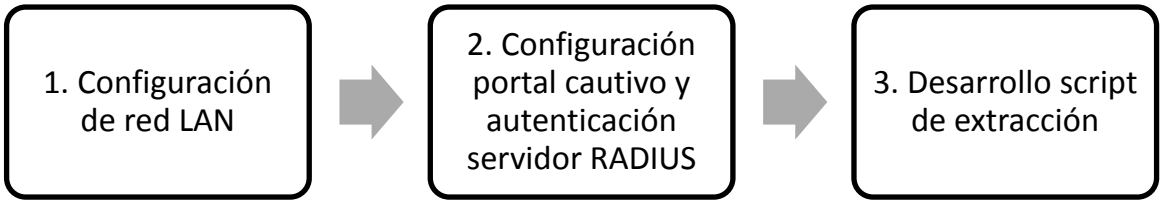
Fuente: Adaptación topología de la red LAN elaborada por los autores.

1.1.3. Configuración Cloud Core ccr1016-12g

Para tomar los registros de la base de datos se desarrolló el script sobre el enrutador principal que administra los puntos de acceso; dicho equipo es el Cloud Core ccr1036-12g-4S, un enrutador de calidad de administración de la marca Mikrotik.

Los pasos de configuración dentro del enrutador de administración se indican en la Figura 14.

Figura 14. Configuración de software de Cloud Core ccr1036-12g-4S para la recolección de datos.



Fuente: Diagrama de procesos de la configuración del Cloud Core elaborado por los autores

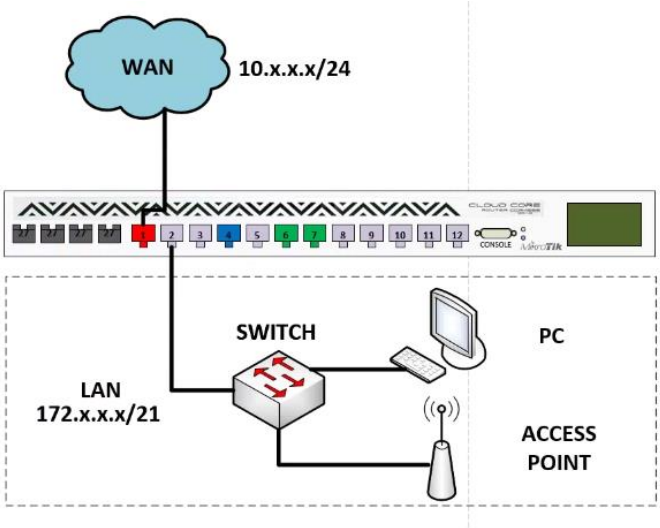
A continuación, se realiza una descripción que resume los pasos descritos en el anterior diagrama de bloques. Sin embargo, es necesario recalcar que los procesos 1 y 2 del anterior diagrama de bloques, correspondieron la red de prueba de portal cautivo de la universidad de Nariño, por lo cual ya se encontraban implementados, y no se aplicó ningún cambio sobre este sistema. Así, se indican los aspectos más importantes abordados desde una perspectiva general de configuración de enrutadores Mikrotik.

- **Configuración red LAN:** Para la configuración de la red, se tuvo en cuenta dos direcciones IP muy importantes comúnmente denominadas IP-WAN e IP-LAN. La primera dirección IP, es la entrada de internet al enrutador, mientras que la segunda dirección IP hizo referencia a la red interna que se ha creado con el equipo de red. En la Figura 15 se muestra un ejemplo de una red con un equipo mikrotik, junto a la dirección IP aparece la máscara de red, un parámetro muy importante que se tuvo en cuenta a la hora de definir la extensión de la red.

  Para continuar con la configuración del equipo, fue necesario también considerar aspectos como el servidor DHCP, para la asignación dinámica de las direcciones

IP; el servidor DNS para la administración del espacio de nombres de dominio; una regla NAT para intercambiar paquetes entre direcciones incompatibles y la creación de una ruta para acceder a la red WAN a través de su puerta de enlace.

Figura 15. Ejemplo de configuración de una red.



Fuente: Ilustración de una configuración de red LAN elaborada por los autores.

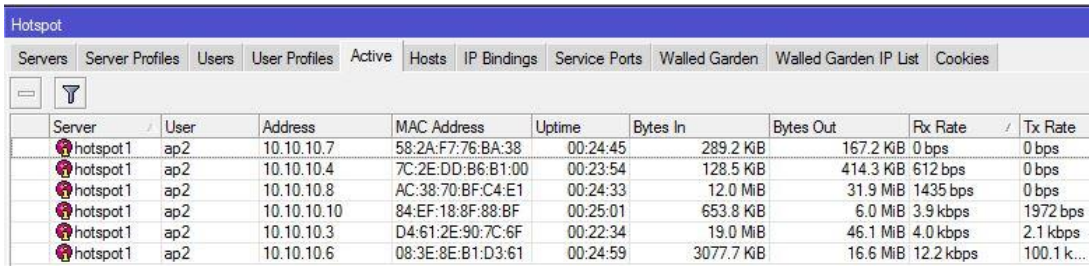
- **Configuración portal cautivo y autenticación servidor RADIUS:** Los portales cautivos basados en el estándar 802.11 ofrecieron internet a través de una red inalámbrica y un enrutador. En RouterOS un portal cautivo tuvo varias características, como la autenticación de clientes usando una base de datos interna o usando un servidor RADIUS externo, como es el caso del portal cautivo que se configuró en la red de prueba de la universidad de Nariño, entre otras ventajas están la creación de perfiles o usuarios individuales y cambios visuales como por ejemplo la pantalla principal de login.

  En la Universidad de Nariño el servidor RADIUS, se utilizó con el fin de tener un registro y control en una base de datos de las personas que podían acceder a internet. En el portal cautivo existieron perfiles para estudiantes, docentes y algunos administrativos, de esta manera, los estudiantes podían acceder a la red con su código estudiantil y los profesores y administrativos con su cedula de ciudadanía.

- **Desarrollo script de extracción:** Una vez que el sistema de autenticación a través de un portal cautivo fue implementado, se procedió a desarrollar el script

que permitió extraer la información deseada. En la Figura 16 se muestra una interfaz dinámica de los hosts activos en la herramienta hotspot de RouterOS.

Figura 16. Interfaz gráfica de usuarios activos en el hotspot del Cloud Core.



Server	User	Address	MAC Address	Uptime	Bytes In	Bytes Out	Rx Rate	Tx Rate
hotspot1	ap2	10.10.10.7	58:2A:F7:76:BA:38	00:24:45	289.2 KiB	167.2 KiB	0 bps	0 bps
hotspot1	ap2	10.10.10.4	7C:2E:DD:B6:B1:00	00:23:54	128.5 KiB	414.3 KiB	612 bps	0 bps
hotspot1	ap2	10.10.10.8	AC:38:70:BF:C4:E1	00:24:33	12.0 MiB	31.9 MiB	1435 bps	0 bps
hotspot1	ap2	10.10.10.10	84:EF:18:8F:88:BF	00:25:01	653.8 KiB	6.0 MiB	3.9 kbps	1972 bps
hotspot1	ap2	10.10.10.3	D4:61:2E:90:7C:6F	00:22:34	19.0 MiB	46.1 MiB	4.0 kbps	2.1 kbps
hotspot1	ap2	10.10.10.6	08:3E:8E:B1:D3:61	00:24:59	3077.7 KiB	16.6 MiB	12.2 kbps	100.1 k...

Fuente: Usuarios activos en la interfaz de winbox utilizada por los autores.

En la Tabla 2 se encuentran consignados los campos extraídos del hotspot, todos estos campos fueron guardados en el equipo de red en un archivo de texto plano .txt. Se tomaron capturas en intervalos de 5 minutos que registraron los consumos en ese instante de tiempo sucesivamente en el horario de 9:00 am a 4:00 pm, en total se obtuvieron 85 archivos diarios. Se registraron hasta un máximo de 70 datos debido a limitaciones del equipo que no permitió guardar archivos superiores a 4 MB.

Tabla 2 Campos extraídos de la herramienta hotspot mikrotik

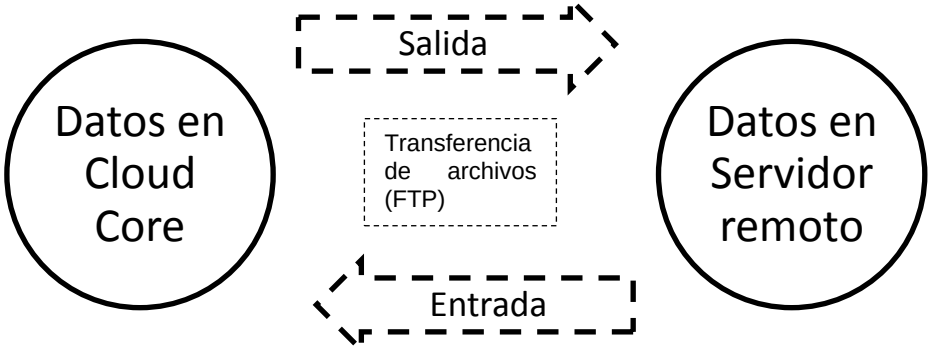
Campos	**User**	**MAC**	**Bytes In [B]**	**Bytes Out [B]**	**Uptime [s]**
Información proporcionada	Tipo de usuarios: Profesor Administrativo Estudiante	Identificación única de la tarjeta de red del equipo	Volumen de descarga	Volumen de carga	Tiempo de conexión en la red

Fuente: Elaborada por los autores.

En este trabajo se propuso realizar un análisis de datos a través de machine learning, por lo cual fue obligatorio que la base de datos se haya extraído a un equipo externo que corresponde al computador que desarrolló el procesamiento de los datos. En este sentido, la extracción de los datos se realizó mediante el protocolo de transferencia de archivos (File Transfer Protocol, FTP). La

transferencia de archivos se hizo a un servidor, en este caso una computadora personal y se utilizó el Administrador de Internet Information Services (IIS) de Windows. Este protocolo permitió descargar los datos del Cloud Core, así como también cargarlos, lo cual resultó muy útil para etapas posteriores. Este proceso se muestra en la Figura 17.

Figura 17. Transferencia de archivos mediante un protocolo FTP.



Fuente: Diagrama de un proceso de transferencia de archivos elaborado por los autores.

En la Tabla 3 se consigna la información más relevante del script 1, que permitió la construcción de la base de datos.

Tabla 3 Información del script 1

Nombre del Script	Extracción datos de consumo
Parámetros de entrada	Variable fecha (string), Variable etiqueta de hora (int)
Parámetros de salida	User, MAC, Bytes In, Bytes Out, Uptime
Descripción	Permite extraer los campos (parámetros de salida) para cada usuario activo en un instante de tiempo con intervalos de 5 minutos, de 9 am (etiqueta hora = 1) a 4 pm (etiqueta hora = 85).
Lenguaje	Mikrotik Scripting Language
Pseudocodigo	1. INICIO

	2. Inicializar variables (fecha, hora) 3. PARA hora=1 hasta hora=85 hacer 4. crear archivo fecha&&hora.txt 5. PARA CADA usuario activo en hotspot hacer 6. escribir en archivo = [User, MAC, Uptime, Bytes In, Bytes Out, hora] 7. FIN PARA CADA 8. transferencia archivo vía FTP 9. retardo 5 minutos 10. FIN PARA

Fuente: Insumo para construcción de la base de datos elaborado por los autores.

1.1.4. Elaboración de base datos mediante SQL

Continuando con el proceso, se optó por desarrollar una base de datos bien estructurada. Para ello se desarrolló un script en el lenguaje de consulta estructurado (Structured Query Language, SQL). Para realizar un adecuado procesamiento de datos, se añadió un nuevo campo que hace referencia a una etiqueta de día. Se utilizó el sistema de gestión de base de datos MySQL para la elaboración. En la Figura 18 se muestra una fracción de una tabla de la base de datos.

Figura 18. Fragmento de base de datos desarrollada mediante el lenguaje SQL.

Usuario	Mac	T_conexion	bytes_in	bytes_out	hora	dia
218028144	5C:51:81:EF:34:5B	00:58:25	570541	1270411	1	lu11
217122182	54:B1:21:B3:F3:EE	00:54:35	392683	1104382	1	lu11
216165101	88:B4:A6:39:02:E0	01:00:18	9179733	65316170	1	lu11
216035102	18:21:95:1E:8A:2F	00:23:37	29574	131730	1	lu11
facea	E8:B4:C8:3E:63:BE	00:55:52	816038	17925278	1	lu11
facea	18:D2:76:DB:65:CA	00:23:31	5144640	3466631	1	lu11
216165008	24:92:0E:C0:3E:FA	00:59:40	887742	10308331	1	lu11
facea	7C:2E:DD:55:C3:F2	00:38:16	887001	2308671	1	lu11
2141272014	E8:93:09:8E:BB:AF	01:36:20	5158894	1815391	1	lu11
217027045	00:73:E0:7A:56:22	01:31:32	10649559	157949991	1	lu11
218116118	14:30:C6:CC:F2:97	00:58:51	172854	818359	1	lu11
217089353	20:A9:0E:42:6A:B0	01:18:50	100544	216264	1	lu11
217095172	84:BE:52:E4:B9:FD	00:21:53	140934	538162	1	lu11
facea	CC:9F:7A:77:62:3D	01:00:29	257776	624265	1	lu11
invitado	B8:EE:65:DD:1B:7E	00:18:49	2328346	31939770	1	lu11

Fuente: Interfaz gráfica del servidor local elaborada por los autores

En la siguiente tabla se muestra la información más relevante del script.

Tabla 4. Información del script 2

Nombre del Script	Elaboración base de datos en MySQL
Parámetros de entrada	Tabla (nombre), variables (varchar): User, MAC, Bytes In, Bytes Out, Uptime, etiqueta de hora, etiqueta de día
Parámetros de salida	Base de datos de las observaciones durante los 3 meses
Descripción	Permite la lectura de los archivos .txt y la conformación de una tabla que articula los 85 archivos diarios y a su vez esta tabla hace parte de la base de datos final, durante los tres meses.
Lenguaje	SQL
Pseudocodigo	1. INICIO 2. CREAR tabla 3. Inicializar variables: User, MAC, Bytes In, Bytes Out, Uptime, etiqueta de hora, etiqueta de día 4. FIN CREAR 5. CREAR procedimiento 6. PARA hora=1 hasta hora=85 hacer 7. cargar datos de archivo fecha&&hora.txt en tabla 8. FIN PARA 9. FIN procedimiento

Fuente: Lectura de los archivos para SQL, creación propia

1.1.5. Resultados de la construcción de la base de datos

En la Tabla 5 se consignan las características de la base de datos. En la Tabla 6 se encuentran el número de registros por periodos de tiempo, en total se obtuvieron 184609 registros, con un promedio de 5594 y 5590 por día y por mes respectivamente.

Tabla 5 Características de la base de datos

Lugar de recolección de datos	Universidad de Nariño sede Panamericana
Equipo para el desarrollo de scripts	Cloud Core ccr1016-12g
Sistema operativo de desarrollo	RouterOS
Tiempo de Observación	Marzo, abril y mayo de 2018
Días de observación	3 días a la semana, aleatorios
Horario de Observación	9:00 a.m a 4:00 p.m
Campos extraidos del Hotspot	User, Mac Address, Bytes In, Bytes Out, Uptime
Intervalo de captura de datos	5 minutos
Máximo de Registros tomados por captura de datos	70
Formato de archivo de la base de datos elaborada	Texto plano (.txt)
Número de archivos Generados por día	85

Fuente: Elaborada por los autores.

Tabla 6 Registros totales obtenidos

Mes		N° registros	Total registros	Promedio
Marzo	Martes 6	5745	50438	5604
	Miércoles 7	5741		
	Jueves 8	5747		
	Miércoles 14	5738		
	Jueves 15	5069		
	Viernes 16	5247		
	Martes 20	5734		
	Miércoles 21	5732		
	Jueves 22	5685		
Abril	Lunes 2	5806	73757	5674
	Martes 3	5804		
	Miércoles 4	5815		
	Martes 10	5786		
	Miércoles 11	5811		

	Viernes 13	5367		
	Lunes 16	5372		
	Miércoles 18	5735		
	Jueves 19	5728		
	Lunes 23	5794		
	Martes 24	5796		
	Miércoles 25	5275		
	Lunes 30	5668		
	Jueves 3	5807		
	Viernes 4	5819		
	Lunes 7	5812		
	Martes 8	5788		
	Viernes 11	5608		
Mayo	Miércoles 16	4546	60414	5492
	Jueves 17	5767		
	Viernes 18	5655		
	Martes 22	4030		
	Miércoles 23	5826		
	Jueves 24	5756		

Fuente: Elaborada por los autores.

Sin embargo, se debió tener en cuenta que los registros anteriormente tomados incluyeron a los usuarios más de una vez. Debido a que el interés principal fue conocer el tiempo de conexión total y los valores de consumo del usuario, se realizó un script en Python que permitió establecer una conexión con la base de datos en MySQL, cuyo propósito fue tomar el último registro de los datos, siempre que el usuario sea repetido. De esta manera se trabajó con un total de 17049 registros, como se consignan en la Tabla 7.

Tabla 7 Registros finales

Mes		N° registros	Total registros	Promedio
	Martes 6	553		
	Miércoles 7	577		
	Jueves 8	520		
	Miércoles 14	573		
Marzo	Jueves 15	435	4585	509
	Viernes 16	360		
	Martes 20	532		
	Miércoles 21	542		

	Jueves 22	493		
	Lunes 2	543		
	Martes 3	553		
	Miércoles 4	525		
	Martes 10	557		
	Miércoles 11	546		
	Viernes 13	537		
Abril	Lunes 16	555	7023	540
	Miércoles 18	554		
	Jueves 19	531		
	Lunes 23	588		
	Martes 24	569		
	Miércoles 25	429		
	Lunes 30	536		
	Jueves 3	523		
	Viernes 4	513		
	Lunes 7	554		
	Martes 8	597		
	Viernes 11	501		
Mayo	Miércoles 16	434	5441	495
	Jueves 17	517		
	Viernes 18	472		
	Martes 22	361		
	Miércoles 23	493		
	Jueves 24	476		

Fuente: Elaborada por los autores.

1.1.6. Análisis estadístico de la base de datos

Con el fin de que las conclusiones obtenidas a partir del experimento resultaran objetivas y adecuadas, se realizó un análisis estadístico de los datos. Los métodos gráficos simples desempeñaron un rol de importancia a la hora de interpretar y analizar los datos. Conviene subrayar que estos métodos proporcionaron pautas generales en cuanto a la confiabilidad y validez de los resultados.

Con la ayuda del software de programación Python se realizó la evaluación estadística, obteniendo tanto datos numéricos como también gráficos. En la Tabla 8 se muestran las principales medidas estadísticas por meses de las variables, se infirió que existe mucha variabilidad en los datos. En la Figura 22 se muestran los diagramas de caja y bigotes para cada una de las variables, la línea naranja indica

la mediana de la muestra y los puntos negros considerados como los valores atípicos ya que se encuentran a 1.5 veces del rango intercuartil.

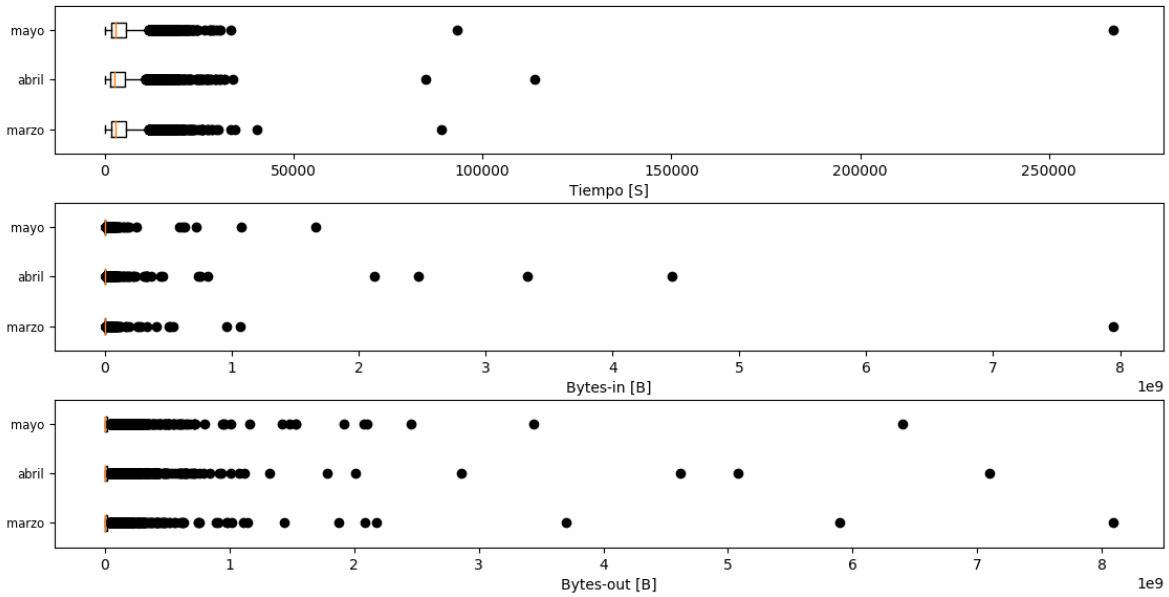
Tabla 8 Medidas estadísticas de las variables de respuesta

Mes	Tiempo conexión [s] μ [+/- σ]	Bytes-in [B] μ [+/- σ]	Bytes-out [B] μ [+/- σ]
Marzo	4315.16 [+/- 4394.45]	6.10×10^6 [+/- 1.21×10^8]	2.90×10^7 [+/- 1.80×10^8]
Abril	3899.97 [+/- 4218.88]	5.63×10^6 [+/- 8.02×10^7]	2.72×10^7 [+/- 1.44×10^8]
Mayo	4348.7 [+/- 5627.88]	4.20×10^6 [+/- 3.33×10^7]	3.04×10^7 [+/- 1.40×10^8]
Total	4154.84 [+/-4762.25]	5.30×10^6 [+/- 8.32×10^7]	2.87×10^7 [+/-1.53×10^8]

Fuente: Elaborada por los autores

En la Figura 19, se muestra claramente que existen datos que pudieron ser interpretados como atípicos. Sin embargo, cabe resaltar que la base de datos fue tomada en una institución educativa donde los estudiantes fueron en cantidad significativamente mayor que los demás funcionarios y en su gran mayoría su tiempo de conexión fue menor comparado con el resto de usuarios, además también está incluido el comportamiento de los administrativos que cumplieron jornadas laborales que por lo general eran de 8 horas diarias, así su consumo en internet fue mayor. Por añadidura solían darse los casos en los que algunas personas se dedicaron a descargar contenido de internet como películas, juegos, programas etc, en consecuencia, su consumo también fue elevado y por ende se marca una diferencia significativamente alta con respecto de los demás usuarios. Esta información fue observada de la base de datos obtenida y explicaba el comportamiento de los datos. Por tal motivo todas las muestras obtenidas en el desarrollo del experimento fueron aceptables y fue apropiado realizar el tratamiento de los datos con los valores considerados como atípicos según el diagrama de caja y bigotes, así como también sin ellos.

Figura 19. Diagrama de caja y bigotes de las variables de consumo por meses.

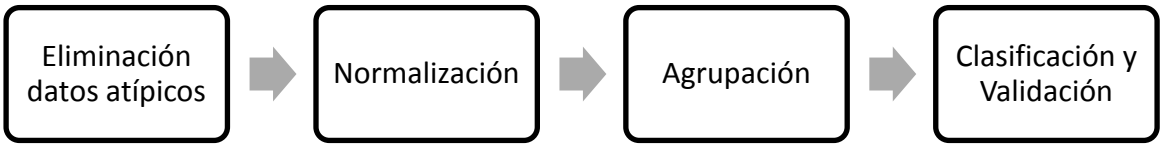


Fuente: Ilustración de datos atípicos elaborada por los autores en Python.

1.2. Procesamiento de datos

En esta etapa se tuvieron en cuenta las pautas para el desarrollo de un experimento. En este sentido, se explica el procedimiento que se llevó a cabo utilizando el lenguaje de programación de alto nivel Python. Los algoritmos de machine learning se desarrollaron haciendo uso de la librería Scikit Learn. El script que realizó el procesamiento de los datos debió tener instrucciones que permitieron realizar la conexión entre la base de datos del servidor local en MySql con Python. Su desarrollo se realizó mediante etapas como se indica en la Figura 20.

Figura 20. Diagrama de bloques del procesamiento de los datos.



Fuente: Diagrama de procesos elaborado por los autores.

1.2.1. Selección de factores y niveles

Los factores y niveles que influyeron en el diseño del experimento para el procesamiento de datos se muestran en la Tabla 9. Las posibles combinaciones de estos se indican en la Tabla 10. De esta manera la realización del experimento se fundamentó en completar la Tabla 10, donde fue consignado el resultado final que consta del promedio de la precisión de clasificación luego de repetir el experimento 10 veces. La selección de los factores y sus niveles se desarrolló y justificó a lo largo de esta sección.

Tabla 9 Factores y rangos en el procesamiento de datos

Factores		**Niveles**		
Preprocesamiento de datos	Diagrama de caja y bigotes	Datos originales		Datos sin atípicos
	Normalización	Sin normalizar		Normalizados
Algoritmo de clasificación	SVM	Lineal	Polinomial (2,3)	RBF

Fuente: Elaborada por los autores.

Tabla 10 Modelo final del resultado del experimento

Media de Precisión (10 iteraciones)	**Datos originales**		**Datos sin atípicos**	
	Sin normalizar	**Normalizados**	**Sin normalizar**	**Normalizado**
Lineal	P1	P2	P3	P4
Polinomial	P5	P6	P7	P8
RBF	P9	P10	P11	P12

Fuente: Elaborada por los autores.

1.2.2. Eliminación datos atípicos

Como se mencionó anteriormente, según el diagrama de caja y bigotes se pudo considerar un valor atípico cuando éste se encontraba a más de 1.5 veces el rango intercuartil y dicho valor pudo ser estimado como erróneo, así como también pudo ser una muestra poco común en la variable medida. Según lo anterior, se procedió a eliminar los valores que podrían ser considerados atípicos y se realizó la estimación de las variables estadísticas resultantes. Estos datos son consignados

en la Tabla 11. Se dedujo que las variables que presentaron menor y mayor variabilidad son respectivamente el tiempo de conexión y Bytes-out.

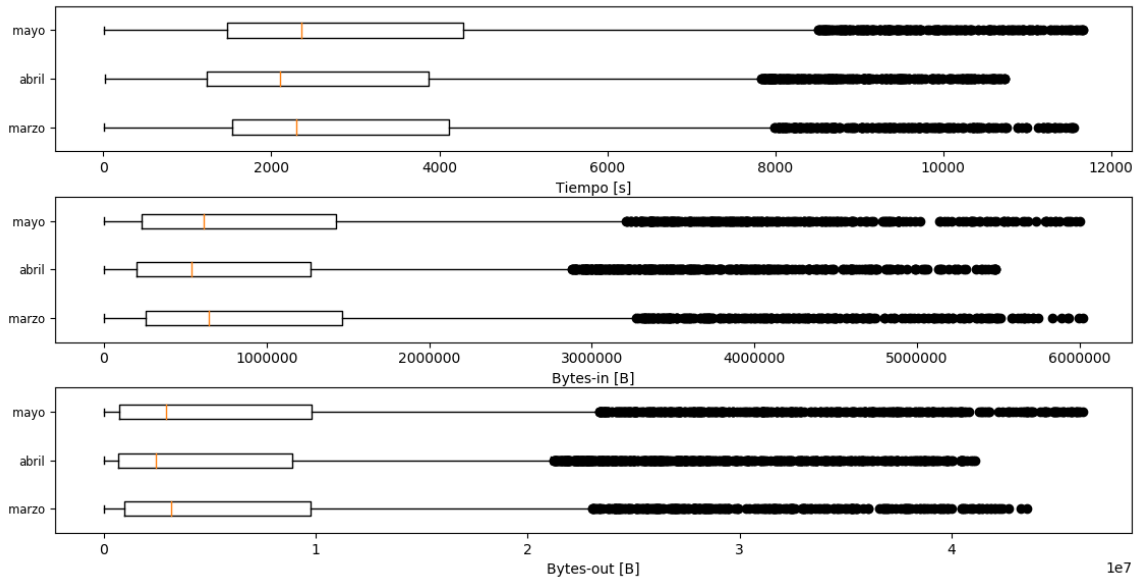
Tabla 11 Medidas estadísticas con valores atípicos eliminados

Mes	Tiempo conexión [s] μ [+/- σ]	Bytes-in [B] μ [+/- σ]	Bytes-out [B] μ [+/- σ]
Marzo	3069.39 [+/-2392.76]	1053321.28 [+/-1134695.47]	7123295.88 [+/-9004409.01]
Abril	2820.17 [+/-2297.52]	914318.10 [+/-1008575.63]	6425537.34 [+/-8587871.18]
Mayo	3111.34 [+/-2490.03]	1022904.87 [+/-1111555.72]	7191607.63 [+/-9653761.03]
total	2939.74 [+/-2376.90]	959733.18 [+/-1052310.58]	6752253.19 [+/-9042701.32]

Fuente: Elaborada por los autores.

En la Figura 21, se indica el diagrama de caja y bigotes para los datos sin los valores considerados como atípicos, se apreció que las muestras restantes fueron dispersas, sin embargo fueron en menor medida con respecto de las anteriores.

Figura 21. Diagrama de caja y bigotes para las variables de consumo sin valores atípicos.

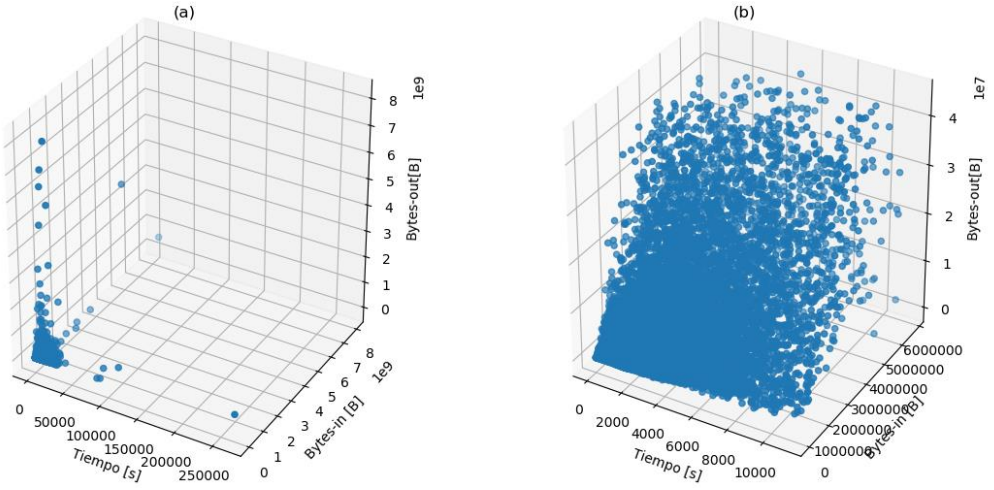


Fuente: Diagrama de eliminación de datos atípicos elaborado en Python por los autores.

En la Figura 22 se indica el diagrama de dispersión para los registros obtenidos del repositorio y los datos cuyos valores por fuera de 1.5 veces el rango intercuartil

fueron removidos. Se apreció que existieron valores con una diferencia en magnitud significativamente alta. Para ejemplificar, obsérvese el dato de mayor magnitud en el eje z correspondiente a bytes-out en la Figura 22(a), el cual tiene un valor de descarga superior a 8 GB a diferencia de la mayoría de los datos que se aprecian en la Figura 22(b) que tienen un consumo promedio aproximado de descarga de 20 MB.

Figura 22. Graficas de dispersión de las variables del sistema. (a) Datos originales, (b) Datos sin valores atípicos.



Fuente: Diagrama tridimensional de las variables del sistema elaborado en Python por los autores.

La información general del script desarrollado para la eliminación de los datos considerados atípicos se consigna en la Tabla 12.

Tabla 12 Información del script 3

Nombre del Script	Eliminación datos atípicos
Parámetros de entrada	Base de datos (proveniente de MySQL)
Parámetros de salida	Arreglo de variables (float): Bytes In, Bytes Out, Uptime.

Descripción	Elimina los datos que no se encuentran en 1.5 veces el rango intercuartil de las tres variables
Lenguaje	Python
Pseudocodigo	1. INICIO 2. cargar base de datos 3. Adecuación datos a arreglos 4. CREAR procedimiento 5. Calcular percentiles 25 y 75 6. Calcular bigotes 7. PARA i=1 hasta longitud arreglo 8. SI arreglo[i]>bigote menor & arreglo[i]<bigote mayor 9. almacenar valor 10. FIN SI 11. FIN PARA 12. FIN procedimiento 13. Graficar

Fuente: Elaborada por los autores.

1.2.3. Normalización

Fue necesario llevar a cabo este tipo de procesamiento, debido a que en las Figuras 19 y 21 se observó que las variables tienen diferentes escalas. La normalización de los datos correspondió a ajustar las 3 muestras obtenidas en una escala común. De esta manera fueron normalizados los datos obtenidos en una escala de 0 a 1, según (57).

$$x_n = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{57}$$

Donde x_n indicó el dato normalizado, x_i el dato original, x_{min} y x_{max} los valores mínimo y máximo respectivamente del total de las muestras que hicieron parte de una variable.

1.2.4. Agrupación

Esta etapa se desarrolló, puesto que fue necesario realizar un etiquetado de los datos de manera que permitió crear los grupos correspondientes a los perfiles de usuarios. Dicho agrupamiento se realizó haciendo uso del algoritmo K-means, esta

es una técnica de aprendizaje no supervisado cuyo objetivo principal fue agrupar los datos en *k* grupos (clusters), donde el valor de *k* debió ser considerado a priori.

Una de las principales cuestiones a la hora de llevar a cabo dicho algoritmo, fue definir un valor de *k* grupos adecuados, esto se debía a que el algoritmo es en cierta medida ingenuo ya que realiza la agrupación de los datos para cualquier valor de *k* sin presentar un error de ejecución, realizando agrupaciones incluso si *k* no es el numero correcto de grupos que se debe utilizar.

En la presente investigación se hizo uso del método de elbow (codo), que ejecutó el algoritmo K-means para un rango de valores de *k* y para estos valores se calculó la suma de los errores cuadrados (Sum of Square Error, SSE) con (55).

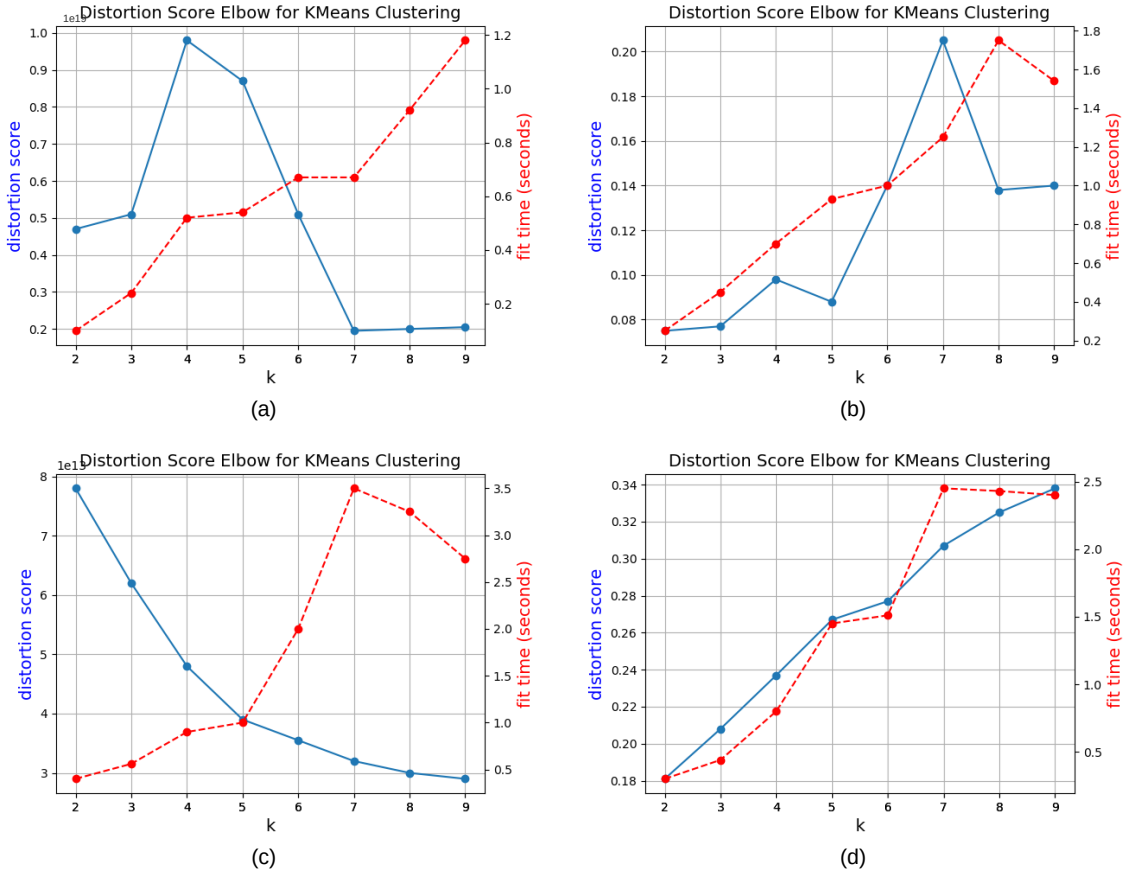
$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist(c_i, x)^2 \tag{55}$$

Donde $dist$ es la distancia euclidiana, x un objeto, C_i el i-ésimo cluster y c_i el centro del cluster C_i.

Finalmente se trazó una línea con los valores resultantes del *SSE* para cada valor de *k*; si el grafico se parecía a un brazo, el valor del "codo" fue considerado como un valor adecuado de *k*.

En la Figura 23, se observan las curvas resultantes de ejecutar el método de elbow, (línea azul) indica el valor del SSE y el tiempo (en rojo) que tarda el algoritmo para cada valor de k. El valor de SSE tiende a disminuir cuando el valor de k aumenta, sin embargo existen "brazos" tanto hacia arriba como hacía abajo y esto fue válido cuando había un punto de inflexión fuerte, lo cual indicó que el modelo se ajustó mejor en ese punto. Los valores de *k* que fueron considerados como adecuados se consignan en la Tabla 13.

Figura 23. Curvas resultantes de la aplicación de la técnica de elbow en los escenarios propuestos (a) Datos originales sin normalizar, (b) Datos originales normalizados, (c) Datos sin valores atípicos sin normalizar, (d) Datos sin valores atípicos normalizados.



Fuente: Gráficas de selección de *K* adecuado elaborado por los autores en Python.

Tabla 13 Valores de ***K*** adecuados

	Datos Originales		**Datos sin atípicos**	
	No Normalizados	**Normalizados**	**No Normalizados**	**Normalizados**
K	3	3	5	5

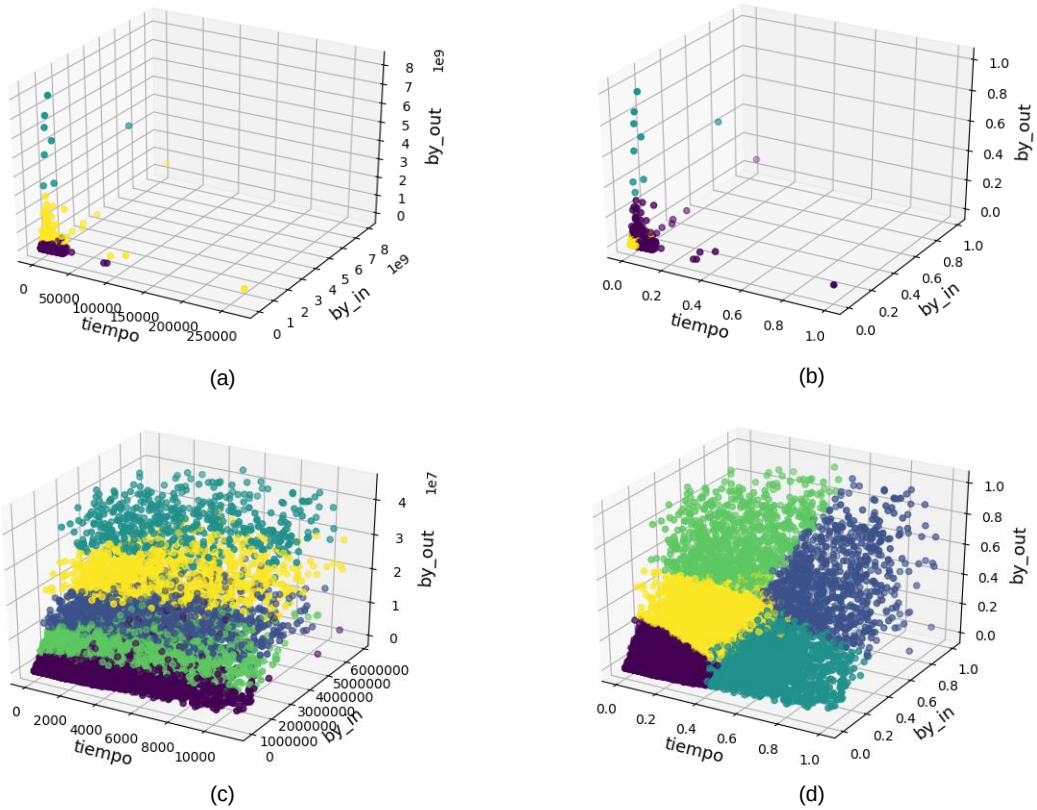
Fuente: Elaborada por los autores.

Es importante aclarar que en la Figura 23 se apreciaron distintos valores de *k* adecuados, sin embargo para su selección también se tuvo en cuenta la etapa de aplicación posterior, que consistió en un modelo de asignación de ancho de banda

por perfiles de navegación, dichos perfiles estaban directamente relacionados con los números de clusters; de manera que, si se seleccionaba un valor de *k* alto (*k*=8, 8 perfiles de navegación), aunque el método de elbow indicaba que es correcto, no era pertinente para la aplicación.

En la Figura 24, se muestran las gráficas resultantes de la aplicación de la técnica de K-means sobre los datos. Se apreció que en general el algoritmo depende en gran medida del valor de la proporción de las variables, lo cual se identificó claramente en las Figuras 24 (a) y (c), donde los grupos resultantes se forman de acuerdo a la magnitud de bytes-out. Esto se debió a que la técnica se basó en las medias de los grupos y por esta razón se organizó a favor de la variable que otorga mayor valor en magnitud, esta es una justificación principal para realizar una normalización de los datos. Las Figuras 24 (b) y (d) mostraron una distribución de grupos donde la importancia a las tres variables se dio en igual medida.

Figura 24. Resultado gráfico de la aplicación de K-means en los escenarios propuestos (a) Datos originales sin normalizar, (b) Datos originales normalizados, (c) Datos sin valores atípicos sin normalizar, (d) Datos sin valores atípicos normalizados.



Fuente: Gráficas de agrupación elaboradas por los autores en Python.

En la Tabla 14, se muestra la información del script desarrollado.

Tabla 14 Información del script 4

Nombre del Script	Agrupación utilizando M-medias
Parámetros de entrada	Variable número de grupos: M (int) Arreglo de datos: Datos originales: Normalizados, sin normalizar Datos sin atípicos: normalizados, sin normalizar
Parámetros de salida	Etiquetas de grupo (0,1,2,3,4)
Descripción	Agrupa los datos teniendo en cuenta la media.
Lenguaje	Python (uso de librería sickit learn)
Pseudocodigo	1. INICIO 2. cargar arreglo de datos (diferentes casos) 3. definir M (curva elbow) 4. llamar función kmeans con M 5. ejecutar algoritmo kmeans 6. graficar

Fuente: Elaborada por los autores.

1.2.5. Clasificación y validación

El método de predicción basado en el aprendizaje de máquina seleccionado para realizar la tarea de clasificación fue la máquina de soporte vectorial, esta técnica básicamente tuvo como objetivo dos funciones; encontrar un hiperplano de separación optimo que maximice el margen entre la muestra más cercana de cada clase al hiperplano trazado y encontrar una función kernel que permita mapear los datos cuando estos no son linealmente separables, en un espacio de características de una dimensión mayor al espacio de entrada. Para la presente investigación fueron utilizadas las funciones kernel lineal, polinomial de grado 2, 3 y RBF, con el objetivo de realizar una comparación de la precisión con la cual clasifican.

La validación de la SVM se desarrolló teniendo en cuenta dos etapas:

- **Selección de parámetros:** Existieron dos parámetros asociados con el kernel RBF y polinomial: C y γ (gamma) que jugaron un papel crucial en el rendimiento de la SVM. De ahí que una selección inadecuada de estos parámetros pudo traer consigo problemas de sobre ajuste o bien, de ajuste insuficiente del modelo. No obstante, en la literatura existieron guías para determinar estos parámetros adecuadamente, en donde fueron sugeridas guías prácticas usando una técnica de búsqueda de cuadricula que utilizaban el método de validación cruzada que fue aplicado en este estudio.

  Las razones por las cuales se hizo el uso de la técnica de búsqueda de cuadricula se exponen a continuación. En primer lugar, el uso de métodos de aproximación o heurísticas, no evitaban realizar búsquedas excesivas de parámetros. En adición, la búsqueda de cuadricula pudo ser fácilmente ubicada, debido a que cada par (C, γ) eran independientes. Cabe resaltar que esta técnica fue principalmente utilizada para encontrar los parámetros de la función kernel RBF; sin embargo, dichos parámetros también se adaptaban a la función kernel lineal, polinomial y sigmoidal.

  El parámetro C reguló la clasificación correcta de las muestras de entrenamiento, contra la maximización del margen de la función de decisión. Para valores de C grandes se aceptaba un margen menor siempre que la función de decisión sea mejor para clasificar correctamente todos los puntos de entrenamiento. Un valor más bajo de C implicó un margen mayor y una función de decisión más simple. De ahí que C se comportó como un parámetro de regularización de la SVM.

  El comportamiento del modelo fue bastante sensible al parámetro γ. Si este era demasiado grande, el radio del área de influencia de los vectores de soporte sólo incluía el propio vector de soporte y ninguna cantidad de regularización de C evitaba el sobreajuste. Cuando gamma fue muy pequeño, el modelo era demasiado restringido y no pudo capturar la complejidad o la "forma" de los datos. La región de influencia de cualquier vector de soporte seleccionado incluía todo el conjunto de entrenamiento.

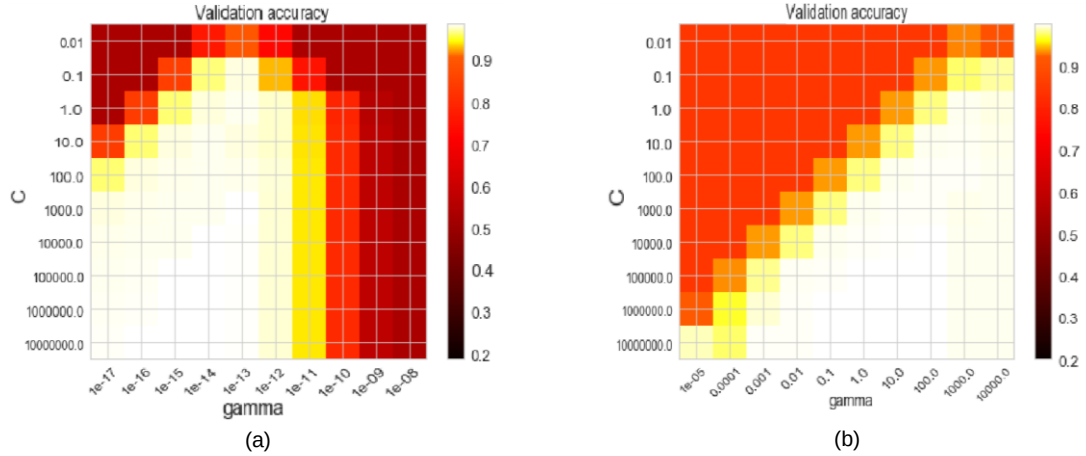
  La librería sickit learn de Python, posibilitó realizar la técnica de búsqueda de cuadricula a través de un método de visualización en forma de mapa de calor, que correspondió a una representación gráfica sencilla de comprender. El mapa de calor contó con una barra de colores que posibilitó localizar la puntuación de la precisión resultante de evaluar la SVM con el método de validación cruzada en la intersección de los parámetros C y γ.

En la Figura 28, se aprecian los resultados de aplicar la técnica de búsqueda de cuadricula para el repositorio de datos en los diferentes escenarios propuestos. En los scripts utilizados se manipuló el rango de búsqueda de los parámetros en una escala logarítmica, esto permitió una mayor facilidad para encontrar valores apropiados.

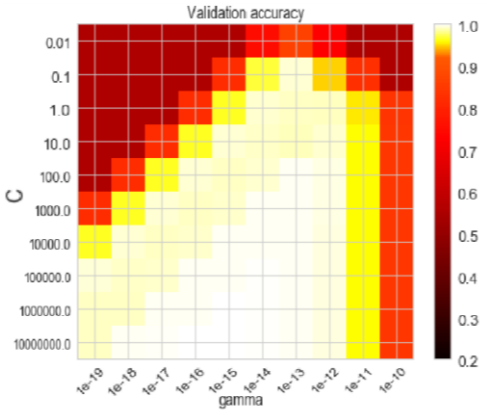
Es posible notar que para los cuatro casos el área en la cual se consideraron los parámetros como adecuados fue amplia, donde la puntuación de la precisión fue mayor a 0.95 (colores cercanos a blanco). Esto implicó que la tarea de clasificación no representó mayor complejidad para los escenarios planteados con el presente repositorio. Los valores de gamma considerados adecuados de la Figura 25 (a) y (c) son excesivamente pequeños, esto coincidió con la magnitud de las variables de los datos que no fueron normalizados, caso contrario ocurre para los datos que cumplieron la etapa de normalización, y esto, es razonable ya que gamma aumentó.

Otra apreciación de la Figura 25 es que existen valores que contaron con el mismo valor de puntuación de precisión, es decir para varios valores de γ y C. Sin embargo, para favorecer los modelos que usaban menos memoria y que son más rápidos de predecir fue preferible seleccionar valores de C bajos. Consecuentemente los valores de C y γ considerados adecuados para la tarea de clasificación se consignan en la Tabla 15. Con estos valores fueron entrenados los modelos de SVM para las cuatro funciones de kernel.

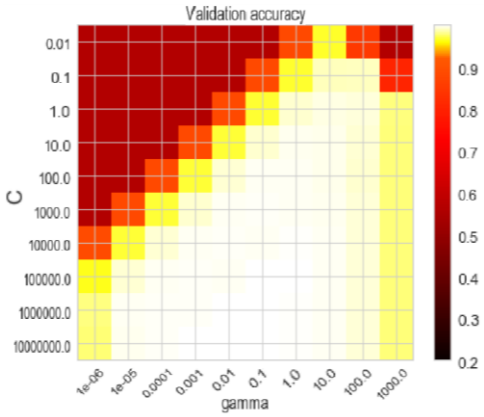
Figura 25. Mapas de calor para selección de C y γ en los escenarios propuestos (a) Datos originales sin normalizar, (b) Datos originales normalizados, (c) Datos sin valores atípicos sin normalizar, (d) Datos sin valores atípicos normalizados



(a) (b)



(c)



(d)

.

Fuente: Gráficas para seleccionar los valores adecuados de rendimiento de la SVM elaborada por los autores.

Tabla 15 Valores de (C, γ) seleccionados

Parámetro	Datos Originales		Datos sin atípicos	
	No Normalizados	Normalizados	No Normalizados	Normalizados
C	1	10	100	1
γ	1e-13	100	1e-13	10

Fuente: Elaborada por los autores.

- **Validación modelos de clasificación:** Por lo general las técnicas basadas en aprendizaje de máquina ajustan el modelo con un conjunto de entrenamiento para posteriormente hacer predicciones sobre datos que no fueron entrenados, de esta manera las muestras utilizadas fueron divididas en un conjunto de entrenamiento y uno de prueba, pero esta situación presentó un problema cuando dicha división no era aleatoria, ya que podría darse el caso en el que un subconjunto tenga muestras de una sola clase, no obstante las técnicas de validación cruzada evitaron este inconveniente.

  El algoritmo que se utilizó en la presente investigación para realizar la validación cruzada es el denominado k-folds, que entregó el valor de la precisión para cada iteración en subconjuntos de entrenamiento y prueba. Posteriormente se promediaron los resultados por cada iteración para obtener la precisión del modelo.

La técnica de validación se ejecutó para los cuatro modelos de las SVM de los escenarios propuestos. Según la literatura los valores recomendados eran de k=5 y k=10, de esta manera en esta investigación se optó por realizar una validación 5-folds; realizando el entrenamiento con el 80% de los datos y la prueba con el 20% restante; esto se realizó en 5 iteraciones y se estimó la media y desviación estándar correspondiente a la precisión del modelo.

En la Tabla 16 se consigna la media y desviación estándar resultantes de realizar el procedimiento anteriormente descrito 10 veces, con el fin de evaluar el rendimiento del sistema y cumplir con las repeticiones establecidas anteriormente en el diseño del experimento. Se percibió que los valores de precisión de los modelos son favorables, esto fue una buena indicación de que el diseño del experimento se desarrolló correctamente.

En primeria instancia se pudo considerar a los datos atípicos como erróneos, sin embargo, como se estipulo anteriormente este término es ambiguo, ya que depende de la naturaleza de los datos, y a lo largo de esta sección se observó que el funcionamiento final de los modelos para los cuales se tenían en cuenta estos valores, no fue afectado.

Las medidas de precisión altas de todos los modelos de clasificación evaluados en la presente investigación indicaron que se realizó una adecuada calibración de los valores de C y γ.

El modelo seleccionado para la ejecución del sistema correspondió a la SVM entrenada que utilizó una función kernel **polinomial de grado 2** para el escenario de datos sin datos atípicos y normalizados. Una de las razones de la selección se debió al valor de precisión, además fue necesario seleccionar un modelo en el que se tuvo en cuenta las tres variables del sistema y esto correspondió a los modelos normalizados, adicionalmente el modelo con datos originales no fue viable utilizarlo en la aplicación ya que el sistema propuesto en la siguiente etapa, tardó un tiempo considerable para realizar la correspondiente asignación dinámica que dependió del consumo del usuario. La precisión del modelo seleccionado se muestra subrayada de color verde en la Tabla 16.

Tabla 16 Resultado del experimento

Media de Precisión (10 iteraciones)		Datos originales		Datos sin atípicos	
		Sin normalizar	Normalizados	Sin normalizar	Normalizados
Lineal	**Iter1**	1	0.9768	1	0.9884
	Iter2	0.9992	0.9701	0.9989	0.9866
	Iter3	1	0.9739	0.9956	0.9830
	Iter4	0.9989	0.9675	0.9990	0.9880
	Iter5	0.9994	0.9736	0.9993	0.9902
	M	0.9995	0.9724	0.9986	0.9872
	(+/-σ)	(+/- 0.0002)	(+/- 0.0033)	(+/- 0.0015)	(+/- 0.0024)
Polinomial Grado 2	**Iter1**	0.9884	0.9959	0.9996	0.9953
	Iter2	0.9866	0.9944	0.9982	0.9967
	Iter3	0.9830	0.9950	0.9993	0.9957
	Iter4	0.9880	0.9947	0.9996	0.9953
	Iter5	0.9902	0.9956	0.9989	0.9956
	M	0.9872	0.9951	0.9991	0.9957
	(+/-σ)	(+/- 0.0024	(+/- 0.0005)	(+/- 0.0005)	(+/- 0.0005)
Polinomial Grado3	**Iter1**	1	0.9877	1	0.9960
	Iter2	0.9980	0.9850	0.9989	0.9964
	Iter3	0.9984	0.9848	0.9993	0.9942
	Iter4	1	0.9839	0.9993	0.9942
	Iter5	0.9994	0.9856	0.9989]	0.9960
	M	0.9992	0.9854	0.9993	0.9954
	(+/-σ)	(+/- 0.0003)	(+/- 0.0013)	(+/- 0.0004)	(+/- 0.0010)
RBF	**Iter1**	0.9903	0.9985	0.9993	0.9942
	Iter2	0.9906	0.9977	0.9989	0.9953
	Iter3	0.9909	0.9974	0.9986	0.9917
	Iter4	0.9915	0.9977	0.9996	0.9935
	Iter5	0.9912	0.9976	0.9986	0.9920
	M	0.9909	0.9977	0.9990	0.9933
	(+/-σ)	(+/- 0.0004)	(+/- 0.0004)	(+/- 0.0004)	(+/- 0.0014)

Fuente: Elaborada por los autores.

Los resultados consignados en la Tabla 16, representaron la etapa más importante de la investigación, puesto que, los modelos de clasificación indicaron valores de precisión altos; esto significó que las técnicas de máquinas de aprendizaje pudieron ser empleadas para realizar la clasificación de usuarios en una red inalámbrica, lo cual correspondió a uno de los retos de esta investigación. Además el sistema propuesto fue innovador con respecto a los modelos de asignación de ancho de banda, ya que estos no eran enfocados en emplear técnicas de machine learning.

La información del script ejecutado para la presente etapa se encuentra consignada en la Tabla 17.

Tabla 17 Información del script 5

Nombre del Script	Clasificación y Validación
Parámetros de entrada	Arreglo de datos con sus respectivas etiquetas: Datos originales: Normalizados, sin normalizar Datos sin atípicos: Normalizados, sin normalizar Parámetros del clasificador SVM: C y gamma Función kernel del clasificador: lineal, polinomial o RBF Número de iteraciones: K = 5 Número de repeticiones del experimento: 10
Parámetros de salida	Arreglo de precisiones (media y desviación estándar) Modelo de clasificador SVM entrenado.
Descripción	Se hace uso de dos funciones que permiten clasificar los datos de manera supervisada y validar el modelo del clasificador mediante el valor de la precisión.
Lenguaje	Python (uso de librería sickit learn)
Pseudocodigo	1. INICIO 2. cargar arreglo de datos (diferentes casos) 3. definir C y gamma (mapa de calor) 4. PARA i=1 hasta 10 5. Definir función kernel (lineal, polinomial, RBF) 6. Llamar función SVM con C, gamma y kernel 7. PARA j=1 hasta K 8. Entrenar modelo SVM (80% entrenamiento, 20% prueba) 9. Almacenar valor precisión [i][j] = precisión SVM 10. FIN PARA 11. FIN PARA

Fuente: Elaborada por los autores

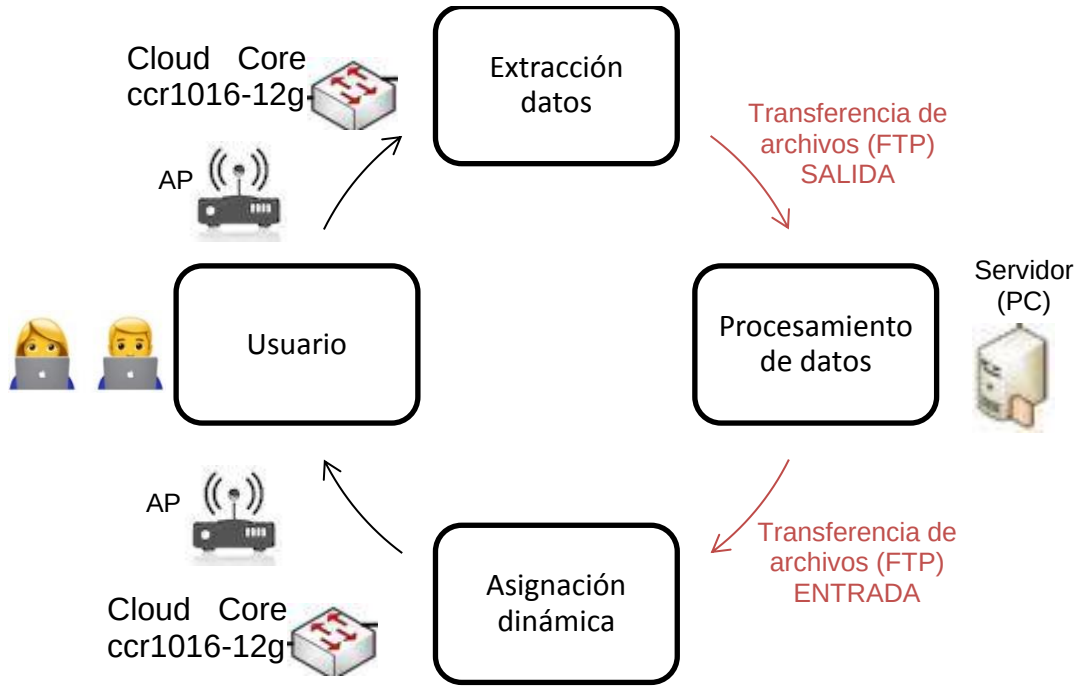
1.3. Asignación dinámica de ancho de banda

Una vez la etapa de procesamiento de datos se desarrolló satisfactoriamente, se procedió a poner en marcha el sistema de asignación de ancho de banda dinámico.

Como se observa en la Figura 26, este proceso corresponde a un ciclo que se desarrolló en tiempo real, para una mejor comprensión a continuación se describen todos los procesos desarrollados en cada una de las etapas.

Es importante destacar que las tareas de extracción de datos y asignación dinámica se llevaron a cabo en el equipo de red Cloud Core, mientras que el procesamiento de datos se realizó en un equipo externo que contó con funcionalidades para aplicar las técnicas de machine learning.

Figura 26. Sistema de asignación dinámica de ancho de banda



Fuente: Diagrama que ilustra el funcionamiento del sistema elaborado por los autores.

1.3.1 Extracción de datos

Esta fase no supuso mayor reto, ya que fácilmente la tarea fue adaptada al código previamente desarrollado para la construcción de la base de datos. Puesto que la asignación dinámica se realizó teniendo en cuenta la dirección IP y una etiqueta que definió un perfil de navegación proveniente de la etapa de procesamiento de datos; se extrajeron los siguientes campos para cada usuario activo:

- Dirección IP
- Bytes-In
- Bytes-Out
- Tiempo de conexión

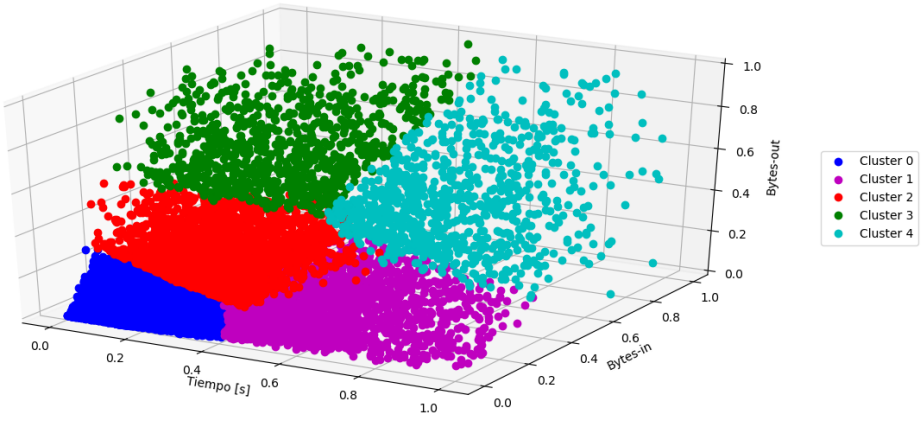
Donde Bytes-In, Bytes-Out y tiempo de conexión fueron las características de entrada al modelo de clasificación propuesto. Estos datos fueron transferidos en formato de texto plano mediante el protocolo de transferencia de archivos FTP en un archivo denominado *"activos.txt"*, el destino fue un directorio del equipo que realizó la etapa de procesamiento de datos. En esta investigación fue utilizado un computador portátil con sistema operativo Windows 7, procesador Intel inside CORE-i5 y memoria RAM de 4 GB, que contó con el administrador de Internet Information Services (IIS), para la creación de un sitio FTP.

1.3.2 Procesamiento de datos

Esta etapa fue descrita paso a paso en la sección anterior desde el punto de vista de machine learning, sin embargo, en este punto se destacó los aspectos técnicos para la aplicación de la asignación dinámica. Como antes fue mencionado, se seleccionó un modelo de SVM polinomial de grado 2, que presentó etapas de eliminación de valores atípicos y normalización; debido a que esto produce un modelo dinámico en cuanto a la asignación de ancho de banda, que permitió notar cambios de perfiles de navegación en cortos periodos de tiempo y consumos de navegación no tan elevados.

La Figura 27, muestra los 5 clusters que debían ser clasificados por la SVM para la asignación de perfil de navegación. Cabe resaltar que no fue indicada una imagen que muestre la clasificación visual de SVM debido a la dimensión del espacio de entrada. Sin embargo, en el Anexo 2, se encuentra un artículo en el cual se realizó el estudio en un espacio bidimensional y se observa imágenes que contienen los límites de clasificación visuales. Python adquiere la información que posteriormente procesó, del archivo *"activos.txt"* en el directorio del sitio FTP.

Figura 27. Tipos de clusters correspondientes a los perfiles de navegación.



Fuente: Gráfica de grupos de usuarios elaborada por los autores en Python.

De la Figura 27 se observan los siguientes comportamientos:

- Cluster 0: Usuarios que se conectaron poco tiempo y tuvieron poco consumo de volumen de carga y descarga.
- Cluster 1: Usuarios que pasado un tiempo considerable, tuvieron un consumo de descarga bajo.
- Cluster 2: Usuarios que tienen un consumo medio en tiempo de conexión, volumen de carga y descarga.
- Cluster 3: Usuarios que en poco tiempo tuvieron volúmenes de carga y descarga bastante elevados.
- Cluster 4: Usuarios que en un tiempo de conexión mayor tuvieron un consumo elevado en tiempo de conexión, volumen de carga y descarga.

Estos comportamientos fueron tomados en cuenta a la hora de conformar los perfiles de usuarios, para la asignación dinámica.

Después de haber realizado el procesamiento de datos en el equipo con el modelo de SVM seleccionado, estos fueron transferidos en formato de texto plano mediante el protocolo FTP en un archivo denominado *"clasificados.txt"*, cuyo destino fue el directorio de archivos del equipo de red Cloud Core. El contenido del archivo incorporó dos campos, la dirección IP de los usuarios activos y la etiqueta de perfil de usuario producto de la clasificación. Una vez esta información fue encontrada en el equipo de red, se procedió a realizar la asignación de ancho de banda.

En la Tabla 18 se consigna la información más importante del script en Python, que desarrolla la tarea de clasificación.

Tabla 18 Información del script 7

Nombre del Script	Clasificación de datos de usuarios activos (tiempo real)
Parámetros de entrada	Address, Bytes In, Bytes Out, Uptime
Parámetros de salida	Variables de etiquetas de grupos.
Descripción	Se carga el modelo entrenado y se clasifica usuarios los usuarios activos, entregando como parámetro de salida las direcciones IP y la etiqueta asociada a un perfil de navegación
Lenguaje	Python (uso de librería sickit learn)
Pseudocodigo	1. INICIO 2. MENTRAS (verdadero) 3. Inicializar variables 4. Abrir directorio de servidor (sitio donde se guarda "activos.txt") 5. cargar Modelo de clasificador SVM entrenado. 6. Clasificar IP en perfiles de navegación según Modelo SVM 7. Almacenar variables en archivo "clasificados.txt" 8. Guardar archivo "clasificados.txt" en directorio del servidor 9. FIN MIENTRAS

Fuente: Elaborada por los autores.

1.3.3. Asignación de ancho de banda

Como se mencionó anteriormente, esta etapa se desarrolló en el equipo de red, Cloud Core. Utilizando la herramienta "queue" (cola), incorporada en RouterOS.

- **Queue:** Las colas en aplicaciones de redes son utilizadas para limitar y priorizar el tráfico, algunas de las funciones se listan en seguida:

- Limitar la velocidad de transmisión de datos para ciertas direcciones IP, subredes, protocolos, puertos, entre otros parámetros.
- Priorizar algunos flujos de paquetes sobre otros.
- Compartir el tráfico disponible entre usuarios por igual o dependiendo de la carga del canal

RouterOS permitió configurar los encolamientos de dos maneras diferentes:

- Menú de cola simple (Queue simple menu): diseñado para facilitar la configuración de simples tareas de encolamiento, por ejemplo, la limitación de carga y descarga de un cliente.
- Menú de árbol de colas (Queue tree menu): diseñado para implementar tareas de encolamiento avanzadas, tales como políticas globales de priorización, limitaciones a grupos de usuarios. Se requiere que previamente el flujo de los paquetes sean marcados en el menú */ip firewall mangle*

En la presente investigación se hizo uso del menú de colas simple de una manera creativa, con el fin de implementar una tarea de encolamiento a diferentes grupos de usuarios provenientes de la clasificación de datos.

Continuando con la descripción del proceso, la información del archivo *"clasificados.txt"* se utilizó y se acopló de manera que el equipo de red la comprenda. Es importante mencionar que esto supuso un gran reto, ya que el archivo se tuvo que discernir en variables que el equipo sea capaz de manejar. Esto se desarrolló en un script dentro de RouterOS, que permitió realizar la agrupación de direcciones IP, según la etiqueta que provino de la etapa de procesamiento de datos. En la Tabla 19, se presenta los perfiles de usuarios y la distribución realizada para la asignación de ancho de banda, donde dichos perfiles reciben un porcentaje del total de ancho de banda disponible.

Tabla 19 perfiles de usuarios

Clúster	Perfil de usuario	Límite de carga	Límite de descarga
0	TIPO 0	10%	10%
1	TIPO 1	15%	10%
2	TIPO 2	15%	20%
3	TIPO 3	25%	25%
4	TIPO 4	35%	35%

Fuente: Elaborada por los autores.

En la Tabla 20 se muestra la información del script que desarrolló las tareas de las etapas de extracción de datos y asignación dinámica llevadas a cabo.

Tabla 20 Información del script 6

Nombre del Script	Extracción de datos y Asignación de ancho de banda.
Parámetros de entrada	Número de usuarios activos: Num, Variables de etiquetas de grupos: IP Clasificadas.
Parámetros de salida	Asignación de ancho de banda
Descripción	Permite extraer los campos: Address (IP), Bytes In, Bytes Out, Uptime para cada usuario activo en un instante de tiempo con intervalos de 5 minutos y transferir archivos mediante FTP a un servidor, luego recibir las variables de etiquetas de grupos del mismo servidor (equipo de procesamiento de datos) para una posterior asignación de ancho de banda por perfiles mediante Queue
Lenguaje	Mikrotik Scripting Language
Pseudocodigo	1. INICIO 2. MIENTRAS (verdadero) 3. Crear archivo1 “activos.txt” 4. Inicializar variable: Num (número de usuarios conectados) 5. PARA i=1 hasta Num 6. escribir en archivo1 = [Address, Bytes In, Bytes Out, Uptime] 7. FIN PARA 8. transferencia archivo1 vía FTP (Enviar) 9. retardo 30 segundos (Mientras el equipo externo procesa datos) 10. transferencia archivo2 vía FTP (Recibir) 11. Leer archivo2 “clasificados.txt” 12. Inicializar variables: perfiles de navegación 13. No. registros archivo2 14. PARA i=1 hasta longitud No. registros archivo 2 15. Asignar a perfiles de navegación = IP clasificadas 16. FIN PARA 17. Asignar ancho de banda a perfiles de navegación 18. Retardo 5 minutos 19. FIN MIENTRAS

Fuente: Elaborada por los autores.

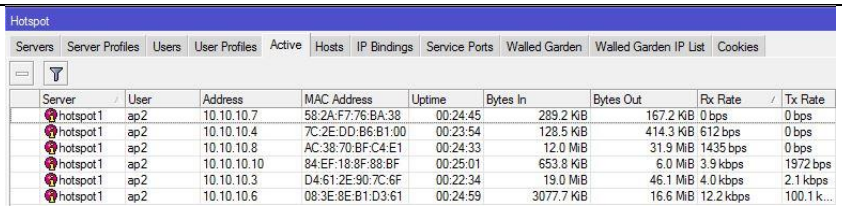
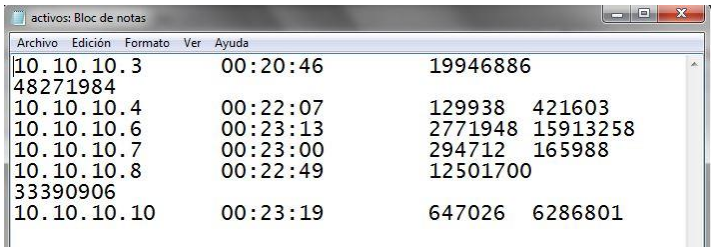
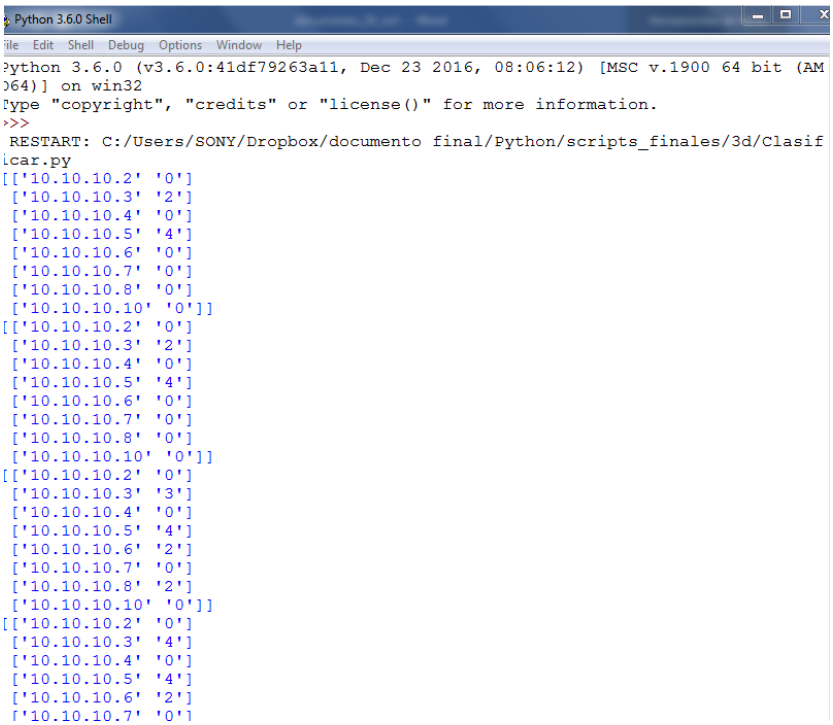
1.3.4. Usuarios

Los usuarios representaron un punto vital del sistema, ya que los datos de navegación provienen de estos, y de esta manera el sistema se ejecutó en lazo cerrado.

Cabe resaltar que este sistema pudo ser implementado en la red inalámbrica de la Universidad de Nariño, pero por motivos de políticas y administración ajenos al trabajo, no se puso en marcha en esta organización. De ahí que este apartado quiere dar a conocer, los resultados obtenidos en una red de prueba, logrando que el sistema trabaje en cualquier entorno.

En la Tabla 21 se consignan las imágenes que representan el funcionamiento del sistema en la red de prueba, dicho sistema fue iterativo y realizó la asignación dinámica cada 3 minutos. En la etapa tres, se indica la transición de las clasificaciones para 6 periodos de tiempo. Sin embargo, se tomó el ultimo intervalo de tiempo para una mayor comprensión del funcionamiento. En la primera etapa se observan los usuarios activos, con sus correspondientes consumos. En segundo lugar, se observa el contenido del archivo de salida del Cloud Core *"activos.txt"* enviado mediante protocolo FTP, para posteriormente obtener la clasificación de la información en el equipo de cómputo como es observado en la etapa No. 3. Luego el archivo *"clasificados.txt"* fue transferido al directorio de Cloud Core. El proceso se continuó realizando la asignación de ancho de banda en cinco perfiles de usuario (ip0, ip1, ip2, ip3, ip4) mediante la herramienta simple "queue". Finalmente se muestra el resultado de un test de velocidad que comprobó el ancho de banda asignado para un usuario.

Tabla 21 Resultados por etapas del funcionamiento del sistema

Etapa No.	Representación gráfica	Descripción
1		Usuarios activos en la red de prueba (Hotspot)
2		Archivo de datos (“activos.txt”)
3		Clasificación de direcciones IP (Script Python)

	['10.10.10.8' '3'] ['10.10.10.10' '0']] [['10.10.10.2' '0'] ['10.10.10.3' '4'] ['10.10.10.4' '0'] ['10.10.10.5' '4'] ['10.10.10.6' '2'] ['10.10.10.7' '0'] ['10.10.10.8' '3'] ['10.10.10.10' '0']] [['10.10.10.3' '4'] ['10.10.10.4' '0'] ['10.10.10.6' '2'] ['10.10.10.7' '0'] ['10.10.10.8' '3'] ['10.10.10.10' '0']]	
4	clasificados: Bloc de notas Archivo Edición Formato Ver Ayuda 10.10.10.3 4 10.10.10.4 0 10.10.10.6 2 10.10.10.7 0 10.10.10.8 3 10.10.10.10 0	Archivo de datos (“Clasificados.txt”)
5	Queue List Simple Queues | Interface Queues | Queue Tree | Queue Types Reset Counters | Reset All Counters | Find # | Name | Target | Upload Max Limit | Download Max Limit | Packet Marks 0 | ip0 | 10.10.10.4, 10.10.10.7, 10.10.10.10 | 200k | 1M 1 | ip1 | 0.0.0.0/0 | 300k | 1M 2 | ip2 | 10.10.10.6 | 300k | 2M 3 | ip3 | 10.10.10.8 | 500k | 2500k 4 | ip4 | 10.10.10.3 | 700k | 3500k	Asignación de ancho de banda por perfiles de usuarios (Queues)
6	BAJADA 0,93 Mbps Datos utilizados 0,88 MB SUBIDA 0,12 Mbps Datos utilizados 0,11 MB PING 64 ms JITTER 3 ms PÉRDIDA – % Level 3 Bogota Claro "AP2" Ubicación del usuario Lat: 1,2147 Lon: -77,2950 IP interna: 10.10.10.7 IP externa: 190.143.22.85 Resultados	Test de prueba de velocidad para un ususario

Fuente: Elaborada por los autores

2. GUÍA DE CONFIGURACIÓN PARA ADMINISTRADOR

Para poner en funcionamiento el sistema se debió realizar la respectiva configuración de los dos equipos principales, equipo que realizó la clasificación y equipo que realizó la asignación.

2.1. Configuración de equipos

Para acceder a las herramientas que mikrotik proporcionó se debió contar con una interfaz gráfica que permitió administrar sus dispositivos, para este caso RouterOS contó con una interfaz de usuario web o el uso de un programa ejecutable Winbox, cabe resaltar que el proceso se tuvo en cuenta después de que el equipo se configuro como enrutador.

Por otra parte como se mencionó en las secciones previas, el procesamiento de los datos se realizó con el software de programación Python, por consiguiente el equipo en el que se realizó dicho procesamiento contó con un compilador que permitió ejecutar el script. Así mismo también debió contar con un programa que posibilitó la transferencia de archivos mediante el protocolo FTP, el IIS de Windows fue utilizado en esta investigación. En la Tabla 22 se presentan las principales configuraciones de los dos equipos, así mismo, en el Anexo 4, se encuentran los pasos que se deben seguir para la configuración de los equipos de la marca Mikrotik.

Tabla 22 Configuración de equipos de red y de cómputo

Herramienta	Descripción	Configuración
Hotspot (equipo de red)	Proporcionar autenticación a los clientes para acceder a la navegación	En el menú: IP >> Hotspot >> Hotspot Setup Seleccionar Interfaz para aplicar Hotspot Seleccionar Pool (rango) de direcciones IP Configurar usuario y contraseña para el Hotspot
Queus (equipo de red)	Herramienta utilizada para configurar los perfiles de navegación	En el menú: Queus >> Simple Queues >> + (Nuevo Queu) Crear tantos Queus como perfiles de usuarios se deseen manejar (e.e.t*: 5), posteriormente identificarlos con nombres. New Simple Queue >> Max. Limit Seleccionar los límites de tráfico de carga y descarga máximos.
Script (equipo de red)	Creación de un script en la interfaz de mikrotik	En el menú: System >> Scripts >> +(Nuevo Script)

		Identificar con un nombre (e.e.t*: "*Extracción_Asignación*") New Script >> Source Cargar líneas de código del script denominado "*Extracción_Asignación*" Nota: La Dirección IP para realizar la transferencia de archivos debe ser la del equipo de computo
Scheduler (equipo de red)	Ejecuta el script a partir del momento determinado, para intervalos de tiempo específicos	En el menú: System >> Scheduler >> +(Nuevo Schedule) Identificar con un nombre (e.e.t*: "*Correr script*") Configurar hora y fecha de inicio, además de el intervalo de tiempo (e.e.t*: 3 minutos). New Schedule >> On Event Escribir el nombre del Script (e.e.t*: *"Extracción_Asignación"*) Nota: La hora y fecha del equipo se deben estar configurar de manera anticipada.
IIS (equipo de cómputo)	Se hace uso del protocolo FTP para la transferencia de los archivos	En el menú: Conexiones >> sitios >> agregar sitio FTP Información del sitio: Nombre del sitio FTP>> Nombre Ruta de acceso física>> Ubicar la ruta de acceso física (donde se alojan los archivos) Configuración de enlaces y SSL: Dirección IP >> Configurar Dirección IP (equipo de cómputo) y Puerto Información de autenticación y autorización: Autorización >> Permitir el acceso a >> Todos los usuarios Permisos >> Leer y Escribir
e.e.t* : En este trabajo		

Fuente: Elaborada por los autores

2.2. Ejecución del sistema

En la Tabla 23 se indica los scripts que debieron ser ejecutados, para poner en marcha el sistema, fue necesario mencionar que se debieron correr los scripts por separado, ejecutando en primer lugar el script denominado "***Extracción_Asignación***" en mikrotik, posteriormente se corrió el script "*Clasificar*" en python, en un periodo de tiempo no superior a 30 segundos.

Tabla 23 Ejecución del sistema

Script	Descripción	Compilación
"Extracción_Asignación"	Script encargado de automatizar las tareas de extracción y asignación. Extrae la información de los usuarios activos (activos.txt) Además, también se encarga del procesamiento y la asignación de ancho de banda de los datos provenientes de la clasificación ("clasificados.txt")	En el menu: System>> Scheduler>> ejecutar "*correr script*"
"*Clasificar*"	Permite la ejecución permanente del script que realiza la clasificación de los datos del archivo "activos.txt", en el fichero denominado "clasificados.txt"	En el compilador ejecutar el script denominado "*clasificar*" NOTA: es necesario tener en cuenta el directorio donde se encuentran alojados los archivos mediante el protocolo FTP. Además también se debe contar con el modelo de la SVM entrenado (e.e.t*: Lineal) ubicado en la carpeta donde se encuentra el script "*clasificar*"
e.e.t* : En este trabajo		

Fuente: Elaborada por los autores.

3. CONCLUSIONES

Los valores de precisión obtenidos en los modelos de SVM permitieron concluir que se realizó una correcta calibración de los parámetros $\boldsymbol{C}$ y $\boldsymbol{\gamma}$ con la técnica de búsqueda de cuadrícula propuesta.

Los mejores resultados para la aplicación del sistema propuesto se presentaron cuando los valores atípicos fueron removidos.

El valor de $k = 5$ en la técnica de validación cruzada de k iteraciones presente en la literatura, permitió validar el funcionamiento correcto del sistema con valores de precisión altos.

Las evidencias presentadas de la ejecución del sistema en la red de prueba y los valores de precisión obtenidos, corroboraron que los procesos llevados a cabo en cada una de las etapas se desarrollaron correctamente.

4. RECOMENDACIONES

Se originaron las posibilidades de realizar los siguientes trabajos futuros: la implementación y prueba de los modelos en sedes de la universidad y/o empresas y la adaptación del sistema desarrollando el proceso de entrenamiento a la vez que se realiza la asignación de ancho de banda dinámico.

Explorar distintas técnicas de inteligencia artificial y aprendizaje profundo, siendo la última de gran interés debido a la gran cantidad de muestras que fueron obtenidas en la base de datos.

Realizar modelos de predicción de consumo basados en machine learning o deep learning, que analicen el tipo de tráfico de navegación de los usuarios.

Desarrollar un sistema de asignación dinámico que tenga en cuenta los datos de consumo instantáneos a diferencia de datos acumulados como se realiza dentro de la investigación.

BIBLIOGRAFÍA

About us. Mikrotik. [en linea]. Disponible en Internet: https://mikrotik.com/aboutus.

BOSER, Bernhard E; ISABELLE M Guyon y Vladimir N Vapnik. A training algorithm for optimal margin classifiers. Fifth ACM Workshop on Computational Learning Theory (COLT). New York, 1992.

BREIMAN, Leo. Statistical modeling: The two cultures (with discussion). En Statistical science. 199p.

CESTERO, Eloy Vicente y CABALLERO, Alfonso Mateos. Data Science y redes complejas: métodos y aplicaciones. Editorial universitaria ramon areces, 2018.

Comportamiento del Tráfico NAP Colombia. NAP COLOMBIA. 2018. [en línea] Disponible en internet: nap.co.

COVER, Thomas. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. En IEEE Transactions on Electronic Computers 14, 1964. 326p.

CRISTIANINI, Nello, y TAYLOR, John Shawe. An Introduction to Support Vector Machines. Cambridge Univ. Press, 2000.

de España, Telefónica. Introducción a la Telemática y a las redes de datos Telefónica: Madrid, 2000.

Editor RFC. Protocolo TCP/IP 05. 20 de Febrero de 2018. [en línea] Disponible en internet: http://www.mheducation.es/bcv/guide/capitulo/8448199766.pdf.

FUKUNAGA, Keinosuke. Statistical Pattern Recognition. World Scientific. 1999.

GOVE, Robert. Using the elbow method to determine the optimal number of clusters for k-means clustering. Robert Gove's Block. 26 de December 2017. [En línea]. Disponible en internet: https://bl.ocks.org/rpgove/0060ff3b656618e9136b.

GUISANDE, Gonzáles; CÁSTOR, Antonio, y BARREIRO, Aldo. Tratamiento de Datos con R, Statistica y SPSS. Diaz de Santos, 2013.

HETAL, Bhavsar; AMIT, Ganatra. Increasing Efficiency of Support Vector Machine using the Novel Kernel Function: Combination of Polynomial and Radial Basis Function. 2014. 2319p.

HSU, Chih-Wei; CHANG, Chih-Chung, y LIN, Chih.Jen. A practical guide to support vector classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University. Taiwan, 2003.

Internet Live Stats. Internet Users [En linea]. Marzo 2018. Disponible en internet: http://www.internetlivestats.com/.

GARETH, James; WITTEN, Daniela y HASTIE Trevor. An Introduction to Statistical Learning. New York: Springer, 2013.

LIU, Xinmei; XIAOKAI, Wan; LI, Wan y YAN, Han. Reliability Prediction of LAN/WLAN Integration Network Based on Artificial Intelligence. International Conference on Computer Application and System Modeling ICCASM. Taiyuan, 2010.

MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc 415p.

MOGUERZA, Javier, MUÑOZ, Alberto. 2006. Support vector machines with applications. Statistical Science. 322p.

MONTGOMERY, Douglas. Diseño y análisis de experimentos. Limusa Wiley, 2005.

NILSSON, Nils. Principles of artificial Intelligence. Morgan Kaufmann, 2014.

PEDREGOSA, F; VAROQUAUX, G; GRAMFORT, A; MICHE ,I; THIRION, B; GRISEL, O; BLONDEL, M y otros. Sickit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011. 2825p.

RAMIREZ, Juan Carlos y DE AGUAS Johan Manuel. Escalafón de la competitividad de los departamentos de Colombia. CEPAL, 2015.

Sickit Learn. Sickit Learn: RBF SVM parameters. [en linea]. Disponible en internet: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

SOLANO, Humberto Llinás y ROJAS, Carlos. Estadística Descriptiva y Distribuciones de Probabilidad. Universidad del Norte, 2005.

TAN, Pang Ning; STEINBACH, Michael; KARPATNE, Anuj y VIPIN, Kumar. Introduction to Data Mining. Pearson, 2013.

TAY, Francis y LIJUAN, Cao. Application of support vector machines in financial time series forecasting, 2001. 309p.

VAPNIK, Vladimir; CHERVONENKIS, Alexey. A note on a class of perceptrons.» Automat. Remote Control 25, 1964.103p.

WALPOLE, Ronald; MYERS, Raymond y MYERS, Sharon. Probabilidad y estadística para ingenieros. México: Pearson, 1999.

WETHERALL, Tanenbawm. Redes de Computadoras. México: Pearson Educación, 2012.

ANEXOS

ANEXO A. Certificado de participación en el 3er Congreso Andino en Computación, Informática y Educación

Este anexo contiene el poster seleccionado para la sustentación en el "3er congreso ANDINO en computación, informática y educacion" desarrollado en Pasto-Nariño, durante los días 1,2 y 3 de noviembre de 2017.

Certificado de poster: "Classification of Hosts in a WLAN with a system based on Support Vector Machine and a Neural Network classifier." CACIED 2017

Poster: Classification of Hosts in a WLAN with a system based on Support Vector Machine and a Neural Network classifier.

Classification of Hosts in a WLAN with a system based on Support Vector Machine and a Neural Network classifier

Andrea Johana Chaves, Oscar Javier Jossa, Mario Fernando Jojoa
Ingeniería Electrónica, Universidad de Nariño.

RESUMEN

En la administración de una red de datos interesa conocer cuál es el comportamiento de un grupo de usuarios respecto a tiempos de conexión y consumo de anchos de banda. Esta información ayuda en la planificación de la red y en la toma de decisiones vitales para el crecimiento de la misma.

En la presente investigación se pretenden desarrollar dos estrategias de clasificación, una red neuronal artificial (ANN, Artificial Neural Network) tipo perceptrón y una maquina de soporte vectorial (SVM, Suport Vector Machine). Estos clasificadores permitirán identificar el tipo de equipo: Celulares, Tabletas, Portátiles y Computadores de Escritorio que utilizan los clientes de una red inalámbrica de área local (WLAN, Wireless Local Area Network) de la Universidad de Nariño, teniendo en cuenta dos características, tiempo de conexión y volumen de descarga. Se desarrollarán las estrategias de clasificación en 3 etapas: entrenamiento, clasificación y evaluación de error.

OBJETIVOS

OBJETIVO GENERAL

Implementar dos estrategias de clasificación, que permitan identificar el tipo de equipo utilizado por el cliente de una red WLAN, según su tiempo de conexión y volumen de descarga.

OBJETIVOS ESPECIFICOS

- ✓ Obtener una base de datos del repositorio del sistema RADIUS de prueba de la universidad de Nariño, con los tiempos de conexión y el volumen de descarga, de los dispositivos utilizados por los clientes de una red WLAN.
- ✓ Implementar dos algoritmos de clasificación: SVM y red neuronal artificial tipo perceptrón.
- ✓ Evaluar los algoritmos implementados con la base de datos, según la tasa de error, tasa de acierto, sensibilidad y especificidad.

METODOLOGIA

CARACTERISTICAS DE LOS DATOS USADOS

Como base se tomaron datos del repositorio del sistema RADIUS de prueba en la universidad de Nariño. Los datos de cada clase corresponden al Tiempo de conexión [min] y volumen de descarga [Megabytes], de cuatro tipos de equipos con conexión a la red WiFi: Celulares, Tabletas, Portátiles y Computadores de Escritorio. La ventana de observación fue de 240 minutos comprendidos entre las 9:00 am y 1:00 pm durante 5 días.

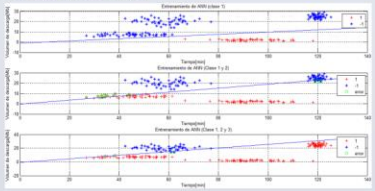
IMPLEMENTACIÓN DE LAS ESTRATEGIAS DE CLASIFICACIÓN

Debido a que la distribución de los vectores de características según las 4 clases es diferenciable entre ellas, se opto por utilizar una red neuronal tipo perceptrón de una neurona, y una maquina de soporte vectorial con kernel lineal. La implementación de los dos clasificadores, se desarrolló mediante la herramienta de software matemático MATLAB. El entrenamiento se realizó con 50 unidades por clase o tipo de equipo y de manera segmentada para cada clase con el fin de obtener el menor error [1].

I. Red Neuronal Artificial (ANN)

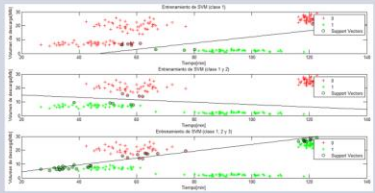
Es un procesador paralelo distribuido constituido por unidades simples de procesamiento que tienen una disposición natural, para almacenamiento de conocimiento experimental. Un tipo de red neuronal es el perceptrón, es esencialmente de una neurona con pesos y bias ajustables, puede separar datos que sean clasificables de forma lineal con una sola neurona [2].

En la siguiente figura se presentan los resultados del entrenamiento de la ANN, según las clases o tipos.



II. Maquina de Soporte Vectorial (SVM)

Su funcionamiento se basa, en una estimación de las funciones de distribución de probabilidad, en la configuración de hiperplanos en el aprendizaje lineal de las máquinas, además su configuración es única [3]. En el aprendizaje supervisado de la máquina se usan algoritmos que buscan hiperplanos que separan las clases de manera óptima de tal forma que se maximice su distancia [4]. La función Kernel (núcleo) más simple es el uso de líneas rectas, planos rectos o hiperplanos N-dimensionales para lograr la separación. A continuación se presentan los resultados obtenidos en el entrenamiento de la SVM, con un Kernel lineal.



RESULTADOS ESPERADOS

Para validar el entrenamiento de los clasificadores, se procede a probar su funcionamiento con valores indicados en la siguiente gráfica, de esta manera se muestra la distribución de los 800 datos o equipos con conexión WiFi, según cada clase, con los cuales se puso a prueba los clasificadores.

Los resultados que se obtuvieron en la clasificación de los 800 datos para los dos clasificadores se presentan a continuación:

MEDIDAS DE RENDIMIENTO

Para presentar los resultados de los clasificadores se procede a estimar los siguientes parámetros:

- ✓ Tasa de acierto o eficiencia (CCR, Correct Clasification Rate): Proporción de patrones correctamente clasificados por el sistema.
- ✓ Tasa de error (ER, Error Rate): Proporción de patrones erradamente clasificados.
- ✓ Sensibilidad (S): Da una indicación de la capacidad del sistema para detectar los patrones de la clase de referencia.
- ✓ Especificidad (E): Da una indicación de la capacidad del sistema para rechazar los patrones que no pertenecen a la clase de referencia.

Los resultados son consignados en la siguiente tabla:

	MEC	CCR	ER	S	E
ANN	SEC	94,38%	5,63%	94,94%	93,83%
SVM	SEC	98,75%	1,25%	99,49%	98,03%

MEC: Método de extracción de Características
SEC: Sin extracción de características

CONCLUSIONES

La distribución de los vectores de características según las cuatro clases, son fácilmente diferenciables, por lo que la tarea de identificación y clasificación no reviste mayor complejidad. Esto explica los valores de error de clasificación bajos para los dos tipos de clasificadores.

De los dos clasificadores SVM y ANN, el primero resulta con menor tasa de error; mayor tasa de acierto, sensibilidad y especificidad, estos valores se acercan a los resultados ideales de clasificación.

Para aplicaciones prácticas, por eficiencia, se prefiere implementar un clasificador de baja complejidad computacional pero igualmente eficaz, lo cual apunta al clasificador SVM como la mejor opción de los dos.

Debido a que en el desarrollo del trabajo, no se realizó la etapa de extracción de características, una posible mejora para la clasificación de los datos, seria implementar esta etapa que permita mejorar la eficiencia de los clasificadores.

En la administración de una red de datos se requiere conocer además del número de clientes conectados, el tipo de dispositivo y el volumen de descarga demandada; parámetros con los cuales se podría optimizar el uso de los recursos de una organización.

REFERENCIAS

[1]. Simon O. Haykin , Neural Networks: A Comprehensive Foundation, Second Edition, Ed. Pearson Education Press, p. 178-270, 2001.

[2]. Holmstrom, L., Koistinen, P., Laaksonen, J., Oja, "E. Neural and statistical classifiers taxonomy and two case studies" IEEE Trans. Neural Networks, 8, 5–17, 1997.

[3] Romo, Harold Probabilidad y procesos estocásticos, First Edition, Ed EAE 2011.

[4] Fukunaga, Keinosuke, Introduction to statical pattern recognition, Second Edition, Ed.Academic Press, p. 20-200, 1990.

CONTACTO

Andrea Johana Chaves, e-mail: andreajohanacv@gmail.com
Oscar Javier Jossa, e-mail: oscaarjte@gmail.com
Mario Fernando Jojoa, e-mail: mario@udenar.edu.co

ANEXO B. Certificado de participación en la XI Conferencia Científica de Telecomunicaciones, Tecnologías de la Información y Comunicaciones

Este anexo contiene el articulo seleccionado para la sustentación en modalidad de ponencia en la "XI Conferencia Científica de Telecomunicaciones, Tecnologías de la Información y Comunicaciones," desarrollado en Quito-Ecuador, durante los días 20,21 y 22 de noviembre de 2017.

Artículo: Classification de Hosts en una red WLAN usando un sistema basado en clasificadores tipo Máquina de Soporte Vectorial y Red Neuronal Artificial.

Clasificación de Hosts en una red WLAN usando un sistema basado en clasificadores tipo Maquina de Soporte Vectorial y Red Neuronal Artificial.

Andrea Johana Chaves Villota, Oscar Javier Jossa Bastidas, Mario Fernando Jojoa Acosta.
Universidad de Nariño
andreajohanacv@gmail.com,oscaarjte@gmail.com,mario@udenar.edu.co

Abstract

This paper presents a strategy for the classification of clients in a network based on two characteristics, time of connection and bytes downloaded. This information could be used as indicators that allow take decisions in subjects of optimization by the network administrator.

1. Introducción

En la administración de una red de datos interesa conocer cuál es el comportamiento de un grupo de usuarios respecto a tiempos de conexión y consumo de anchos de banda. Esta información ayuda en la planificación de la red y en la toma de decisiones vitales para el crecimiento de la misma. Los sistemas de gestión automatizados como el portal cautivo ayudan a tener estadísticas, pero es necesaria una clasificación de estos datos para la observación de sus patrones estadísticos como una herramienta fundamental en el análisis de comportamiento. Un software muy difundido es RADIUS de licencia GPL, que usa un motor MYSQL como repositorio de esta información.

En la presente investigación se pretenden desarrollar dos estrategias de clasificación, una red neuronal artificial (ANN, Artificial Neural Network) tipo perceptrón multicapa y una máquina de soporte vectorial (SVM, Suport Vector Machine). Estos clasificadores permitirán identificar el tipo de equipo: Celulares, Tabletas, Portátiles y Computadores de Escritorio que utilizan los clientes de una red inalámbrica de área local (WLAN, Wireless Local Area Network) de la Universidad de Nariño, teniendo en cuenta dos características, tiempo de conexión y volumen de descarga.

2. El clasificador SVM (Support Vector Machine)

Este tipo de clasificador basa su funcionamiento en las siguientes características [1]:

- ✓ Hace una estimación de las funciones de distribución de probabilidad.
- ✓ Configura hiperplanos en el aprendizaje lineal de las máquinas.
- ✓ Su configuración es única.

En la figura 1, se muestra un problema de clasificación de dos clases:

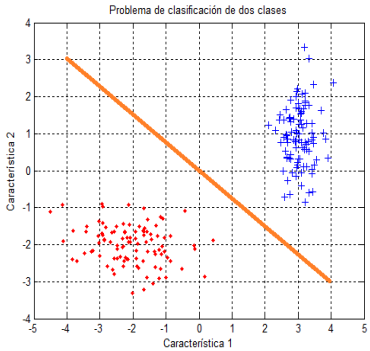


Figura 1. Problema de clasificación de dos clases. Fuente propia.

En el aprendizaje supervisado de la máquina se usan algoritmos que buscan hiperplanos que separan las clases de manera óptima de tal forma que se maximice su distancia [2]. La función Kernel (núcleo) más simple es el uso de líneas rectas, planos rectos o hiperplanos N-dimensionales para lograr la separación. Existen funciones núcleo más sofisticadas como polinomiales, perceptrón, radiales gaussianas y sigmoideas dependiendo del problema.

3. El clasificador basado en Redes Neuronales Artificiales

La estructura de redes neuronales conocida como perceptrón multicapa, está constituida de una capa de entrada, una o más capas ocultas y una capa de salida. La señal de entrada se propaga a través de la red hacia adelante (forward) capa por capa. Esta estructura ha sido aplicada satisfactoriamente con el algoritmo de Backpropagation, que es basado en el aprendizaje de corrección de errores.

Un perceptrón multicapa tiene tres características distintivas [3]:

- ✓ Una función de activación no lineal suave (i.e., continuamente diferenciable) es incluida en el modelo de cada neurona de la red.
- ✓ La red contiene una o más capas ocultas que no hacen parte de la entrada o salida de la red.
- ✓ La red tiene un alto grado de conectividad determinado por la sinapsis de la red.

En la figura 2 se indica la arquitectura de un perceptrón multicapa con dos capas ocultas, una capa de entrada y una de salida, se puede apreciar que la red está conectada completamente, esto significa que una neurona en cualquier capa está conectada con todas las neuronas de la capa previa.

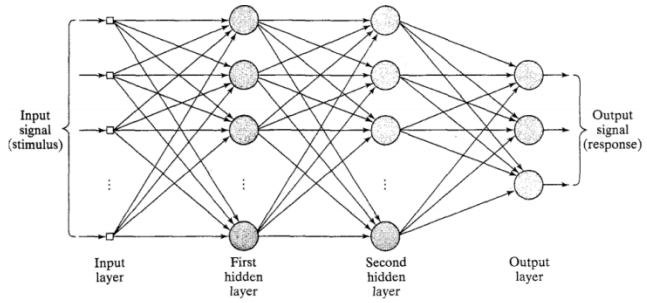


Figura 2. Arquitectura grafica de un perceptrón multicapa con dos capas ocultas. Fuente: [4]

4. Características de los datos usados

Como base se tomaron datos del repositorio del sistema RADIUS de prueba en la universidad de Nariño. Los datos de cada clase corresponden al Tiempo de conexión [min] y volumen de descarga [Megabytes], de cuatro tipos de equipos con conexión a la red WiFi: Celulares, Tabletas, Portátiles y Computadores de Escritorio. La ventana de observación fue de 240 minutos comprendidos entre las 9:00 am y 1:00 pm durante 5 días, con un flujo promedio por clase de 40 equipos. En total se observaron 50 unidades por clase o tipo de equipo para el entrenamiento y 800 unidades para la prueba de los clasificadores.

5. Resultados

5.1. Entrenamiento de Clasificadores

5.1.1. Clasificador SVM

El entrenamiento se hace de manera segmentada para cada clase con el fin de obtener el menor error [5]. A continuación se describe las tareas realizadas. El primer paso corresponde al entrenamiento para identificación de la clase celulares de los demás equipos. En la figura 2, se muestra la recta que separa la clase Celulares de los demás equipos, representados los aciertos con (*) y valor numérico 1 y los desaciertos con (+) y valor numérico 0; los círculos en cada región representan los vectores de soporte correspondientes.

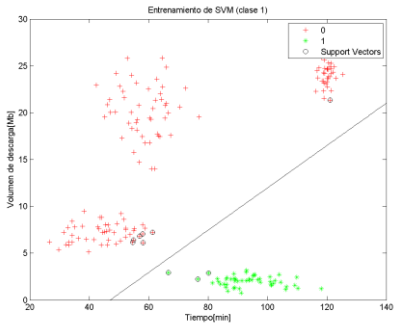


Figura 3. Entrenamiento del SVM Clase 1 (Celulares). Fuente propia.

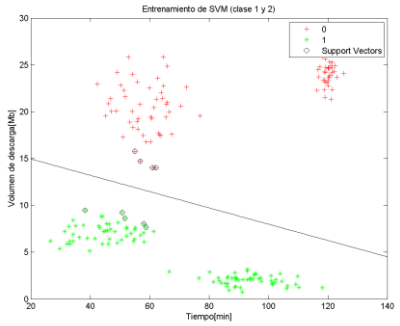


Figura 4. Entrenamiento del SVM Clase 1 y 2 (Celulares y Tabletas). Fuente propia.

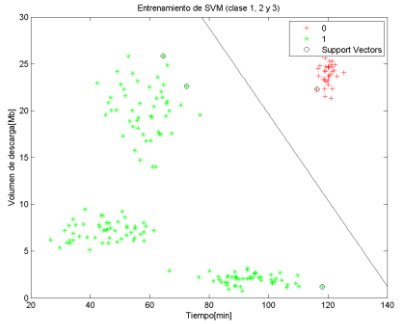


Figura 5. Entrenamiento del SVM Clase 1, 2 y 3 (Celulares, Tabletas y Portátiles). Fuente propia.

En la figura 4, se muestra la recta que separa las clases Celulares y Tabletas de los demás equipos, bajo la misma convención que la primera fase. Finalmente, se entrena el clasificador para identificar las clases Celulares, Tabletas y Portátiles del cuarto tipo de equipo Computadores de escritorio. En la figura 5 se muestra la recta que separa las tres primeras clases de la cuarta.

5.1.2. Clasificador ANN

Para el proceso de entrenamiento del clasificador tipo ANN, se hizo uso del toolbox de Matlab ***nntool,*** se llevó a cabo con el algoritmo de backpropagation de tipo feedforward, teniendo en cuenta los siguientes pasos:

- ✓ Configurar los datos de entrenamiento: Se cargó la base de datos para el entrenamiento de la red neuronal; así, se dispuso de 200 muestras con 2 características correspondientes a los input Data, como se mencionó anteriormente las muestras corresponden al tipo de equipo conectado a la red WiFi. De la misma manera también se cargó las etiquetas para cada clase, que corresponden a los Target data, debido a que este algoritmo utiliza aprendizaje supervisado. Cabe resaltar que para este clasificador, el entrenamiento también se hace de manera segmentada para cada clase.
- ✓ Creación de la red: Se tuvieron en cuenta los siguientes parámetros; Tipo de red: Feed Forward Backpropagation, Función de entrenamiento: TRAINLM, Función de aprendizaje de adaptación: LEARNGDM, Función de rendimiento: MSE, Numero de capas: 4 capas ocultas, y 1 capa de salida. Se utilizaron 10 neuronas para las capas ocultas con función de transferencia Tan-Sigmoid para todas las neuronas, incluyendo la capa de salida.
- ✓ Entrenar la red: Para la etapa de entrenamiento se tuvo en cuenta los valores de la tabla 1.

Tabla 1. Parámetros de Entrenamiento ANN

Parámetro	Valor
epochs	1000
time	inf
show	25
goal	0
min_grad	1,00e-07
max_fail	6
mu	1,00e-03
mu_dec	0.1
mu_inc	10
mu_max	1,00e+10

A continuación se dará una breve explicación del funcionamiento del algoritmo [6], para la etapa del entrenamiento de la red. El parámetro de mu es el valor inicial de μ, y este actúa de acuerdo a como se comporta la función de desempeño, así μ es multiplicado por mu_dec siempre que la función de desempeño reduzca su paso y por el contrario es multiplicado por mu_inc siempre que la función de desempeño aumenta un paso, y sí μ es mayor que mu_max, el algoritmo se detiene. El algoritmo también se detiene sí la magnitud del gradiente está por debajo de min_grad, el tiempo en esta configuración no afecta y max_fial está asociado con la técnica de detención temprana.

5.2. Clasificación

Una vez se tienen los clasificadores entrenados, se procede a probar su funcionamiento con los valores indicados en la figura 6, de esta manera se muestra la distribución de los 800 datos o equipos con conexión WiFi, según cada clase, con los cuales se puso a prueba los dos clasificadores. Los resultados que se obtuvieron en la clasificación de los 800 datos para los dos clasificadores se presentan en la figura 7.

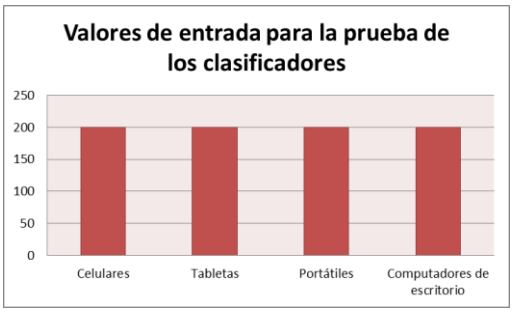


Figura 6. Valores de entrada para la prueba de los clasificadores. Fuente propia.

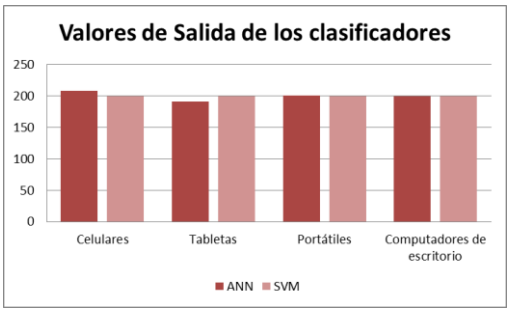


Figura 7. Valores de salida de los clasificadores. Fuente propia.

5.3. Medidas de Rendimiento de los Clasificadores

Para presentar los resultados de los clasificadores es común el uso de la matriz de contingencia o de confusión, que recoge el número de aciertos y fallos del sistema [7], para entender dicha matriz es necesario tener en cuenta las siguientes definiciones:

- ✓ Detección correcta o aceptación verdadera (TP, True Positive): el número de patrones de la clase 1 que el sistema clasifica correctamente como pertenecientes a la clase 1.
- ✓ Falso Rechazo (FN, False Negative): el número de patrones de clase 1 que el sistema clasifica incorrectamente como pertenecientes a la clase 0.
- ✓ Falsa aceptación (FP, False Positive): el número de patrones de la clase 0 que el Sistema clasifica incorrectamente como pertenecientes a la clase 1
- ✓ Rechazo verdadero (TN, True Negative): el número de patrones de la clase 0 que el sistema clasifica correctamente como pertenecientes a la clase 0.

La matriz de confusión se muestra en la tabla 2.

Tabla 2: Matriz de confusión

		Clase real	
		1	0
Clase estimada	1	TP	FP
	0	FN	TN

Los parámetros que se pueden estimar, para evaluar los sistemas de clasificación, según la matriz de confusión son los siguientes:

- ✓ Tasa de acierto o eficiencia (CCR, Correct Clasification Rate): Proporción de patrones correctamente clasificados por el sistema.

$$CCR = \frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

- ✓ Tasa de error (ER, Error Rate): Proporción de patrones erradamente clasificados.

$$ER = 1 - CCR = \frac{FN+FP}{TP+FN+FP+TN} \quad (2)$$

- ✓ Sensibilidad (S): Da una indicación de la capacidad del sistema para detectar los patrones de la clase de referencia.

$$S = \frac{TP}{TP+FN} \quad (3)$$

- ✓ Especificidad (E): Da una indicación de la capacidad del sistema para rechazar los patrones que no pertenecen a la clase de referencia.

$$E = \frac{TN}{TN+FP} \quad (4)$$

Los resultados que se obtuvieron para la medición del rendimiento de los clasificadores a partir de los datos de clasificación son consignados en la tabla 3.

Tabla 3. Parámetros de medición de rendimiento

Clasificador	TP	TN	FP	FN	CCR	ER	S	E
ANN	400	397	3	0	99,63%	0,38%	100,00%	99,25%
SVM	400	400	0	0	100,00%	0,00%	100,00%	100,00%

6. Conclusiones

La distribución de los vectores de características según las cuatro clases, son fácilmente diferenciables, por lo que la tarea de identificación y clasificación no reviste mayor complejidad. Esto explica los valores de error de clasificación bajos para los dos tipos de clasificadores.

De los dos clasificadores SVM y ANN, el primero resulta con tasa de error nula; tasa de acierto, sensibilidad y especificidad ideales, se concluye que la SVM es la adecuada para la tarea de clasificación.

En la administración de una red de datos se requiere conocer además del número de clientes conectados, el tipo de dispositivo y el volumen de descarga demandada; parámetros con los cuales se podría optimizar el uso de los recursos de una organización, esta es la aplicación futura que se pretende continuar partiendo de la actual investigación.

7. Referencias

[1] Romo, Harold Probabilidad y procesos estocásticos, First Edition, Ed EAE 2011.
[2] Fukunaga, Keinosuke, Introduction to statical pattern recognition, Second Edition, Ed.Academic Press, p. 20-200, 1990.
[3] Haykin, S, Neural Networks: A Comprehensive Foundation, Second Edition, Ed. Pearson Education Press, p. 178- 270, 2001.
[4] Haykin, S, Neural Networks: A Comprehensive Foundation, Second Edition, Ed. Pearson Education Press, p. 181, 2001 [Figura]
[5] Holmstrom, L., Koistinen, P., Laaksonen, J., Oja, "E. Neural and statistical classifiers taxonomy and two case studies" IEEE Trans. Neural Networks, 8, 5–17, 1997.
[6] trainlm, (Introduced before R2006a), [online]. Disponible en: https://es.mathworks.com/help/nnet/ref/trainlm.html.
[7] Villas T, "Metodología de análisis tiempo-frecuencia para la evaluación automática de la voz de pacientes con enfermedad de Parkinson", M.s. tesis, Universidad de Antioquia, Medellin, Colombia, 2015.

8. Hoja de vida de los autores

Andrea Johana Chaves Estudiante de Ingeniería Electrónica de la Universidad de Nariño, Colombia. Sus temas de investigación de interés son las aplicaciones de la inteligencia artificial aplicadas a sistemas telemáticos.

Oscar Javier Jossa Estudiante de Ingeniería Electrónica de la Universidad de Nariño, Colombia. Sus temas de investigación de interés son las aplicaciones de la inteligencia artificial aplicadas a sistemas telemáticos.

Mario Fernando Jojoa Acosta Recibió el título de ingeniero electrónico de la Universidad de Nariño en el año 2009, es un estudiante de maestria en Electrónica y Telecomunicaciones en la Universidad del Cauca. Sus temas de investigación de interés son las aplicaciones de clasificadores y redes neuronales en sistemas telemáticos.

Certificado de ponencia:

Centro Internacional de Investigación Científica en Telecomunicaciones, Tecnologías de la Información y las Comunicaciones CITIC

Centro de Excelencia de la Unión Internacional de Telecomunicaciones UIT, Organismo de Naciones Unidas ONU.

Certifica que:

Oscar Javier Jossa Bastidas

Participó como ponente del artículo "Clasificación de Hosts en una red WLAN usando un sistema basado en clasificadores tipo Máquina de Soporte Vectorial y Red Neuronal Artificial." en la

XI CONFERENCIA CIENTÍFICA DE TELECOMUNICACIONES, TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

20, 21 Y 22 de Noviembre de 2017
Quito-Ecuador
Patrimonio Cultural de la Humanidad

Ing. Mauro Flórez C. Ph.D, MsC, Esp, PhD(c)
PRESIDENTE CITIC

Ing. Zoila Ramos R. Ph.D (c), Esp. Dpl.
DIRECTORA GENERAL CITIC

ANEXO C. Certificado de participación en el IV Congreso Internacional de Innovación y Tendencias en Ingeniería

Este anexo contiene el articulo seleccionado para la sustentación en modalidad de ponencia en el “IV CONGRESO INTERNACIONAL DE INNOVACIÓN Y TENDENCIAS EN INGENIERÍA,” desarrollado en Bogotá-Colombia, durante los días 3, 4 y 5 de octubre de 2018.

Publicación:

- El artículo es publicado en las memorias del congreso CONIITI 2018
- El árticulo es aceptado para publicarse en “IEEE Xplore® digital library”, a finales de noviembre del presente año, con indexación en Scopus.

Certificado de ponencia: ““IV CONGRESO INTERNACIONAL DE INNOVACIÓN Y TENDENCIAS EN INGENIERÍA.” CONIITI 2018

UNIVERSIDAD CATÓLICA de Colombia
Vigilada Mineducación

Coniiti

Hace constar que

Andrea Johana Chaves Villota, Oscar Javier Jossa Bastidas and Mario Fernando Jojoa Acosta

Participaron en el IV Congreso Internacional de Innovación y Tendencias en Ingeniería CONIITI, con la ponencia:

“Classification of Hosts in a Wlan based on Support Vector Machine”

Realizado en Bogotá del 03 al 05 de Octubre de 2018

Registro No.AAFI053134CI18

EDWIN DANIEL DURAN
CONIITI General Chair

JAIME DÍAZ ORTIZ
Decano Facultad de Ingeniería

Artículo: Classification of Hosts in a WLAN based on Support Vector Machine.

Classification of Hosts in a WLAN based on Support Vector Machine

Andrea Chaves
Universidad de Nariño
Pasto, Colombia
andreajohanacv@udenar.edu.co

Oscar Jossa
Universidad de Nariño
Pasto, Colombia
oscarjte@udenar.edu.co

Mario Jojoa
Universidad del Norte
Barranquilla, Colombia
jojoam@uninorte.edu.co

***Abstract*—Nowadays, wireless local area networks (WLAN) are communication structures highly used in organizations and residences, since they allow greater flexibility, adaptability and comfort for users; However, they have difficulty in optimizing the bandwidth depending on the traffic demand, since some users require more or less bandwidth according to the activity carried out. Therefore, granting a quality service in WLANs is a latent need, hence this problem has been the subject of multiple studies involving artificial intelligence, to determine the behavior of telecommunications networks and as support to making management decisions. In this paper, the use of an artificial intelligence technique is proposed to study behavior of the WLAN users at the University of Nariño.**

***Index Terms*—WLAN, K-means, SVM, K-folds clustering**

I. Introduction

The number of internet users reflects that it has become indispensable for the daily life . In this moment around 40% of the world population has an Internet connection, indicating that the number of users has increased in 1000 % percent between the years 1999 and 2013 [1]. Currently, it is common to find connections to the Internet through local area networks (LAN) and WLAN; hence it is important to provide a quality service for network customers. On the other hand, with the rise of modern technology, in recent decades computer systems have been incorporated with many purposes like diagnose diseases, plan the synthesis of complex organic chemical compounds, solve complex differential equations, analyze electronic circuits or understand human speech in different languages; these systems have been developed with some technique of artificial intelligence with high levels of reliability [2].

In consequence the increasing bandwidth demand of end-users renders the need to get efficient resources to manage next generation wireless networks with new approachs as prediction process based on machine learning techniques [3]. A well-known learning algorithm is Support Vector Machine which is very popular and is used by researchers from different areas of knowledge, reporting good results compared with other machine learning as decision trees or neural networks. This paper focuses the study of how could be used the clients behavior of the wlan of the University of Nariño, where professors, administrative and students connect daily to manage effectively the bandwidth using ML techniques. To do this a repository was builded with the extracted features from users; the features extracted are 1) instance connection time and 2) download and upload streaming data, later a clustering k-means algorithm is used to label the users data and perform a classification with a SVM, the model is finally evaluated with a K-folds cross validation technique.

978-1-5386-8131-2/18/$31.00 ©2018 IEEE

II. Background

A. Wireless Local Area Network (WLAN)

Wireless LAN networks are increasingly popular; in homes, offices, cafeterias, libraries, airports and other public places are being equipped with this type of networks to connect computers, tablets and smartphones to the Internet. Wireless LAN networks can also be used to allow two or more computers that are close to each other to communicate without using the Internet [4]. The main standard for wireless LAN is 802.11, published IEEE 802.11-2007 [5]. In this research, the wlan at the University of Nariño has been studied, the wireless network administration is carried out by a Cloud Core Router wich through the radius server do authentication of students, administrative and professors. The Cloud Core allows to take data in real time of the users consumption, the data have been used as base of study in this paper.

B. Cluster analysis: K-Means

The goal in cluster analysis is to find groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated) to the objects in other groups, the greater the homogeneity within a group and the greater the difference between groups, the better or more distinct the clustering. Clustering can be regarded as a form of classification in that it creates a labeling of objects with class (cluster) labels, for this reason is sometimes referred to as unsupervised classification [6]. K-means algorithm is one of the most popular clustering techniques, it has been used widely on science, industry and business [7]. This algorithm is based on distance and partitions N data into K clusters, where the number of clusters K has to be known a priori [8]. Its cluster similarity criterion is the distance between data objects. The data of same cluster similar and the data of different clusters is different. K-means defines a prototype in terms of a centroid, which is usually the mean of a group of points. The

basic K-Means algorithm is based in first K initial centroids, where K is the number of clusters desired. Each data point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. The assignment and update is repeated until no point changes clusters, or equivalently, until the centroids remain the same. To assign a point to the closest centroid is necessary a proximity measure that quantifies the notion of closest for the specific data under consideration. Euclidean distance is often used for data points in Euclidean space, thus the goal of clustering is to find the objective function that minimize the squared distance of each point to its closest centroid. The sum of the squared error (SSE) is used to calculate the error of each data point, i.e. its Euclidean distance to the closest centroid, and then compute the total sum of the squared errors, the SSE is defined in 1

$$SSE = \sum_{i=1}^{k} \sum_{x \in c_i} dist(c_i, x)^2 \quad (1)$$

Where $dist$ is the standard Euclidean distance, x an object, C_i the i^th cluster and c_i the centroid of cluster C_i.

Given these assumptions, it can be shown that the centroid that minimizes the SSE of the cluster is the mean [6] defined by 2

$$c_i = \frac{1}{m} \sum_{x \in c_i} x \quad (2)$$

Where m_i is the number of objects in the i^th cluster. The Basic K-means algorithm is described in the Table I.

TABLE I
BASIC K-MEANS ALGORITHM

Algorithm
1: Select K points as initial centroids
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid
4: Recompute the centroid of each cluster
5: **until** Centroids do not change

C. Support Vector Machine (SVM) Classifier

SVM is a statistical learning has attracted a great deal of attention in the last decade technique, used in various classification problems that. SVMs are based on statistical learning theory and structural risk minimization principle with the aim of determining the location of decision boundaries, also known as hyper-plane, that produce the optimal separation among the classes [9]–[13]. For linearly separable data, the machine constructs an optimal separating hyper-plane as a decision surface, to divide the data of different categories in the vector space produce the optimal separation among the classes. However, for the non-linearly separable data, the Kernel functions are used to extend the concept of the optimal. Separating hyper-plane so that the data can be linearly separable [10], [11], [13]. The kernels functions have different characteristic and the performance of the SVM is highly influenced by these. Let us assume initially that for a given training set according to 3.

$$S = \{(x_1, y_1), ..., (x_i, y_i)\} \quad (3)$$

Where $x_i \in R^d$ and $y_i \in \{-1, 1\}$. If the training data are linearly separable, the hyper-plane that optimally separates the linearly separable data is obtained by minimizing the following function

$$\phi(w) = \frac{1}{2 \|w^2\|} \quad (4)$$

Subject to $y_i(w \cdot x_i + b) \geq 1, \forall (x_i, y_i) \in D$, where w is weight vector and b is bias. For the non-linearly separable data, The kernels functions are used, so the optimization problem in the high dimensional feature space turns to be in 5

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j K(\bar{x_i}, \bar{x_j}) \quad (5)$$

Subject to 6

$$\sum_{i=1}^{l} \alpha_i y_i = 0, and \geq \alpha_i \geq 0 \quad (6)$$

Where, C is a regularization parameter that controls the trade-off between the margin and the training error term, α is a Lagrangian multiplier and $K(\bar{x_i}, \bar{x_j})$ is the kernel function, the classification function of SVM is defined by 7

$$f(x) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(\bar{x_i}, \bar{x_j}) + b) \quad (7)$$

In this research the radial Basis Function Kernels (RBF) has been used, whose kernel 8 is represented

$$k(x_i, y_i) = exp(\frac{\|x_i - y_i\|^2}{2\sigma^2}) \quad (8)$$

Some characteristics the RBF kernel function are: The feature space is infinite dimensional and the kernel function produces a Gaussian separating hyper-plane [14].

D. K-Fold Cross Validation

In the development of classifiers, the performance measure is a main factor to find the best algorithm that solves the task in question. In algorithm evaluation, is compared several learning algorithms, to choose the one that obtains a better performance measure according to the criteria. A practical approach consists in using the K-fold CrossValidation technique (KCV) which to estimate the generalization error of the classifier [15]–[17]. The KCV technique is one of the most used approaches by practitioners for model selection and error estimation of classifiers. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method

is fit on the remaining $k1$ folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error, MSE_1, MSE_2... MSE_k. The k-fold CV estimate is computed by averaging these values according to 9

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i \tag{9}$$

There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically is performs k-fold cross-validation using $k = 5$ or $k = 10$, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance [16], [18].

III. EXPERIMENTAL RESULTS

A. Data base characteristics

The repository was formed with data from the wireless network of the University of Nariño, panamericana campus, where there is 150 users average connected to the network and can reach up to 300 clients during peak hours. The first step was to extract the data, using a script developed in the operating system RouterOS of mikrotik,which extracted average number of 4800 registers in real time per day. The following fields in plain text were included: user, mac, connection time, bytes in, bytes out, time label. The units are connection time [s], bytes in [B], bytes out [B]. Subsequently with this data proceeds to develop a database in the Structured Query Language (SQL). An observation window was taken according to the traffic statistics provided by NAP (Network Access Point) Colombia, which indicates that as from 8 am the traffic tends to grow. Likewise at 2017 there was a high navigation consumption in March and April. In addition, it was taken into account that institution develops academic and administrative activities during these months and in these hours. Therefore, it was decided to make the data collection from 9 a.m. to 4 p.m., during the months of March and April of 2018.

B. Data processing

The data processing was carried out in the high-level programming language Python and the machine learning algorithms using the library [19]. A script that allows to make the connection with the database in mysql in a local server is developed, the goal is to do the data processing described in Fig. 1.

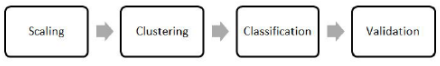


Fig. 1. Data processing

1) Scaling: In this research the learning scheme numeric attributes were measured on ratio scales, so the question of normalization arises. Attributes are often normalized to lie in a fixed range usually from 0 to 1 by dividing all of the values by the maximum value encountered or by subtracting the minimum value and dividing by the range between the maximum and minimum values [20]. Consequently it was necessary to develop a simple scaling as follows: let us assume $x = \{x_1, x_2, , x_n\}$, where x is a feature vector and $x_n \epsilon R$. The scaling was developed according to 10

$$x' = \frac{1}{max(x)} \tag{10}$$

In this research the features, connection time and bytes out were taken as variables of study. Its shown in Fig. 2.

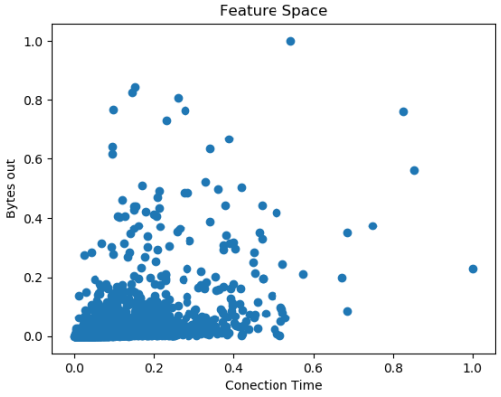


Fig. 2. Feature Space

2) Clustering: To group the data k-means clustering algorithm was used, to obtain a class label that depends on the number of chosen clusters. For the selection of the most suitable K, the elbow curve was used, shown in the Fig. 3.

Specifically, a range from 1 to 18 was used (which represents the number of clusters), and score variable denotes the percentage of variance explained by the number of clusters, is shown that the graph levels off rapidly after 5 clusters, implying that addition of more clusters do not explain much more of the variance in our relevant variable. In this way in Fig. 4 the 5 clusters are indicated, this represent the 5 classes that will be classifying.

3) Classification: Once the class label is obtained, the next step is to train the vector support machine, using linear, polynomial and rbf kernels. The C and $gamma$ values were taken as parameters to establish the performance of the vector support machine and were obtained according to Fig. 5.

The map plot has a special colorbar with a midpoint value close to the score values of the best performing models so as to make it easy the parameter selection. The behavior of the

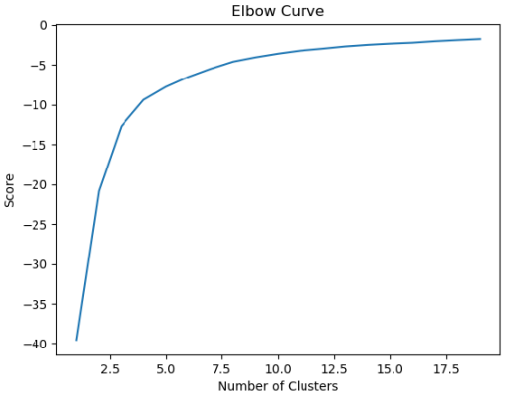


Fig. 3. Elbow Curve

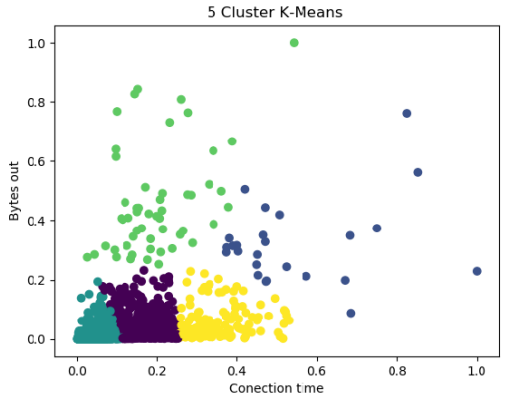


Fig. 4. Labeled classes

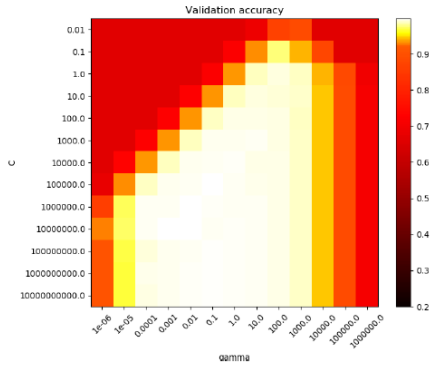


Fig. 5. Parameters Validation

model is very sensitive to the gamma parameter. If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. When gamma is very small, the model is too constrained and cannot capture the complexity or shape of the data. The region of influence of any selected support vector would include the whole training set. Finally one can also observe that for some ivalues of gamma we get equally performing models when C becomes very large: it is not necessary to regularize by limiting the number of support vectors. However in this work to limit the number of support vectors with a lower value of C so as to favor models that use less memory and that are faster to predict [21]. According to the Fig. 5, the values of C and gamma chosen for this paper are 10in both cases. The results obtained for the SVMs of different kernels are shown in Fig. 6, Fig. 7 and Fig. 8.

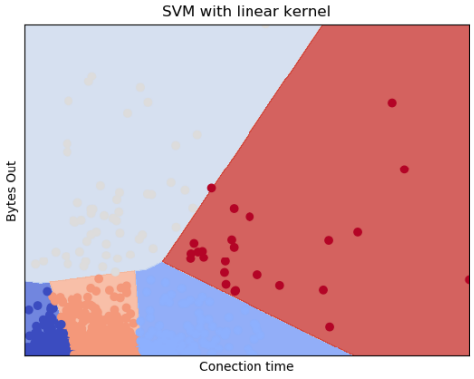


Fig. 6. Linear SVM C = Gamma = 10

4) Validation: The KCV technique is used to validate the classifiers performance with a $k = 5$; the results obtained are recorded in II.

TABLE II
K-FOLDS CROSS VALIDATION RESULTS

Classifier	Accuracy	Scores
SVM Linear	0.99 (+/- 0.00)	[0.9937, 0.9884, 0.9883, 0.9901, 0.9901]
SVM Polynomial	0.99 (+/- 0.00)	[0.9973, 0.9928, 0.9946, 0.9928, 0.9928]
SVM RBF	1.00 (+/- 0.00)	[0.9991, 0.9946, 0.9964, 0.9964, 0.9937]

In a model with values C and gamma: 100, 1 respectively, the SVM kernel that obtains the best results are the rbf and the linear. It should be noted that the polynomial model can achieve greater precision with different C and gamma values.

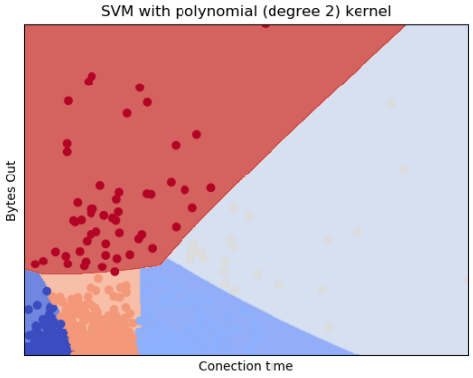


Fig. 7. Polynomial SVM C = Gamma = 10

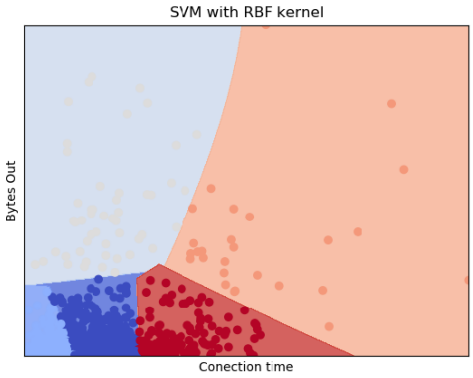


Fig. 8. RBF SVM- C = Gamma = 10

However, the computational cost and an others parameters must be taken into account in the classifiers designs.

IV. Analysis of Experimental Results

From the experimental results, it's concluded that users have a different performance when surfing the internet. This is reflected in the clusters, in this way it was found that much of the samples are related to the data coming from the repository, as follows:In the cluster of blue color of figure 4, it is observed that users who spend more time connected and download most are located, and it was found in the database that a large part of these users corresponds to administrative; which is logical if analyzed since they have to be connected most of the day and therefore have a greater download volume. The difference in terms of students and teachers was not as marked, However it could also be observed that the latter tend to be located in the green cluster, this can be interpreted as follows, teachers don't connect for long periods of time but download more, due to the use of platforms and resources such as YouTube. The majority of students are divided into the first three clusters, with a greater participation in the first tiffany blue, Which also makes sense since most students use applications in which the connection time is not quite long, and low consumption such as whatsapp, messenger among others. As for the remaining two clusters, as mentioned there is also a large number of students, which could be deduced that these students are those who meet in places like the library to do assignment, and therefore aren't so fleeting, If not spend more time connected to the university network.

On the other hand the SVM did a good job in the classification stage identifying the 5 clusters, which is of great importance for the next step corresponding to the allocation of bandwidth, since it corresponds to 5 navigation profiles. Now the importance of identifying these profiles is that the bandwidth will be assigned to new clients according to the group they belong to, thus the channel will be distributed dynamically taking into account the user's consumption levels. A possible immediate application emerges analyzing 4, as mentioned above the cluster of blue color has a high degree of relationship with the administrative staff that have high consumption rates and long connection times, therefore it would be convenient for the system to allocate more resources to these profiles and mitigate resources for the student group. Or more applications may arise, depending on how the system is configured. A great advantage of this is that an autonomous prototype is proposed, in which the administrator does not have to intervene.

V. Conclusions

The heat map, turned out to be of great help because according to its analysis it was obtained more easily the adequate values so that the classifier obtains precisions up to 99%, optimizing values of C and thus obtaining a low computational cost In the administration of a data network it is required to know in addition to the number of connected clients, the volume of download demanded; parameters which can optimize the use of the resources of an organization, according to the above the next step is to make a dynamic allocation of the available bandwidth in a wlan network, doing relations between users behavior and the navegation profiles in this work were accomplished The idea of implementing a dynamic bandwidth allocation system based on the navigation data is correct, since, as in this work is observed, is possible to differentiate the behavior of the users of the network

References

[1] (2016) Internet live stats. [Online]. Available: www.Internetlivestats.com/internet-users/

[2] N. J. Nilsson, *Principles of Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1980.

[3] I. Loumiotis, T. Stamatiadi, E. Adamopoulou, K. Demestichas, and E. Sykas, "Dynamic backhaul resource allocation in wireless networks using artificial neural networks," *Electronics Letters*, vol. 49, no. 8, April 2013.

[4] A. Tanenbaum, *Redes de computadoras*. Editorial Alhambra S. A. (SP), 2003. [Online]. Available: https://books.google.com.co/books?id=d_m3W_Yob8kC

[5] IEEE, "Ieee standard for information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-1997*, pp. i –445, 1997.

[6] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Education, 2006.

[7] L. W. Li Beibei, Liu Bo and Z. Ying, "Performance analysis of clustering algorithm under two kinds of big data architecture," *Journal of High Speed Networks*, vol. 23, no. 1, pp. 49–47, 2017.

[8] B. Peralta, P. Espinace, and A. Soto, "Enhancing k-means using class labels," *Intell. Data Anal.*, vol. 17, no. 6, pp. 1023–1039, Nov. 2013. [Online]. Available: http://dx.doi.org/10.3233/IDA-130618

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[10] H. Bhavsar and A. Ganatra, "Eudic svm: A novel support vector machine classification algorithm," *Intell. Data Anal.*, vol. 20, pp. 1285–1305, 2016.

[11] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[12] O. Bousquet and B. Schölkopf, "Comment on support vector machines with applications by j. m. moguerza and a. muoz," *Statistical Science*, vol. 21, no. 3, pp. 337–340, Aug. 2006.

[13] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998. [Online]. Available: https://doi.org/10.1023/A:1009715923555

[14] H. Bhavsar and A. Ganatra, "Increasing efficiency of support vector machine using the novel kernel function : Combination of polynomial and radial basis function," vol. 3, no. 5, 2014, pp. 17–54.

[15] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The 'k' in k-fold cross validation," in *ESANN*, 2012.

[16] I. Guyon, A. Saffari, G. Dror, and G. Cawley, "Model selection: Beyond the Bayesian/frequentist divide," *The Journal of Machine Learning Research*, vol. 11, pp. 61–87, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756009

[17] D. Anguita, A. Ghio, S. Ridella, and D. Sterpi, "K-fold cross validation for error rate estimate in support vector machines," in *DMIN*. CSREA Press, 2009, pp. 291–297.

[18] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078195

[20] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[21] (2011) Scikit-learn. [Online]. Available: http://scikit-learn.org/stable/auto$_e$$xamples/svm/plot_rbf_parameters.html$

ANEXO D. Tutorial de Configuraciones de enrutadores marca Mikrotik.

El presente anexo muestra los pasos a seguir para la configuración de enrutadores de la marca Mikrotik.

1. CONFIGURACIÓN RED LAN

A continuación, se presenta un breve tutorial de la configuración de una red LAN, en este caso fue utilizado un routerboard wAP sin embargo el tutorial se acopla a todos los enrutadores de la marca Mikrotik que cuenten con RouterOS como sistema operativo, en este documento se indica los pasos con la interfaz de winbox.

1.1. Cambio nombre de interfaces de red: Por motivos de organización es conveniente trabajar con nombres en las interfaces de red, de ahí que se cambiara el nombre.

Para esto seguimos los siguientes pasos:

- Click en **Interfaces** en el menú principal del lado izquierdo.

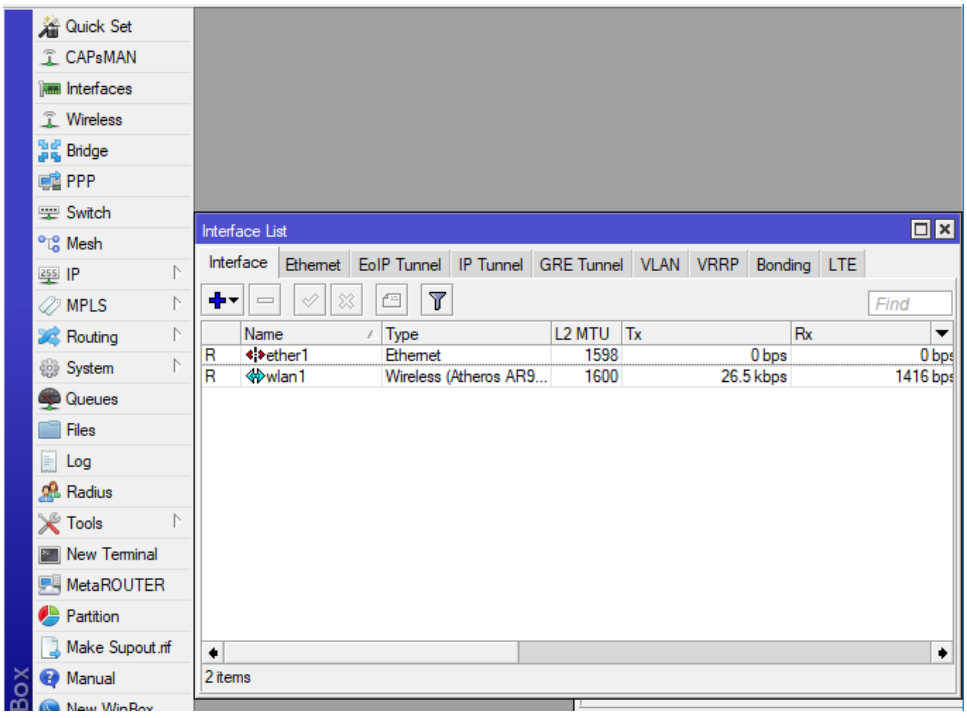


Figura 1. Ventana de interface list

- Se selecciona la interfaz deseada y se da doble click, de inmediato se abre una nueva ventana y en el campo **Name** cambiamos el nombre, en este caso ETHER1-WAN y damos click en **Apply**, la llamamos de esa forma, ya que esta interface en nuestro caso corresponde a la entrada de internet

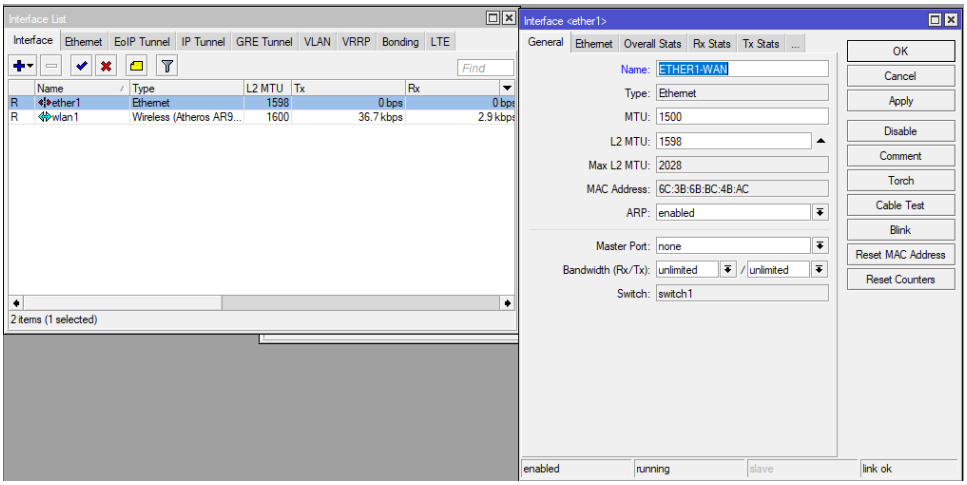


Figura 2. Cambio de nombre de interface

Así se hace para el resto de las interfaces que se desee cambiar.

1.2. Asignación de listas de direcciones IP: Luego se debe configurar la interface que se conecta a internet con una dirección IP dentro de la misma red que nos conectamos (comúnmente denominada WAN). Esto se puede hacer con un cliente DHCP o como en este tutorial con una dirección IP Fija, Para esto seguimos los siguientes pasos:

- Click en **IP** luego en **Addresses,** aparece una nueva ventana, ya dentro le damos click en el simbolo **+** como se indica en el recuadro de la siguiente imagen.

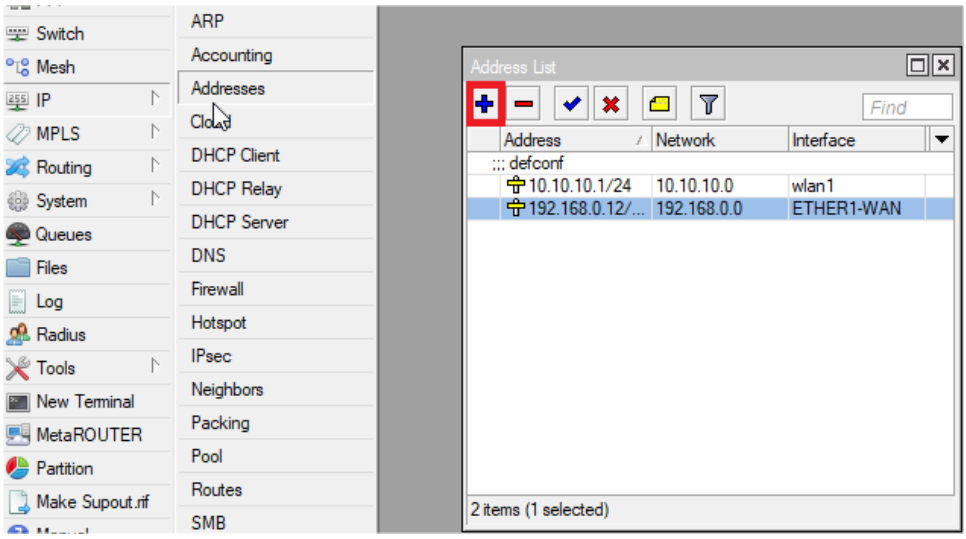


Figura 3. Ventana de Address List

- De inmediato aparece una ventana, en el campo de **Address,** digitamos la dirección IP, junto con la máscara de red y le damos click en **Apply,** inmediatamente la dirección de red se ubica automáticamente, en el siguiente camo seleccionamos la **interface** a la que deseemos aplicar el cambio, en este caso a ETHER1-WAN, luego damos en **OK.**

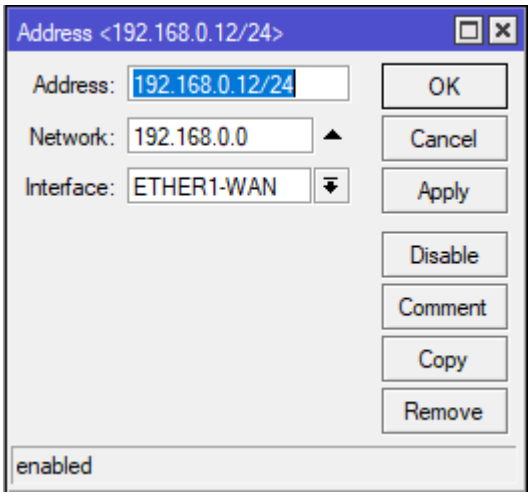


Figura 4. Parámetros de configuración interface WAN

- De la misma forma que para el caso anterior, configuramos la interface que creara nuestra red LAN, para ello damos el direccionamiento deseado, luego de estar en la ventana **Address list** como se indica en la figura 3 damos clcik nuevamente en el signo **+** y configuramos los parámetros de nuestra red LAN. En este caso se hacen las

configuraciones para wlan1 que corresponde a una interface inalámbrica incorporada en la RouterBoard wAP.

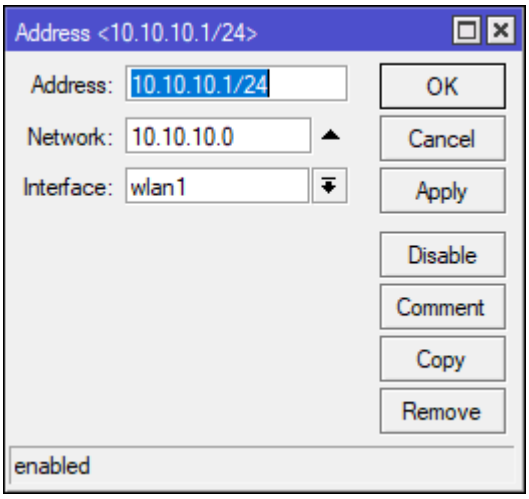


Figura 5. Parámetros de configuración interface LAN

1.3. Configuración Route: Se establece una ruta, ubicando la dirección de la puerta de enlace de la red a la cual nos estemos conectando.
Para esto seguimos los siguientes pasos:

- Click en **IP** luego en **Routes,** aparece una nueva ventana llamada **Routes List**, ya dentro le damos click en el simbolo **+** como se indica en el recuadro de la siguiente imagen.

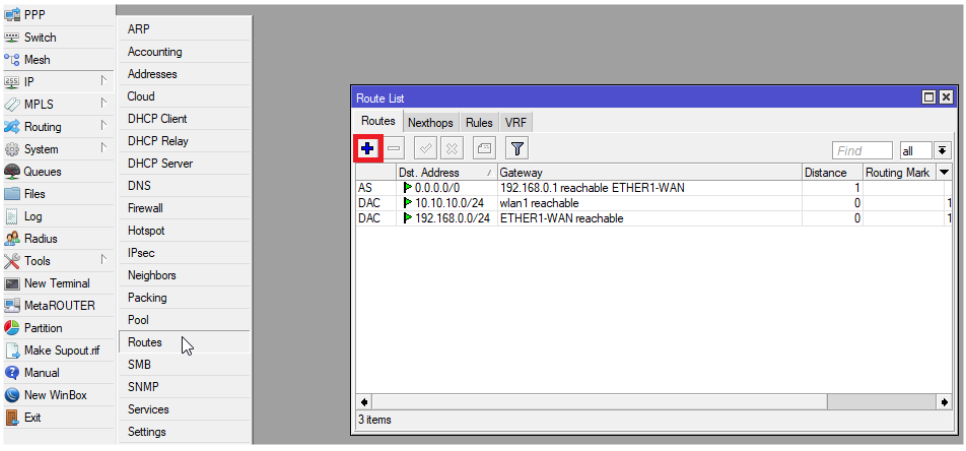


Figura 6. Panel principal Routes List

- En el campo de **Gateway,** ubicamos la dirección IP de la puerta de enlace perteneciente a la red a la que nos estemos conectando y le damos **OK.**.

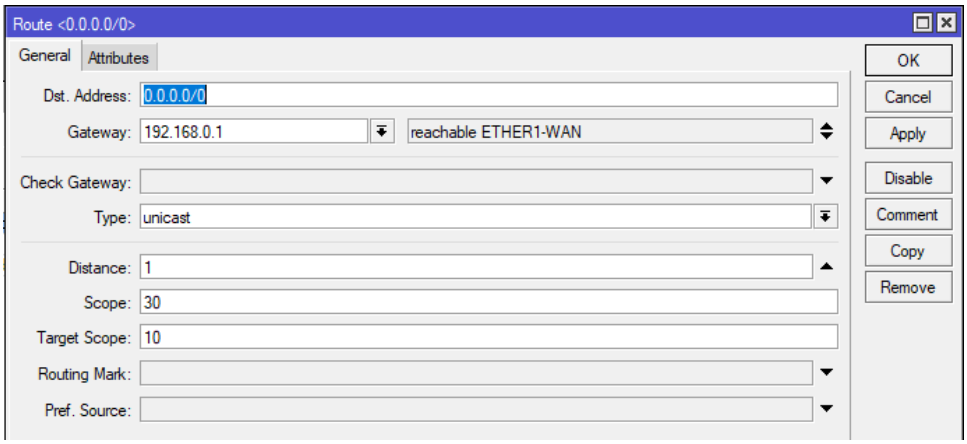


Figura 7. Configuración ruta- puerta de enlace

- Hecho esto podemos hacer una comprobación, haciendo un ping a la puerta de enlace de dicha red, y si esta red tiene salida a internet con un ping a 8.8.8.8 podremos confirmarlo desde el equipo. Para hacer esto damos click en **New Terminal,** y en el terminal digitamos el comando como se muestra en la figura,

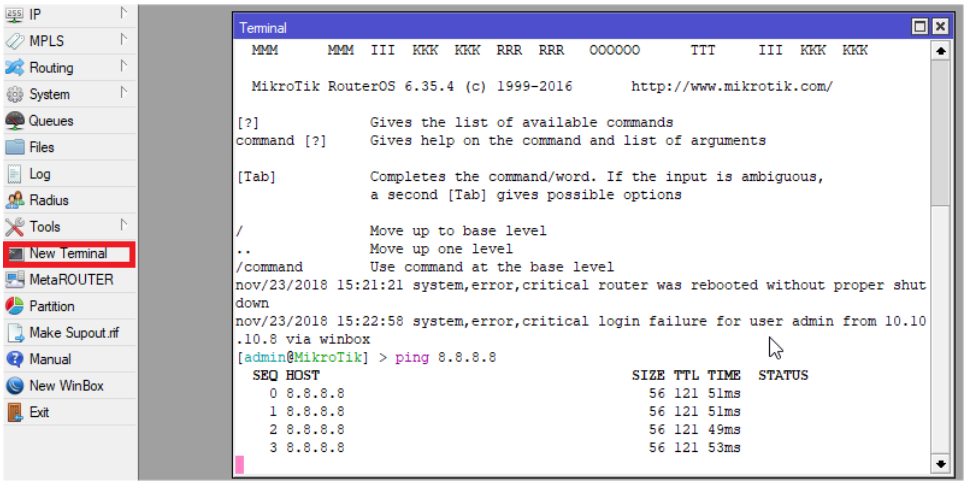


Figura 8. Comprobación ping, salida a internet desde el equipo

1.4. Configuración servidor DHCP: Posteriormente se debe configurar el servidor DHCP a la interface de la LAN para que se realice una asignación dinámica de las direcciones IP.

Para esto se sigue los siguientes pasos:

- Click en **IP** luego en **DHCP Server,** aparece una nueva ventana, ya dentro le damos click en **DHCP Setup,** y seguimos los siguientes

pasos

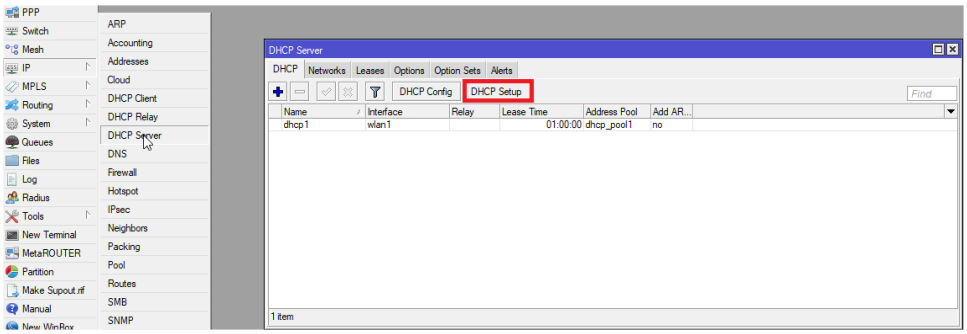


Figura 9. Panel principal configuración servidor DHCP

- Seleccionamos la Interface

Figura 10. Selección interfaz para configurar DCHP

- Seleccionamos la red para el direccionamiento DHCP

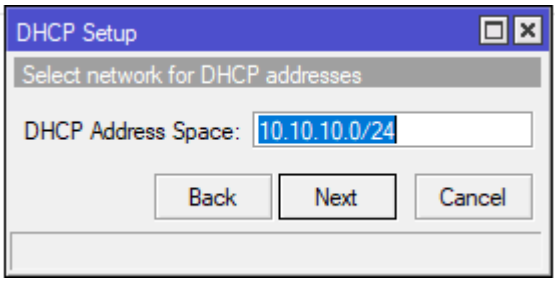


Figura 11. Selección red para configurar DCHP

- Seleccionamos el gateway

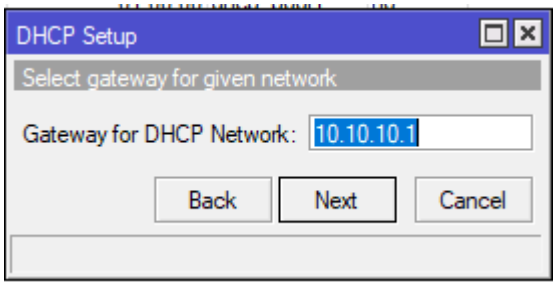


Figura 12. Selección puerta de enlace para configurar DCHP

- Seleccionar un rango deseado de direcciones IP a las que se asignará a través del servidor DHCP

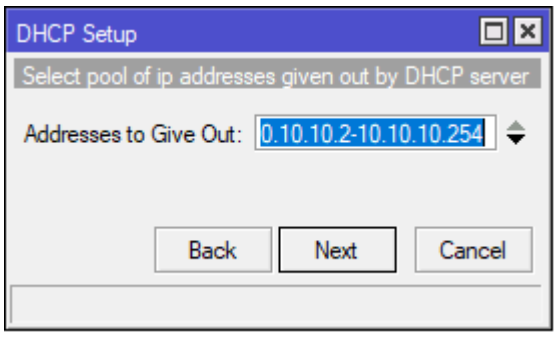


Figura 13. Selección rango de IP

- Seleccionar los servidores DNS

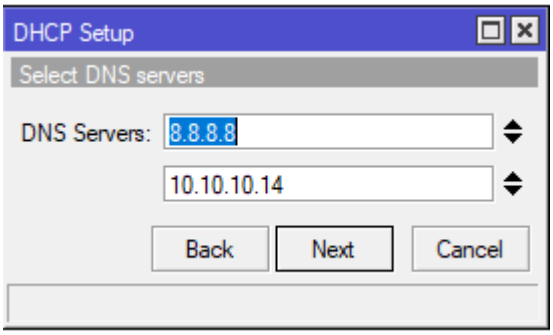


Figura 14. Configuración servidores DNS

- Establecer el tiempo de arrendamiento para el servidor DHCP

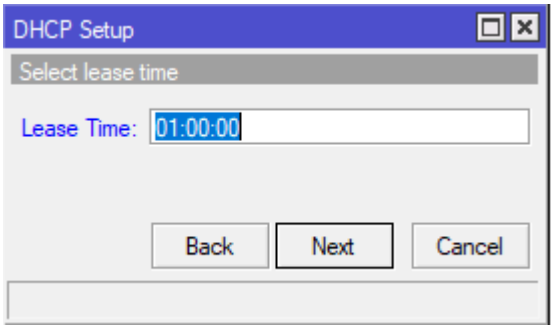


Figura 15. Configuración tiempo de arrendamiento

- Si se realizó los pasos correctamente aparecerá un mensaje de que el procedimiento se llevó a cabo correctamente como el de la figura 16.

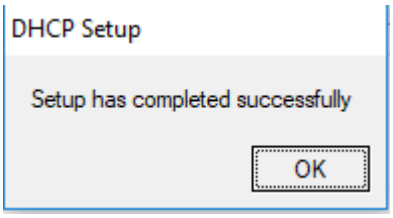


Figura 16. Mensaje de éxito de configuración de servidor DHCP.

1.5. Configuración NAT: Posteriormente para compartir acceso a internet se debe usar NAT, con el objetivo de enmascarar las direcciones IP de los equipos que se conecten a la LAN con la dirección IP de la red WAN.
Para esto se sigue los siguientes pasos:

- Click en **IP** luego en **Firewall,** aparece una nueva ventana, ya dentro le damos click en **+,** y seguimos los siguientes pasos

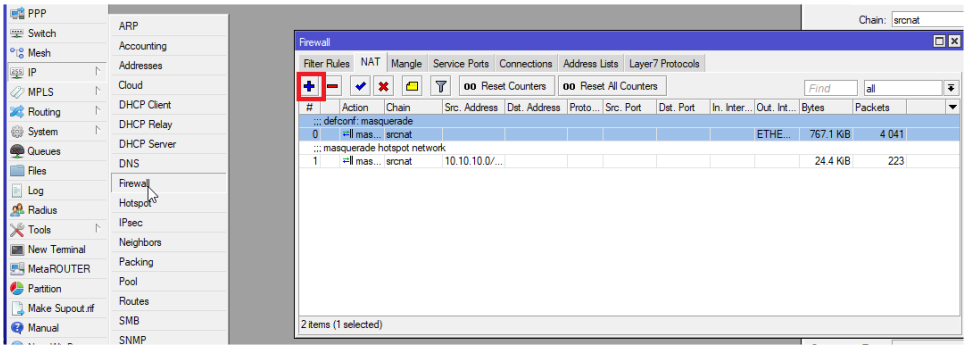


Figura 17. Configuración NAT desde el panel de Firewall.

- En el campo **Chain** seleccionamos **scrnat**, luego seleccionamos la interface de salida de internet, en el campo de **Out. Interface,** en este caso **ETHER1-WAN.**

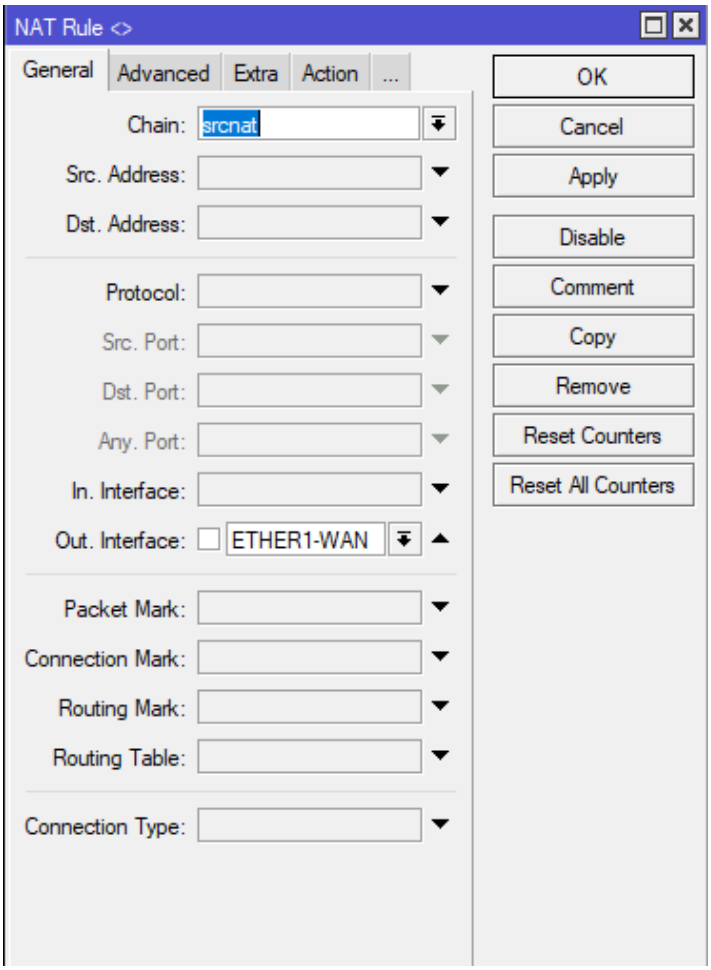


Figura 18. Panel general NAT Rule

- En el panel de **NAT Rule,** Navegamos hasta la pestaña Action, y en el campo **Action** seleccionamos **masquerade**.

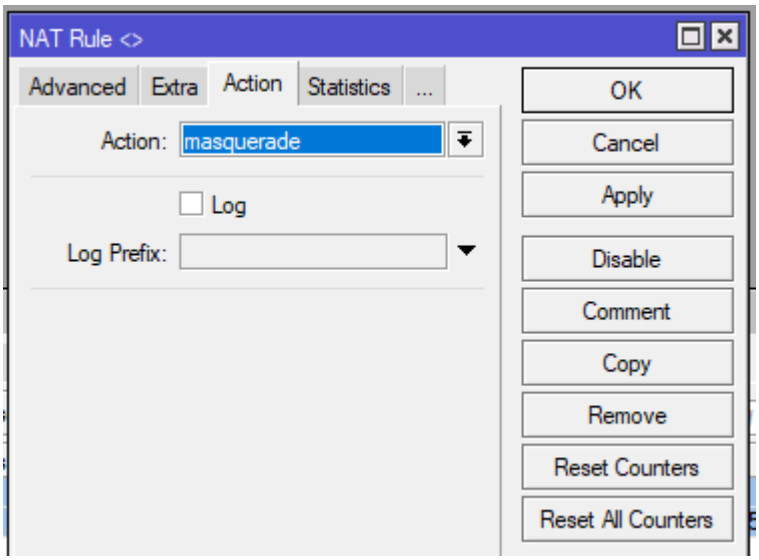


Figura 19. Pestaña action en el menú NAT Rule

1.6. Comprobación final: desde un computador nos conectamos mediante cable a la interfaz LAN y el equipo nos debe asignar IP y debemos tener salida a Internet. Esto se puede hacer navegando directamente en una página de internet o por consola ya sea por Windows Linux u cualquier sistema operativo. Cabe resaltar que la configuración indicada esta para una interfaz, sin embargo, si se quiere conectar más equipos existen muchas formas, como por ejemplo colocar un bridge, hacia los demás interfaces si el equipo de la marca Mikrotik cuenta con ellas, configurar una interfaz inalámbrica si el equipo cuenta con ella o conectar equipos de comunicación como access point o switch en el puerto configurado.

2. CONFIGURACIÓN HOTSPOT EN MIKROTIK

A continuación, se presenta una mini guía para la configuración de un Hotspot en equipos de la marca MIkrotik, este sistema nos permite capturar el tráfico http (web) de los clientes conectados a nuestra red y realizar un direccionarlo a un portal con fines de autenticación.

Para esto se sigue los siguientes pasos:

- Click en **IP** luego en **Hotspot,** aparece una nueva ventana, ya dentro le damos click en **Hotspot Setup,** y seguimos los siguientes pasos

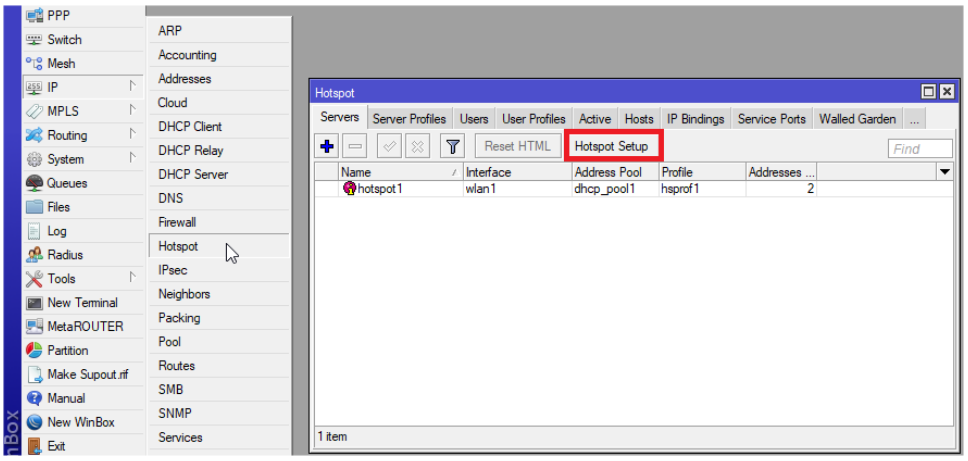


Figura 20. Panel principal Hotspot

- Seleccionamos la Interface

Figura 21. Selección interface Hotspot

- Seleccionamos la red, en la cual los clientes se van a autenticar.

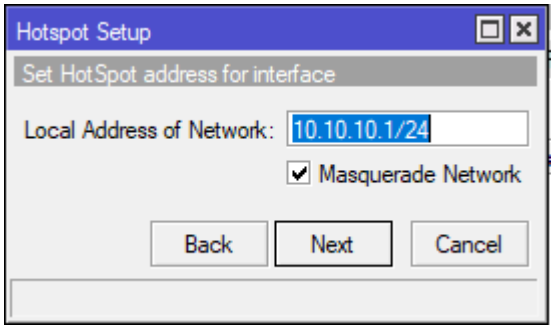


Figura 22. Selección de red Hotspot

- Seleccionar un rango deseado de direcciones IP que tenga el Hotspot

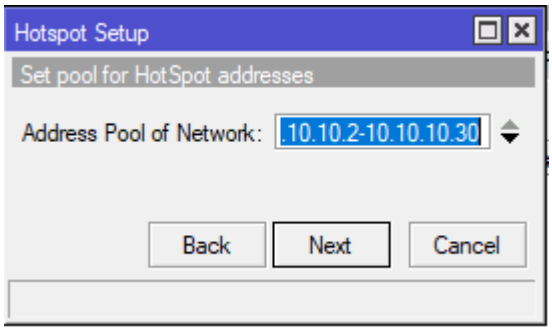


Figura 23. Selección de rango de direcciones IP Hotspot

- Se selecciona un certificado SSL si se tiene (opcional).

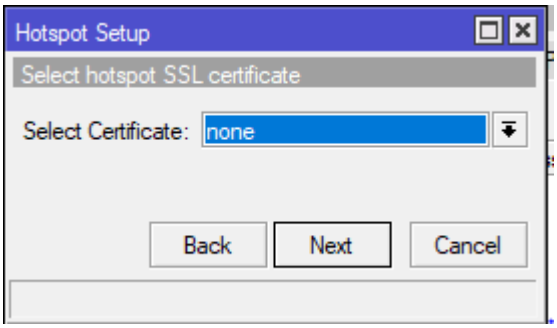


Figura 24. Certificado SSL Hotspot

- Se digita la dirección el servidor SMTP si se cuenta con este (opcional).

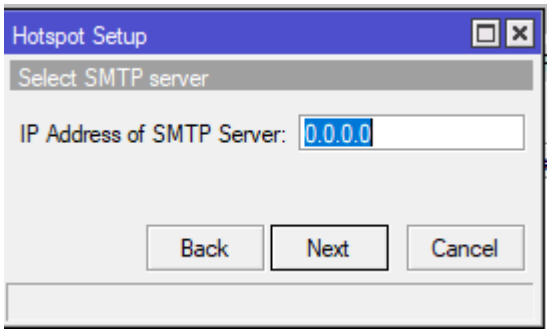


Figura 25. Servidor SMTP Hotspot

- Se configura los servidores DNS

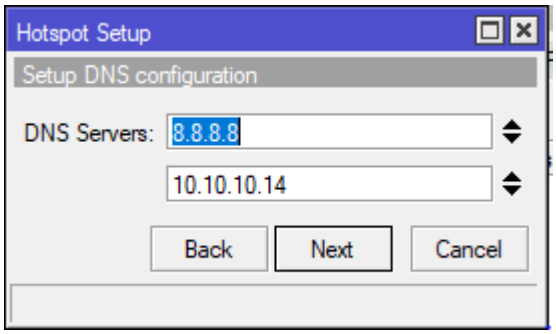


Figura 26. Selección servidor DNS Hotspot

- Se crea un nombre de perfil de usuario local (cualquiera) y se selecciona una contraseña.

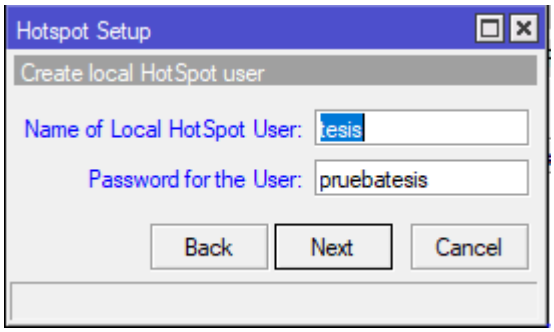


Figura 27. Creación de usuario y contraseña.

- Para permitir que mas usuarios se conecten desde el mismo perfil. Dar click en **users profiles,** luego en el campo **shared users** colocar el número de usuarios compartidos.

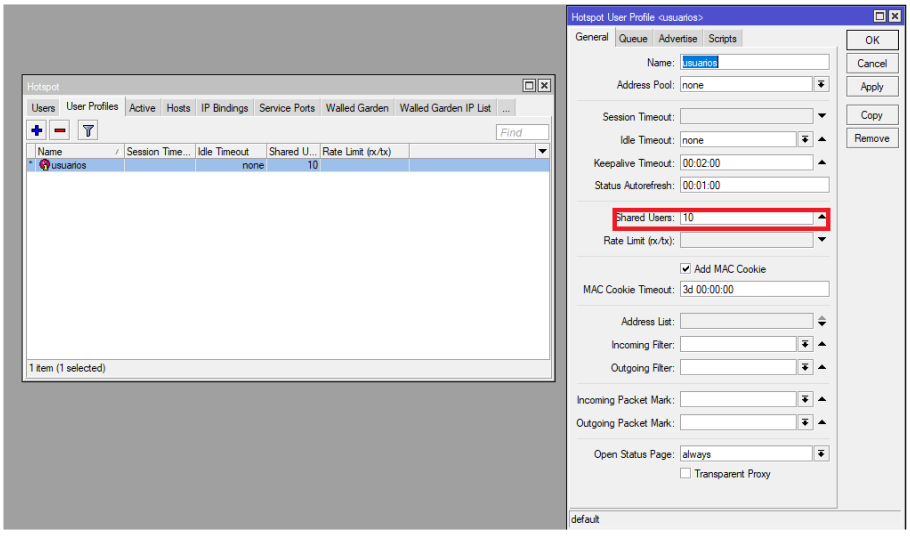


Figura 28. Perfiles de Usuario Hotspot

Con el portal cautivo implementado los clientes realizan la autenticación en pestañas como **active** podemos observar los clientes conectados y datos de navegación que se emplean en esta tesis.

3. CONFIGURACIÓN QUEUES EN MIKROTIK

A continuación, se presenta una mini guía para la configuración de **Simple Queues** en equipos de la marca MIkrotik, esta herramienta nos permite limitar la velocidad de descarga y carga para los clientes con direcciones IP. Para esto se sigue los siguientes pasos:

- Click en **QUEUES** luego en **+**

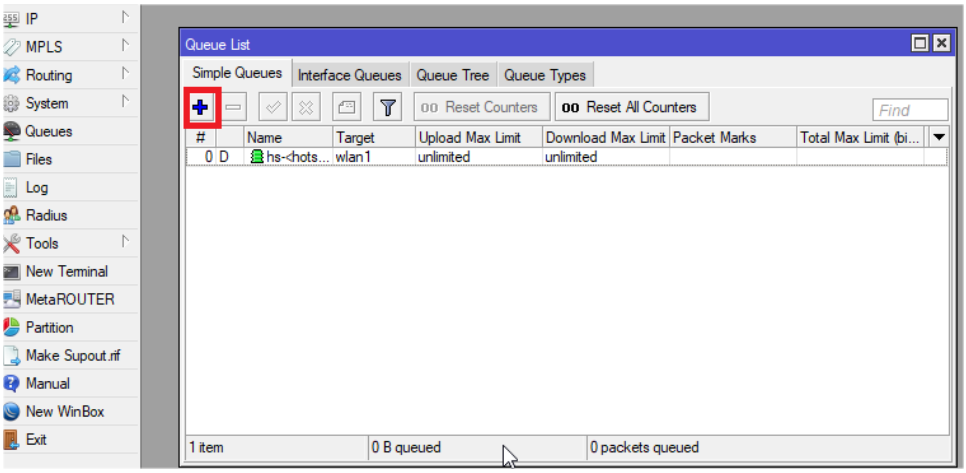


Figura 29. Panel principal Queues

- En el campo **name** colocamos el nombre deseado de la cola, en el campo **target** digitamos las direcciones IP y en los campos inferiores se puede limitar por ejemplo el límite de carga y de descarga,

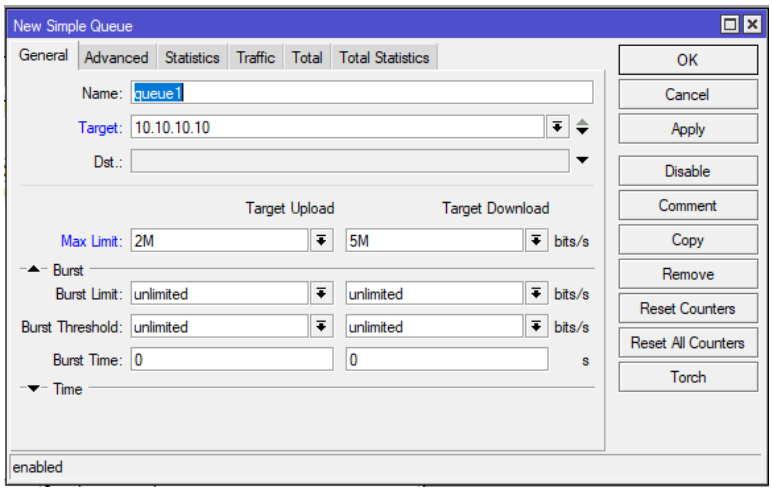


Figura 30. Parámetros de configuración simple Queue.