ISSN: 0798-1015

REVISTA ESPACIOS

Vol. 41 (29) 2020 • Art. 1

**Recibido**/Received: 02/01/2020 • **Aprobado**/Approved: 27/07/2020 • **Publicado**/Published: 06/08/2020

# Herramienta para el análisis de latencia y pérdida de paquetes en redes haciendo uso de algoritmos de clasificación soportados en árboles de decisión

## Tool for the analysis of latency and packet loss in networks using classification algorithms supported in decision trees

CHANCHÍ GOLONDRINO, Gabriel E. [1]
DURÁN DORADO, Diego F. [2]
CAMPO MUÑOZ, Wilmar Y. [3]

**Resumen**
A partir de la necesidad de monitorear los fallos que ocurren en los nodos de una topología de red, este artículo propone una herramienta para el análisis de latencia y pérdida de paquetes mediante algoritmos de clasificación soportados en árboles de decisión. La herramienta usa en segundo plano la librería MTR de linux para la captura de los datos y la API de weka para el análisis de los datos.
**Palabras clave:** algoritmos de clasificación, árboles de decisión, fallos de red, J48.

**Abstract**
Based on the need to monitor the failures that occur in the nodes of a network topology, in this article we propose a tool for the analysis of latency and packet loss using classification algorithms supported in decision trees. The tool uses in the background the linux MTR library for data capture and the weka API for data analysis.
**key words:** classification algorithms, decision trees, J48, network failures.

## 1. Introducción

Debido a que la seguridad y operatividad de sistemas tales como las redes telemáticas y sitios Web dependen de las características específicas del tráfico (por ejemplo, latencia y pérdida de paquetes), el análisis de éstas tiene una importante significancia teórica y práctica en lo referente al diseño y optimización del rendimiento de las redes, y en la fiabilidad y disponibilidad de los servicios desplegados en ellas (Murillo, 2010).

Al respecto, pruebas de diagnóstico de red son consideradas como procedimientos de evaluación requeridos para identificar problemas funcionales de la red (Ming & Hassan, 2015), (Peña, da Silva, Flebes, & Calderón,

[1] Profesor del Programa Ingeniería de Sistemas. Facultad de Ingeniería. Universidad de Cartagena. gchanchig@unicartagena.edu.co.
[2] Investigador adscrito al Grupo de Ingeniería Telemática. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca. dduran@unicauca.edu.co
[3] Profesor del Programa de Ingeniería Electrónica. Facultad de Ingeniería. Universidad del Quindío. wycampo@uniquindio.edu.co.

2018). Esto implica la necesidad de proporcionar herramientas que provean información útil sobre el estado de los sistemas telemáticos, apoyando así la identificación y solución de problemas. Herramientas y o librerías como MTR (*My Trace Route*) trazan caminos entre dos puntos de red, al tiempo que identifican problemas en algún nodo en términos de latencia y pérdida de paquetes. Con esta información, planificadores, analistas y administradores de red pueden tomar decisiones dirigidas a ofrecer Calidad del Servicio (Limoncelli, Hogan, & Chalup, 2007).

Por otra parte, modelos de clasificación del campo del aprendizaje automático han venido siendo explotados al interior de la comunidad investigativa para dotar de capacidad de razonamiento a diferentes sistemas. Por ejemplo, los modelos de clasificación basados en árboles de decisión se han aplicado en diferentes estudios así: en (Limoncelli, Hogan, & Chalup, 2007) se aplican en sistemas de gestión de aprendizaje para descubrir patrones de conocimiento sobre bases de datos que registran el comportamiento de estudiantes; en (Al Nasseri , Tucker , & de Cesare, 2015) se introducen en un sistema inteligente para analizar y extraer la semántica de términos que expresan sentimientos de un blog con el fin de determinar si se recomienda la compra de un artículo específico; en (Schetinin, y otros, 2007) se aplican sobre conjuntos de datos de lesiones traumáticas para identificar relaciones entre variables como la edad, el tipo de trauma y su severidad.

En concreto, los modelos de clasificación son utilizados para identificar automáticamente relaciones de dependencia entre variables de entrada (difícilmente identificables por procesos mecánicos) tales como la frecuencia de uso de cierto servicio de comercio electrónico, las calificaciones alcanzadas en cierto curso en línea y el estado de ánimo.Precisamente, en el presente artículo se prepone una herramienta útil en el diagnóstico de fallos de red, la cual introduce las ventajas ofrecidas por los modelos de clasificación basados en árboles de decisión para determinar probabilísticamente relaciones de dependencia entre variables tales como la latencia, pérdida de paquetes y la homogeneidad de los datos a partir del análisis de red ofrecido en segundo plano por la por herramienta y/o librería libre MTR[3]. La herramienta propuesta opera en el sistema operativo linux y fue escrita en el lenguaje Java haciendo uso de la API de weka para el análisis de los datos capturados mediante modelos de clasificación basados en árboles de decisión. La herramienta se propone como un aporte para analistas y administradores de redes en la identificación de alternativas y en la toma de decisiones enfocadas en la solución de problemas de red, basadas en la pérdida de paquetes y la latencia.

El artículo está organizado de la siguiente manera: la Sección 2 describe la metodología de la presente investigación. La Sección 3 presenta un marco teórico que trata aspectos generales sobre MTR y árboles de decisión. En la sección 4 se describen un conjunto de trabajos relacionados que se tuvieron en cuenta para el desarrollo de la presente investigación. La Sección 5 describe la herramienta de diagnóstico propuesta, señalando aspectos de diseño, de implementación e indicando los modelos de clasificación utilizados para la puesta en marcha de los árboles de decisión. La Sección 6 describe un estudio de caso en el cual se señalan las prestaciones de la herramienta propuesta en el diagnóstico de red entre un *host* origen de red domiciliaria y un *host* destino que corresponde a un servidor Web. Finalmente, la Sección 7 presenta las conclusiones y esboza las líneas de trabajos futuros.

[3] MTR - https://debian.pkgs.org/9/debian-main-amd64/mtr_0.87-1_amd64.deb.html

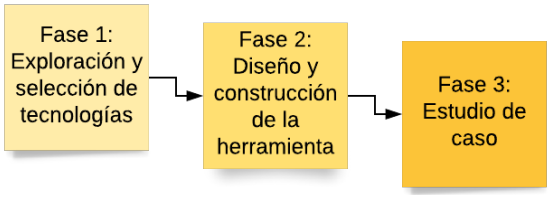
## 2. Metodología

Para el desarrollo de la presente investigación se tuvieron en cuenta tres fases metodológicas a saber: exploración y selección de tecnologías, diseño e implementación de la herramienta y estudio de caso (ver Figura 1).

### Fase 1 – Exploración y selección de tecnologías

En esta fase se realizó la exploración y selección de tecnologías más apropiadas para la construcción de la herramienta para el diagnóstico de redes propuesta en este artículo. Como resultado de esta fase, se seleccionó la herramienta libre MTR, la cual funciona desde la consola de Linux, integra en conjunto las funcionalidades de los comandos ping y traceroute, además de permitir la obtención de la latencia y el porcentaje de paquetes perdidos en los diferentes nodos o saltos que se encuentran entre un *host* origen y un *host* destino. De la misma manera, se escogió la librería de minería de datos Weka, dado que es una de las herramientas más difundidas en minería de datos, la cual permite la construcción de modelos de clasificación a partir de diferentes fuentes de datos (CSV, ARFF, bases de datos). De este modo la herramienta propuesta aprovecha los datos obtenidos mediante la herramienta MTR ejecutándose en segundo plano del sistema operativo linux, para generar información relevante a partir de los algoritmos de clasificación provistos por la librería de la herramienta Weka.

**Figura 1**
Metodología propuesta




Fuente: Autores

### Fase 2 – Diseño y construcción de la herramienta

A partir de las tecnologías identificadas en la fase 1 de la metodología, se construyó una herramienta para el diagnóstico de fallas de red en el lenguaje Java, la cual usa en segundo plano la librería MTR de Linux. La herramienta propuesta enriquece y extiende la funcionalidad de esta librería, al proveer una interfaz gráfica que permite a un administrador ejecutar pruebas y visualizar la latencia y el porcentaje de paquetes perdidos entre los nodos comprendidos entre un *host* origen y un *host* destino. De la misma manera a partir de los datos obtenidos en las pruebas realizadas en segundo plano con MTR, la herramienta propuesta permite aplicar algoritmos de clasificación sobre los datos capturados, los cuales posibilitan la obtención de relaciones entre los diferentes saltos, el porcentaje de pérdidas y la latencia de los diferentes nodos de red. Para el caso de la presente investigación, se escogió el algoritmo de árboles de decisión J48, teniendo en cuenta que los datos recolectados contienen elementos no categóricos. La información obtenida mediante el análisis pretende servir de guía a un administrador de red para la toma de decisiones en los diferentes nodos o saltos de red.

### Fase 3 – Estudio de caso

Como medio de verificación de las funcionalidades de la herramienta propuesta, en este artículo se realizaron pruebas de diagnóstico de red teniendo como origen un *host* de red domiciliaria en la ciudad de Popayán (Colombia) y como *host* destino el servidor web que almacena el portal de la Universidad de Cartagena (Colombia). Así mismo se hizo una prueba de diagnóstico de red tomando como origen un *host* de red domiciliaria en Cartagena y como *host* de destino el servidor que almacena el portal de la Universidad de Cartagena. Las

pruebas realizadas permitieron identificar datos relevantes con respecto a la latencia y el porcentaje de paquetes perdidos entre los dos nodos mencionados.

## 3. Marco teórico

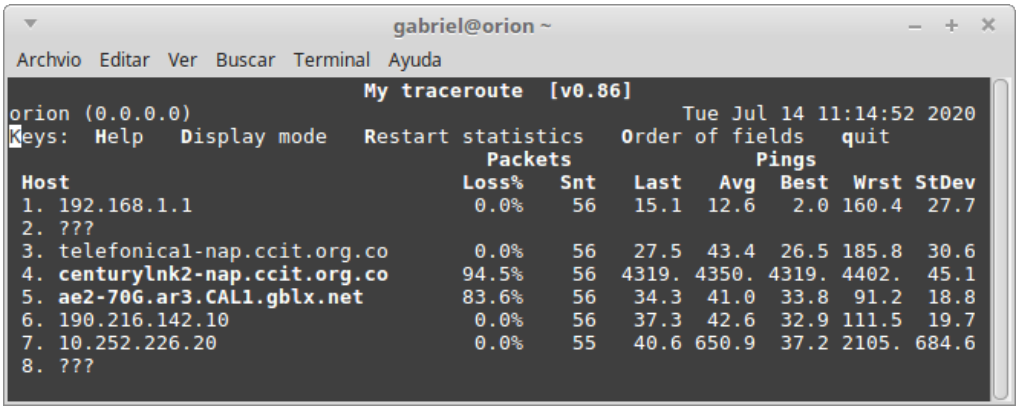
A continuación, se presenta un resumen de los aspectos teóricos en los cuales se enmarca el presente estudio.

### 3.1. MTR (My Trace Route)

MTR combina la funcionalidad de *Ping* y *Traceroute*, ofreciendo una herramienta robusta para propósitos de solución de fallos de red. Con la información ofrecida por MTR los analistas de red pueden determinar aspectos como la existencia de conectividad entre dos dispositivos y la calidad del enlace en términos de la pérdida de paquetes y la latencia (Wilson, 2019). Por defecto, MTR envía paquetes de solicitud de respuesta ICMP y usa mensajes de error de tiempo excedido para determinar los dispositivos en el enlace entre la fuente y el destino (Wilson, 2019). Tal como se aprecia en la Figura 2, la herramienta MTR permite la obtención de los siguientes datos entre dos nodos de red determinados: nombre del salto (Host), porcentaje de perdida de paquetes en cada salto (Loss%), el tiempo de ejecución del comando (Snt), el último valor de latencia (Last), la latencia promedio de cada salto (Avg), la mejor latencia (Best), la peor latencia (Wrst) y la desviación estándar de la latencia (Std).

Se ha escogido el uso de la herramienta MTR en el presente artículo, dado que es una herramienta o librería de software libre que se ejecuta sobre la consola de Linux, la cual permite integrar las funcionalidades de los comandos de red ping y traceroute en uno solo, obteniendo además información básica sobre la latencia y pérdida de paquetes de los diferentes nodos de red, la cual puede ser enriquecida a partir del uso de algoritmos de árboles de decisión. La elección de la librería MTR de Linux da la posibilidad de que pueda ser ejecutada de manera portable en plataformas tipos SBC como Raspberry PI.

**Figura 2**
Herramienta MTR




Fuente: Autores

### 3.2. Árboles de decisión

Un árbol de decisión es un método analítico que a través de una representación esquemática de las alternativas disponible facilita la toma de mejores decisiones, especialmente cuando existen riesgos, costos, beneficios y múltiples opciones. El nombre se deriva de la apariencia del modelo parecido a un árbol y su uso es amplio en el ámbito de la toma de decisiones bajo incertidumbre junto a otras herramientas como el Análisis del Punto de Equilibrio. Actualmente gozan de gran popularidad en el manejo de grandes volúmenes de datos dadas las posibilidades que brinda y la facilidad con que son comprendidos sus datos por cualquier usuario. En síntesis, un árbol de decisión permite: 1) segmentar: establecer qué grupos son importantes para clasificar cierto ítem; 2) clasificar: asignar ítems a uno de los grupos en que está particionada una población; 3) Predecir: establecer reglas

para hacer predicciones de ciertos eventos; 4) reducir la dimensión de los datos: identificar qué datos son importantes para modelar un fenómeno; 5) identificar relaciones: identificar qué variables y relaciones son importantes para ciertos grupos a partir del análisis de datos; y 6) recodificar: discretizar variables o establecer criterios cualitativos perdiendo la menor cantidad de información relevante (Bouza & Santiago, 2012) (Lino, Rocha, Macedo, & Sizo, 2019).

## 4. Trabajos relacionados

Se ha evidenciado la existencia de diferentes trabajos de la literatura en donde se enfoca el diagnóstico de fallos de red a partir del uso de redes Bayesianas. En (Liang, Liu, Qu, & Zhang, 2019) se presenta un método de alarma basado en inferencia bayesiana con el propósito de estimar el estado de la red. Por su parte, en (Gosselin, Courant, Romaric, & Vaton, 2017) se propone un motor de inferencia bayesiano aplicado en el diagnóstico de fallos en redes ópticas. Del mismo modo en (Carrera, 2010) se propone un sistema de diagnóstico distribuido basado en una arquitectura multi-agente soportada en redes bayesianas, con el fin de obtener las causas más probables de un fallo de red. De otra parte, en (Garcia-Algarra, González-Ordás, Arozarena, Afonso, & Carrera, 2014) se desarrolló un sistema mulitagente denominado fitIT basado en redes bayesianas para la detección de fallos en redes de fibra optica y la reparación de los mismos. En general, las redes bayesianas se presentan como un enfoque de diagnóstico muy poderoso, pero cuya complejidad en la capacidad de inferencia se incrementa exponencialmente con el número de nodos y dependencias entre ellos. Como una alternativa al uso de este enfoque en la detección de fallos, en el presente artículo se usan los árboles de decisión para determinar probabilísticamente modelos de clasificación con relaciones de dependencia entre variables de red para predecir su comportamiento. A diferencia de las redes bayesianas, con esta propuesta se busca determinar modelos de clasificación aplicables a cualquier red sin importar su topología. De hecho, la complejidad en el diagnóstico de fallos de red es un problema que ha venido siendo abordada en trabajos como el presentado en (Bennacer, Amirat, Chibani, Mellouk, & Ciabaglia, 2015), el cual combina de manera híbrida el razonamiento basado en casos sobre las redes bayesianas para reducir la complejidad de la inferencia.

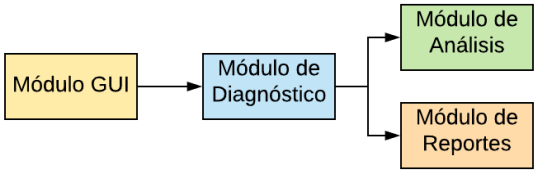
## 5. Herramienta propuesta

En esta sección se presenta el diseño y la implementación de la herramienta de diagnóstico de fallos de red propuesta, basada en el análisis de latencia y pérdida de paquetes. En cuanto al diseño se presenta un diagrama de bloques con los diferentes módulos funcionales y un diagrama de flujo que describe el proceso realizado por la herramienta de diagnóstico de fallos de red. En lo referente a la implementación, se presenta el diagrama de clases de la herramienta y su interfaz final con las funcionalidades de sus pestañas.

### 5.1. Diseño de la herramienta

En la Figura 3 se muestra el diagrama de bloques de la herramienta de diagnóstico de fallos propuesta, dentro de la cual se destacan los siguientes módulos funcionales:

**Figura 3**
Diagrama de bloques

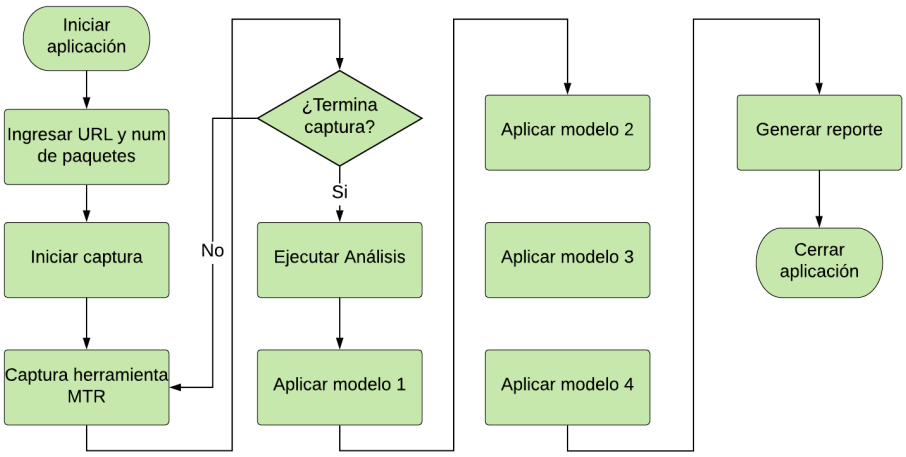



Fuente: Autores

El módulo GUI se encarga del control y la gestión de la interfaz gráfica de la herramienta de diagnóstico de fallos propuesta, para lo cual se hace uso de las clases pertenecientes a la librería *swing* de Java. El módulo de diagnóstico de fallos permite la ejecución de pruebas de red entre *host* determinados, identificando el porcentaje de paquetes perdidos y la latencia presente en los saltos de red que hacen parte de la ruta entre el *host* origen y el *host* destino. Para la ejecución de estas pruebas de red, la herramienta propuesta hace uso en segundo plano de la librería MTR, la cual combina desde la consola de Linux los comandos convencionales de *ping* y *tracerout*, de tal modo que mediante el uso del protocolo ICMP determina el porcentaje de paquetes perdidos y la latencia presente en los diferentes saltos de red entre dos *host*. El módulo de análisis por su parte se encarga de capturar los datos obtenidos mediante la herramienta MTR para aplicar mediante la librería Weka, diferentes modelos de clasificación basados en el algoritmo de árboles de decisión J48, mediante los cuales es posible obtener una relación clara entre atributos de los modelos de clasificación tales como: el porcentaje o nivel de paquetes perdidos, la latencia y los diferentes saltos de red. Finalmente, el módulo de reportes permite la generación de un documento PDF con los resultados de los diferentes modelos de clasificación aplicados a los datos obtenidos mediante el uso en segundo plano de la librería MTR. Este reporte es generado mediante el uso de la librería iText de Java.

A partir de lo descrito en el diagrama de bloques de la Figura 3, en la Figura 4 se presenta un diagrama de flujo que describe el proceso empleado por la herramienta propuesta para realizar el diagnóstico de fallos de red asociados a la latencia y la pérdida de paquetes.

**Figura 4**
Diagrama de flujo herramienta




Fuente: Autores

En primer lugar, se ingresa la URL del *host* destino y el número de paquetes a enviar vía ICMP en la interfaz principal de la herramienta. A partir de los datos anteriores, el administrador de red puede dar inicio a la captura de los datos, para lo cual la herramienta propuesta usa en segundo plano la librería MTR de Linux, cuyo objetivo es identificar la latencia y el porcentaje de paquetes perdidos en los diferentes saltos o nodos comprendidos entre el *host* origen y el *host* destino. Los datos capturados son presentados de manera continua en la interfaz gráfica de la herramienta, de tal modo que la captura puede ser interrumpida por el administrador de red en cualquier momento. Una vez el administrador de red interrumpe la captura de los datos, puede proceder con el análisis de los datos capturados haciendo uso de la librería para minería de datos Weka. Mediante esta librería, la herramienta implementa cuatro modelos de clasificación, basados en el algoritmo de clasificación de árboles de decisión J48. Dichos modelos permiten al administrador obtener relaciones entre los diferentes atributos del

clasificador, de tal modo que pueda tomar decisiones sobre los diferentes nodos desplegados en una topología de red.

## 5.2. Diseño de los modelos de clasificación

La herramienta propuesta en este artículo implementa 4 modelos de clasificación, los cuales hacen uso de los atributos definidos a partir de los datos proporcionados por la herramienta MTR. Estos modelos de clasificación son basados en el algoritmo de árboles de decisión J48. El objetivo de los cuatro modelos es permitir al administrador de redes contar con diferentes variantes en el análisis de los datos capturados sobre la red, dado que la mayoría de los modelos usan una diferente configuración de los atributos y el atributo clase, teniendo en cuenta variables semejantes (saltos, latencia, coeficiente de variación, etc.). A continuación, se describen los diferentes atributos de los cuatro modelos definidos.

**Tabla 1**
Modelo de clasificación 1

| Atrib 1 | Atrib 2 | Atrib 3 | Atrib 4 | Atrib 5 | Atrib 6 | Clase |
|---|---|---|---|---|---|---|
| Nivel de pérdidas | Última latencia | Latencia promedio | Mejor latencia | Peor latencia | %Coeficiente de Variación | Salto |
| Muy bajo<br>Bajo<br>Medio<br>Alto<br>Muy alto | Valor continuo | Valor continuo | Valor Continuo | Valor Continuo | Homogénea<br>Heterogénea | 1....n |

Fuente: Autores

En la Tabla 1 se presentan los atributos del modelo de clasificación 1, los cuales han sido adaptados a partir de los datos capturados de cada nodo o salto de la red por la librería MTR en segundo plano. De este modo, en lo que respecta al atributo 1 del clasificador, a partir del porcentaje de paquetes perdidos en cada nodo o salto de red se definió una escala discreta denominada nivel de pérdidas (muy bajo, bajo, medio, alto, muy alto). Los atributos 2, 3, 4 y 5 por su parte toman el valor continuo obtenido por la librería MTR para la latencia en cada nodo, teniendo en cuenta que este parámetro no tiene una escala definida. En lo que respecta al atributo 6, éste corresponde al porcentaje del coeficiente de variación. De este modo, el atributo 6 es igual a la relación entre la desviación estándar y la media de la latencia (coeficiente de variación), multiplicada por 100, donde un valor inferior o igual al 80% hace referencia a que la media no es representativa del conjunto de datos, mientras que un valor superior al 80% indica que la media representa al conjunto de datos. Finalmente, la clase del clasificador está asociada al número de salto o nodo donde se capturaron los valores de latencia y pérdida de paquetes, es decir es un valor discreto entre 1 y n.

**Tabla 2**
Modelo de clasificación 2

| Atrib 1 | Atrib 2 | Atrib 3 | Atrib 4 | Atrib 5 | Atrib 6 | Clase |
|---|---|---|---|---|---|---|
| Salto | Última latencia | Latencia promedio | Mejor latencia | Peor latencia | %Coeficiente de variación | Nivel de pérdidas |
| 1...n | Valor continuo | Valor continuo | Valor Continuo | Valor Continuo | Homogénea<br>Heterogénea | Muy bajo<br>Bajo<br>Medio<br>Alto<br>Muy alto |

Fuente: Autores

En la Tabla 2 se presentan los atributos del modelo de clasificación 2, los cuales fueron adaptados a partir de los datos capturados de cada nodo o salto de la red por la librería MTR en segundo plano. Tal como se aprecia en la tabla 2, a diferencia del modelo 1 se presentan cambios en el atributo 1 y en la clase, de tal manera que como atributo 1 se ha tomado el número de salto o nodo donde se capturan los valores de latencia y como clase se ha elegido el nivel discreto de pérdidas. La diferencia en la configuración de este modelo, permite al administrador de la red tener otra perspectiva del análisis de los datos capturados a partir de la red.

**Tabla 3**
Modelo de clasificación 3

| Atrib 1 | Atrib 2 | Atrib 3 | Atrib 4 | Atrib 5 | Atrib 6 | Clase |
|---|---|---|---|---|---|---|
| Salto | Nivel de pérdidas | Última latencia | Latencia promedio | Mejor latencia | Peor latencia | %Coeficiente de Variación |
| 1...n | Muy bajo<br>Bajo<br>Medio<br>Alto<br>Muy alto | Valor continuo | Valor continuo | Valor continuo | Valor continuo | Homogénea<br>Heterogénea |

Fuente: Autores

En la tabla 3 se presentan los atributos del modelo de clasificación 3, los cuales fueron adaptados a partir de los datos capturados de cada nodo o salto de la red por la librería MTR en segundo plano. Tal como se aprecia en la tabla 3, a diferencia del modelo 1 y 2 se presentan cambios en la clase, la cual corresponde en este caso al porcentaje de variación de los valores de latencia. Por su parte, los demás atributos corresponden a los atributos restantes asociados al número de salto, el nivel de pérdida de paquetes y a los valores de latencia en cada uno de los nodos o saltos.

**Tabla 4**
Modelo de clasificación 4

| Atrib 1 | Atrib 2 | Atrib 3 | Atrib 4 | Atrib 5 | Clase |
|---|---|---|---|---|---|
| Salto | Última latencia | Latencia promedio | Rango medio | %Coeficiente de Variación | Nivel de pérdidas |
| 1...n | Valor continuo | Valor continuo | Valor Continuo | Homogénea<br>Heterogénea | Muy bajo<br>Bajo<br>Medio<br>Alto<br>Muy alto |

Fuente: Autores

En la tabla 4 se presentan los atributos del modelo de clasificación 4, los cuales son una adaptación del modelo de clasificación 2. A diferencia del modelo de clasificación 2, el modelo de clasificación 4 contiene 5 atributos y una clase, de tal modo que los atributos 4 y 5 del modelo de clasificación 2 fueron unificados en un solo atributo denominado rango medio. El rango medio es entendido como la diferencia entre el máximo y el mínimo valor de la latencia entre 2. Este modelo fue considerado a partir de la necesidad de precisar el modelo de clasificación 2.

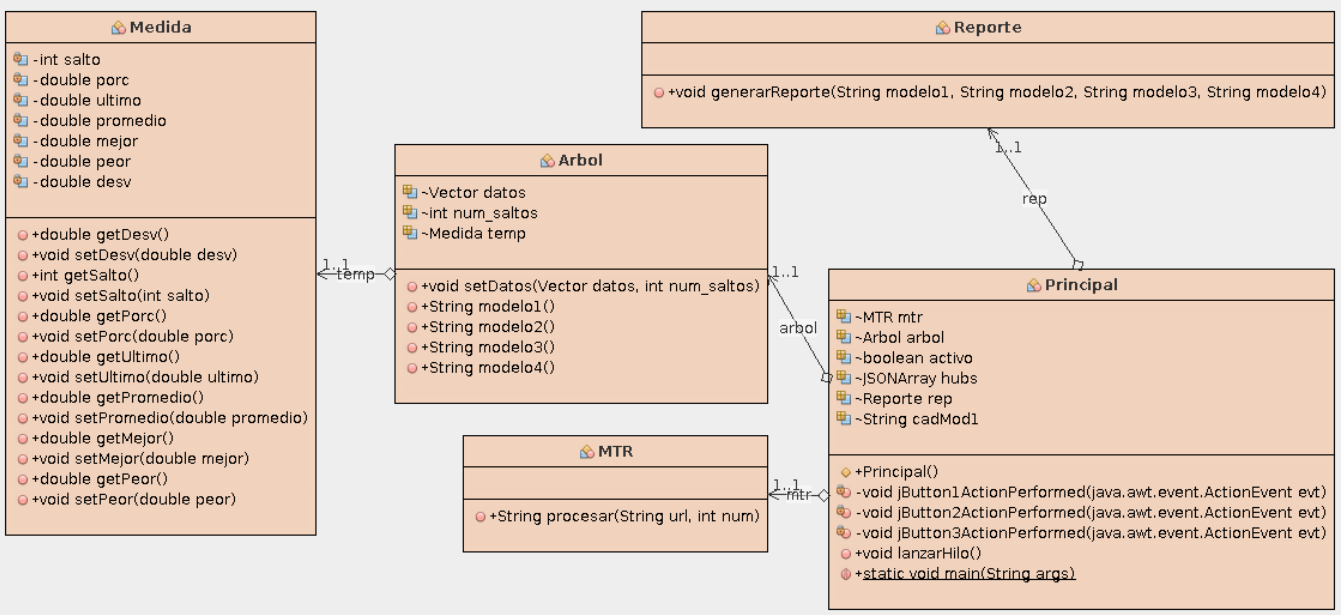
## 5.3. Implementación de la herramienta

A partir de la funcionalidad descrita en la sección 4.3, en esta sección se presenta la implementación de la herramienta para el diagnóstico de fallos en la red. De este modo en la Figura 5 se presenta el diagrama de clases de la herramienta de diagnóstico de fallos propuesta, en la cual se destacan las clases: *MTR*, *Arbol* y *Reporte*. La clase MTR es la encargada a través de sus métodos de ejecutar y procesar los datos capturados en segundo plano realizadas mediante la librería MTR. La clase *Arbol* es la encargada a través de sus métodos de configurar y

ejecutar los cuatro modelos de clasificación basados en el algoritmo de árboles de decisión J48 definidos en la sección 5.2. La clase Reporte por su parte es la encargada de generar los reportes en PDF con los resultados de la aplicación de los cuatro modelos de clasificación a los datos capturados en segundo plano con la herramienta MTR. Finalmente, la clase Principal es la encargada de gestionar la interfaz gráfica de la herramienta para el diagnóstico de fallas de red.

Para que la herramienta propuesta pueda realizar el análisis de los datos capturados a partir de la red, así como generar el reporte de los análisis realizados, se articularon las librerías weka y itextpdf respectivamente. La herramienta weka permite así potenciar las ventajas provistas por la herramienta MTR, la cual permite desde la consola de linux la captura de los datos de pérdida de paquetes y latencia entre dos *host* de red, pero no posibilita el análisis de los datos capturados.

**Figura 5**
Diagrama de clases de la herramienta




Fuente: Autores

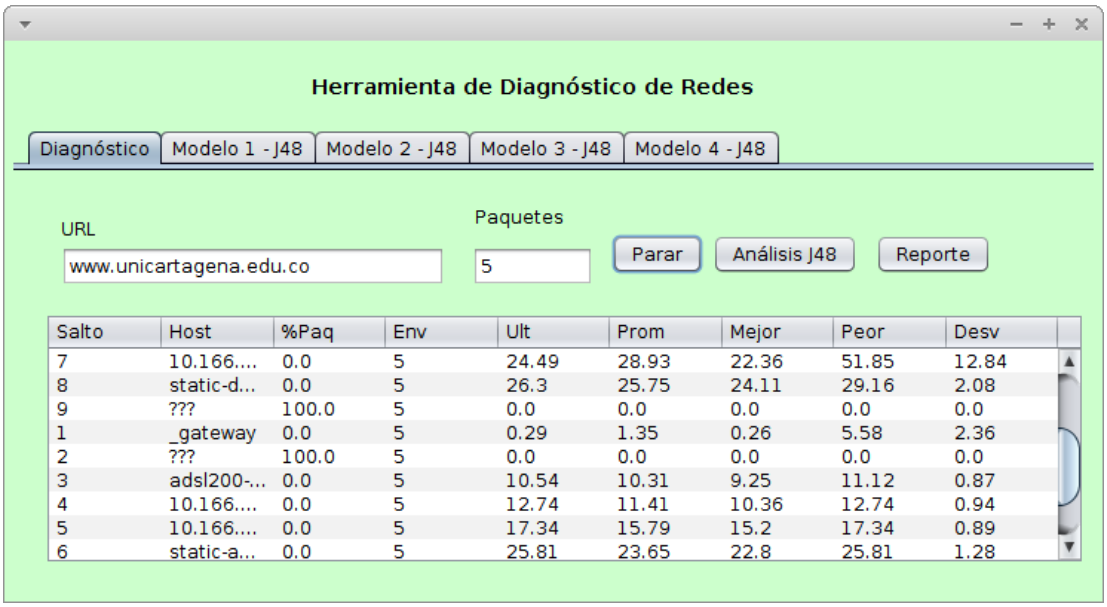
En la Figura 6 se presenta la interfaz principal de la herramienta desarrollada en el lenguaje de programación Java haciendo uso de las clases pertenecientes al paquete *Swing*. Tal como se aprecia en la Figura 6, la interfaz de la herramienta la cual está conformada por cinco pestañas: *Diagnóstico*, *Modelo 1 - J48*, *Modelo 2 - J48*, *Modelo 3 - J48* y *Modelo 4 - J48*.

La pestaña *Diagnóstico* contiene dos campos de textos asociados a la URL del *host* destino y al número de paquetes ICMP a enviar a los diferentes nodos de red comprendidos entre el *host* origen y el *host* destino. Una vez la URL y el número de paquetes han sido ingresados, el administrador de red puede dar inicio al diagnóstico presionando el botón *Iniciar*, el cual permite la conformación de un comando (con la URL y el número de paquetes) que será ejecutado en segundo plano haciendo uso de la herramienta MTR de Linux. Para permitir la ejecución en segundo plano de la herramienta MTR, se hace uso de las clases *Runtime* y *Process* de Java, las cuales hacen posible la ejecución de comandos del sistema operativo, así como la obtención de la salida o el resultado de aplicación de dichos comandos. Esta salida está configurada a nivel de los comandos para ser obtenida mediante el formato JSON.

Los resultados obtenidos por la herramienta MTR para cada nodo de la ruta son presentados en una tabla con nueve columnas a saber: el número de salto o nodo, el identificador del nodo o *host*, el porcentaje de paquetes

perdidos, el número de paquetes enviados, el valor de la última latencia del salto en cuestión, el promedio de la latencia del salto, el mejor valor de latencia del salto, el peor valor de latencia del salto y finalmente la desviación estándar de la latencia del salto. Cabe resaltar que estos nueve datos son obtenidos para cada uno de los nodos o saltos comprendidos entre un *host* origen y un *host* destino. Una vez se han enviado el número determinado de paquetes a los diferentes nodos el proceso se repite, de tal modo que la tabla presentada en la interfaz gráfica sigue creciendo, presentando y apilando nuevos datos para los diferentes saltos.

**Figura 6**
Interfaz principal de la herramienta




Fuente: Autores

A modo de ejemplo, en la Figura 5 se ha escogido como destino el servidor que almacena el portal web de la Universidad de Cartagena (Colombia) y se han seleccionado 5 paquetes ICMP para la prueba de red. Tal como se aprecia en la Figura 5, entre el *host* origen y el *host* de destino se distinguen un total de 9 saltos o nodos, de los cuales una vez obtenida los valores de porcentaje de paquetes perdidos y latencia, se procede automáticamente a repetir la prueba con 5 paquetes ICMP obteniendo y apilando nuevos valores hasta que el administrador de red decida detenerla.

Una vez el administrador de red decide parar la captura de los datos de pérdida de paquetes y latencia de los nodos comprendidos entre un *host* origen y un *host* destino, es posible presionar el botón *Análisis J48*, el cual permite la conformación de los datos de los modelos de clasificación y la ejecución de los mismos con los datos cargados haciendo uso de la librería Weka. De este modo, de la pestaña 2 a la pestaña 5 se presentan los árboles resultantes de aplicar los cuatro modelos de clasificación basados en árboles de decisión y descritos en la sección 5.2. Estos modelos de clasificación son conformados y ejecutados mediante el uso de la librería de minería de datos Weka. Así, en la Figura 7 se muestra el árbol de decisión obtenido al aplicar el modelo de clasificación 1 a los datos capturados con ayuda de la herramienta MTR.

Finalmente, una vez generados los árboles de decisión asociados a cada uno de los cuatro modelos de clasificación, la herramienta permite la generación de un reporte con los resultados obtenidos en cada modelo. Para la generación del reporte, la herramienta propuesta hace uso de la API iText de Java, la cual permite obtener un archivo PDF con la información correspondiente.

**Figura 7**
Árbol modelo de clasificación 1


Fuente: Autores

## 6. Estudio de caso

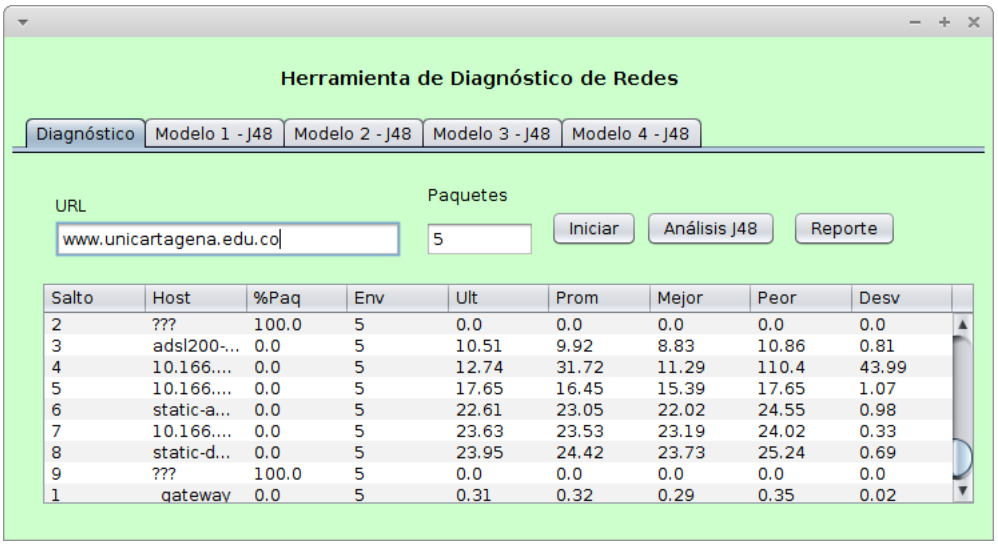
A modo de estudio de caso, se realizaron dos pruebas mediante la herramienta propuesta en el presente artículo. En la primera prueba se realizó el diagnóstico de red entre un *host* origen de red domiciliaria en la ciudad de Popayán (Colombia) y un *host* destino correspondiente al servidor web que almacena el portal de la Universidad de Cartagena (Colombia). De otra parte, la segunda prueba fue realizada entre un *host* de origen de red domiciliaria en la ciudad de Cartagena (Colombia) y un *host* destino perteneciente al servidor web que aloja el portal de la Universidad de Cartagena.

La primera prueba fue realizada a partir de los datos capturados por la herramienta de diagnóstico de fallos de red durante 15 minutos, realizando el envío continuo de 5 paquetes ICMP a los diferentes saltos comprendidos entre el *host* origen y el *host* destino. De este modo, se obtuvieron un total de 801 datos, lo que corresponde a 89 datos por cada uno de los 9 saltos existentes entre el *host* origen y el *host* destino (ver Figura 8). A partir de estos datos se conformó el *dataset* para la ejecución de los cuatro modelos de clasificación presentados en la sección 5.2.

A modo de ejemplo, en esta sección se presentan los datos arrojados por los modelos de clasificación 3 y 4, los cuales tienen como atributo clase la desviación estándar y el nivel de pérdidas respectivamente. Estos modelos fueron considerados en la presente sección teniendo en cuenta que presentan los árboles de decisión más sencillos de interpretar.

De acuerdo a los resultados mostrados en la Figura 9 y asociados al modelo de clasificación 3, algunas de las conclusiones que pueden obtenerse a partir del árbol generado por el algoritmo J48 son: cuando la peor latencia en los diferentes saltos está menor o igual a 62.71 milisegundos, la variación de la latencia es homogénea; cuando el nivel de pérdida de paquetes en los diferentes saltos de la red es muy alto, la variación de la latencia en dichos saltos tiende a ser clasificada como heterogénea; cuando la peor latencia de los saltos es mayor a 62.71, la mejor latencia es menor a 11.97 y la variación de la latencia es clasificada como heterogénea.

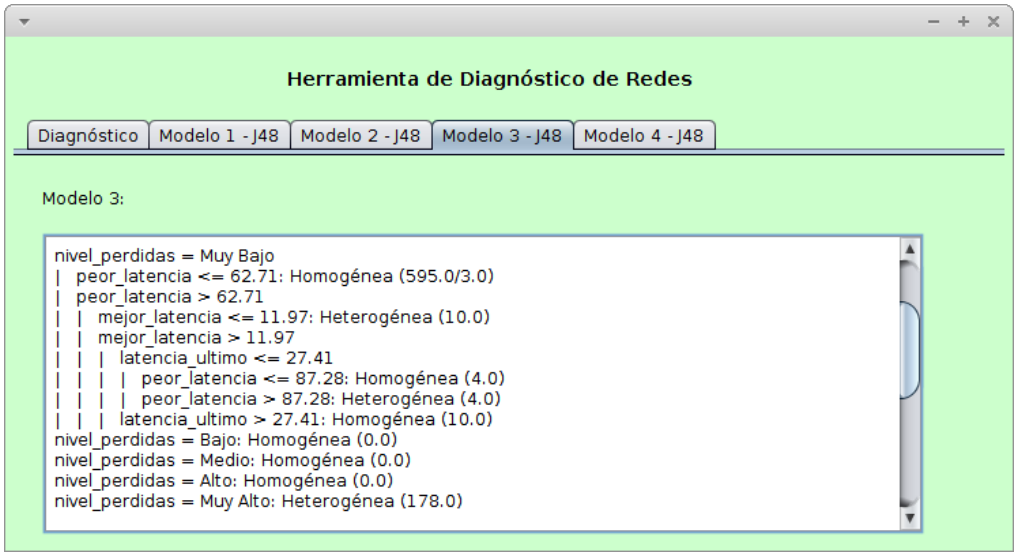
**Figura 8**
Estudio de caso prueba 1




Fuente: Autores

---

**Figura 9**
Modelo de clasificación 3 prueba 1



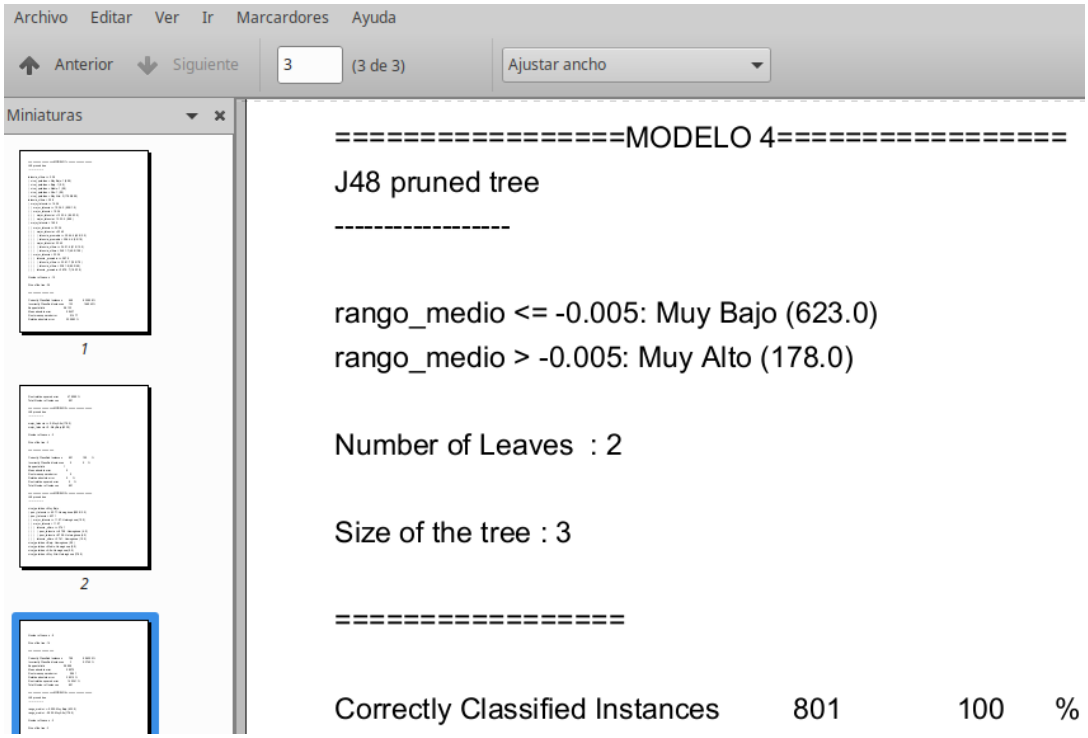

Fuente: Autores

Adicionalmente es importante mencionar que el modelo de clasificación 3 presenta un error medio absoluto de 0.0075 con un total de 798 instancias clasificadas de manera correcta sobre un total de 801. Lo anterior es obtenido a partir del reporte que genera la herramienta, el cual incluye el resultado del modelo, así como la evaluación del mismo, tomando como métrica el error medio absoluto.

Por su parte, en lo que respecta al modelo de clasificación 4, en la Figura 10 se presenta el árbol de decisión obtenido por este modelo y presentado en el reporte generado por la herramienta propuesta. De acuerdo al árbol de decisión de la Figura 10, cuando el rango medio de la latencia es más alto, es decir, cuando la diferencia entre la mejor latencia y la menor latencia es más alta, entonces el nivel de pérdida de paquetes tiende a ser muy alto. Del mismo modo, cuando el rango medio de la latencia es más bajo, es decir, cuando la diferencia entre

la mejor latencia y la peor latencia es más baja, entonces el nivel de pérdida de paquetes tiende a ser muy bajo. Dado que el modelo de clasificación 4 cuenta con un atributo menos que los otros modelos, más atributos discretos que los otros modelos y una clase con menos niveles de clasificación, este modelo presentó un total de 801 instancias correctamente clasificadas.

**Figura 10**
Modelo de clasificación 4 prueba 1

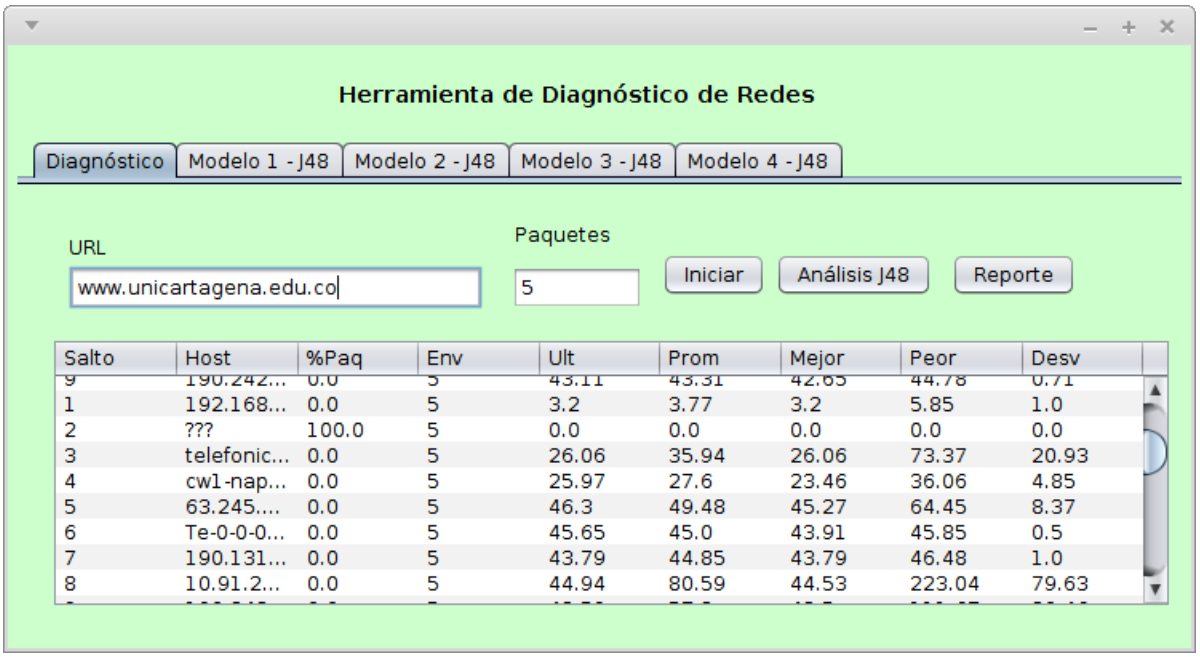



Fuente: Autores

La segunda prueba se realizó a partir de los datos capturados por la herramienta durante 15 minutos, realizando el envío continuo de 5 paquetes ICMP a los diferentes saltos del *host* origen y el *host* destino. De este modo, se obtuvieron un total de 495 instancias, lo que corresponde a 55 capturas por cada uno de los 9 saltos existentes entre el *host* origen y el *host* destino (ver Figura 11). A partir de estos datos se conformó el *dataset* para la ejecución de los cuatro modelos de clasificación presentados en la sección 5.2.

Del mismo modo, en la Figura 12 se presentan los resultados generados por la herramienta en cuanto a la aplicación del modelo de clasificación 3 a los datos capturados.

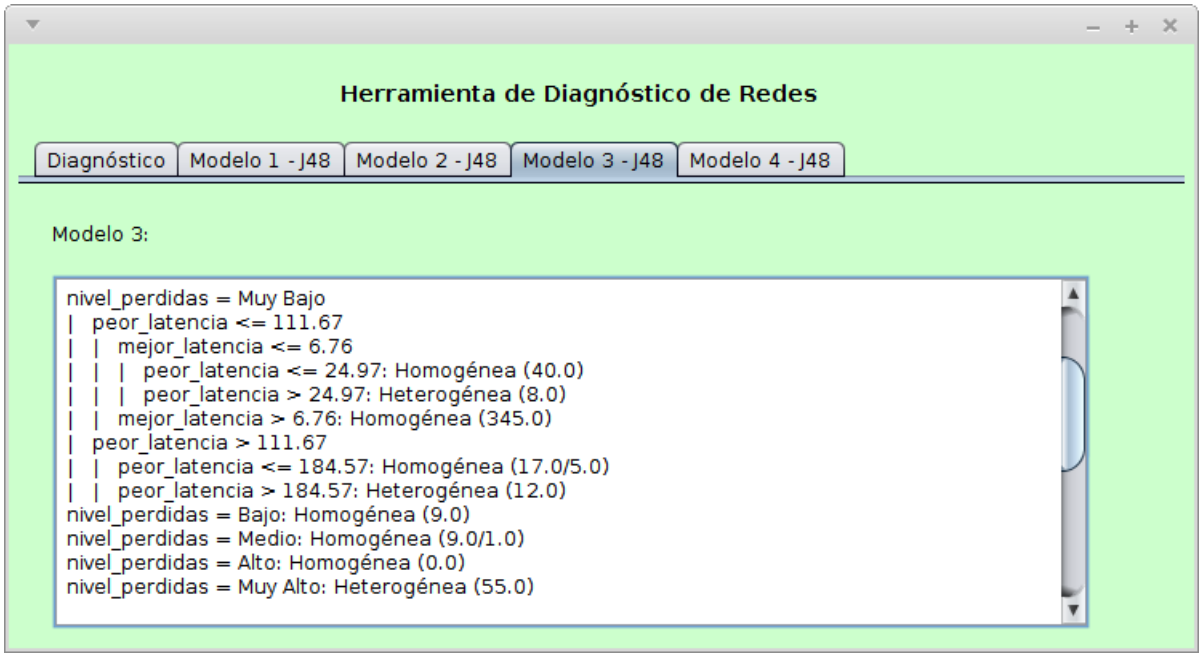
**Figura 11**
Estudio de caso prueba 2




Fuente: Autores

---

**Figura 12**
Modelo de clasificación 3 prueba 2

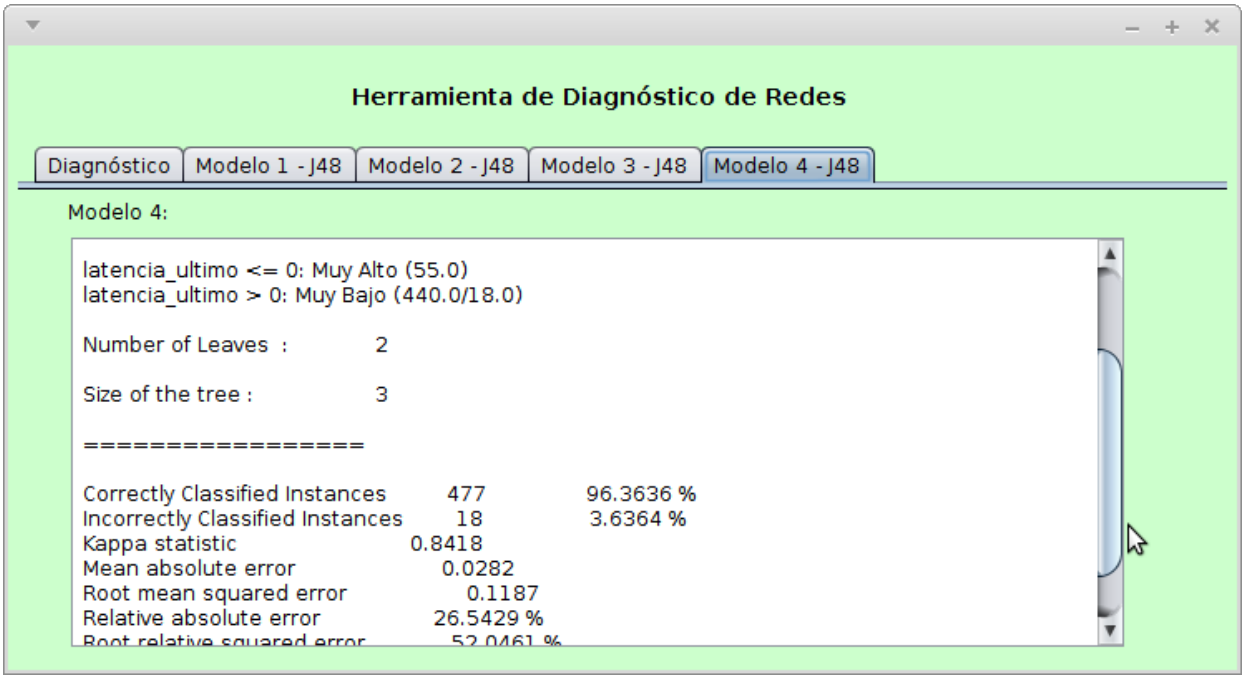



Fuente: Autores

Dentro de las conclusiones que se pueden obtener a partir del modelo de clasificación 3 están: cuando el nivel de pérdida de paquetes es muy alto, la variación de la latencia es heterogénea. Así mismo cuando el nivel de pérdida de paquetes es bajo o medio la variación de la latencia tiene a ser homogénea en la mayoría de los casos. De la misma manera, otra de las conclusiones que puede obtenerse es que cuando el nivel de pérdidas es muy bajo y la peor latencia es mayor a 184.57, la variación de la latencia es heterogénea. Adicionalmente los reportes

generados por la herramienta permiten obtener que el modelo de clasificación 3 tiene un error medio absoluto de 0.0179 con un total de 489 instancias clasificadas de 495 instancias en total. De la misma manera como se hizo en la prueba 1, en la Figura 13 se presentan los resultados obtenidos mediante la aplicación del modelo 4 a los datos capturados en la prueba 2. De acuerdo a la Figura 12 cuando el valor de la latencia es mayor a cero se obtiene una tendencia marcada al nivel de pérdidas muy bajo. Del mismo modo, cuando la latencia es 0 el porcentaje de pérdidas es clasificado como muy alto. El error medio absoluto para este modelo es de 0.0282 y el número de instancias clasificadas correctamente es de 477 sobre 495.

**Figura 13**
Modelo de clasificación 4 prueba 2




Fuente: Autores

## 7. Conclusiones y trabajos futuros

En este trabajo se propuso como aporte el desarrollo de una herramienta para la ejecución de pruebas de diagnóstico de red entre los nodos comprendidos entre un *host* origen y un *host* destino, la cual hace uso en segundo plano de la librería MTR de Linux y tiene como valor agregado el uso de algoritmos de árboles de decisión J48, los cuales permiten la obtención de información relevante sobre los datos de latencia y pérdida de paquetes capturados en los diferentes saltos o nodos de red. Así, esta herramienta pretende servir de apoyo a administradores y planificadores de red en cuando a la identificación de algunos tipos fallas en los nodos de una topología de red determinada, teniendo en cuenta la latencia y la pérdida de paquetes identificadas a partir del protocolo ICMP.

Para el desarrollo de la herramienta propuesta para el análisis de pérdida de paquetes y latencia, mediante el uso de algoritmos de clasificación soportados en árboles de decisión, se seleccionaron las herramientas libres más adecuadas para dar soporte a los procesos de la captura de los datos y su posterior análisis. En este sentido la librería libre MTR, la librería de minería de datos provista por la herramienta Weka y la librería para la generación de reportes iText, demostraron ser adecuadas para satisfacer los requisitos inicialmente planteados.

El principal aporte de la herramienta es la generación y conformación de cuatro modelos de clasificación especificados en la sección 5.2, los cuales a partir del algoritmo de árboles de decisión J48 permiten obtener una

relación entre parámetros arrojados por la herramienta MTR en segundo plano tales como: la latencia, el nivel de paquetes perdidos, la desviación estándar de la latencia y el número de salto o el nodo de red en el que dichas medidas son tomadas. Esto permite que el administrador o planificador de red tenga una visión más clara sobre los posibles fallos que pueden presentarse en la red, ante la gran cantidad de datos que puede arrojar la herramienta.

El uso de los modelos de clasificación basados en árboles de decisión J48 resultó ser adecuado para la naturaleza de los datos capturados en segundo plano mediante la librería MTR. En este sentido, parámetros como la latencia al no tener una escala establecida requiere ser tratada como un dato continuo, mientras que otros parámetros como la pérdida de paquetes puede ser adaptado a un dato de tipo discreto. De acuerdo a lo anterior, los árboles de decisión a diferencia de otros algoritmos de clasificación tienen como ventaja la facilidad para trabajar con ambos tipos de datos y obtener relaciones entre ellos.

La información arrojada por los modelos de clasificación considerados en la herramienta y basados en árboles de decisión, puede resultar relevante para los administradores de red no solo en la identificación de fallos de red, sino también de cara a garantizar la calidad del servicio (QoS) o la calidad de la experiencia (QoE) en escenarios de trasmisión de contenidos multimedia de alta calidad.

Como trabajo futuro derivado de la presente investigación, se propone la vinculación a la herramienta propuesta de otros algoritmos de clasificación y asociación tomando en cuenta los datos discretos de la captura, de tal modo que se pueda complementar el análisis obtenido mediante los árboles de decisión J48. Del mismo modo, es posible aplicar algoritmos de aprendizaje no supervisado que permitan agrupar los datos capturados de acuerdo a las variables de latencia y porcentaje de pérdida de paquetes.

## Referencias bibliográficas

Al Nasseri , A., Tucker , A., & de Cesare, S. (2015). Quantifying StockTwits semantic terms' trading behavior in financial markets: An effective application of decision tree algorithms. *Expert Systems with Applications, 42*(23), 9192-9210.

Bennacer, L., Amirat, Y., Chibani, A., Mellouk, A., & Ciabaglia, L. (2015). Self-Diagnosis Technique for Virtual Private Networks Combining Bayesian Networks and Case-Based Reasoning. *IEEE Transactions on Automation Science and Engineering, 12*(1), 354-366.

Bouza, C., & Santiago, A. (2012). La minería de datos: árboles de decisión y su aplicación en estudios médicos. En C. Bouza, *Modelación matemática de fenómenos del medio ambiente y la salud* (págs. 64-78). La Habana: Universidad de la Habana.

Carrera, A. (2010). *Uso de redes bayesianas para diagnosticar fallos inciertos de red.* Madrid: Universidad Politécnica de Madrid.

Garcia-Algarra, J., González-Ordás, J., Arozarena, P., Afonso, R., & Carrera, A. (2014). Un enfoque probabilistico en la autoreparación de redes G-PON. *Revista Iberoamericana de Automática e Informática Industrial*, 80-85.

Gosselin, S., Courant, J., Romaric, S., & Vaton, S. (2017). Application of probabilistic modeling and machine learning to the diagnosis of FTTH GPON networks. *International Conference on Optical Network Design and Modeling* , (págs. 1-3). Budapest- Hungría.

Liang, R., Liu, F., Qu, J., & Zhang, Z. (2019). A Bayesian-based Self-Diagnosis Approach for Alarm Prognosis in Communication Networks. *En 8th International Symposium on Next Generation Electronics*, (págs. 1-3). Zhengzhou, China.

Limoncelli, T., Hogan, C., & Chalup, S. (2007). *The Practice of System and Network Administration.* Boston: Addison-Wesley.

Lino, A., Rocha, Á., Macedo, L., & Sizo, A. (2019). Application of clustering-based decision tree approach in SQL query error database. *Future Generation Computer Systems, 93*, 392-406.

Ming, Z., & Hassan, A. (2015). A Survey on Load Testing of Large-Scale Software Systems. *IEEE Transactions on Software Engineering , 41*(11), 1091-1118.

Murillo, J. (2010). Mejoramiento de la latencia de la red mediante el cambio del tamaño de búfer para aplicaciones FTP utilizando el modelo para cliente/servidor según el tamaño promedio de los archivos a ser transmitidos . *Uniciencia*, 74-81.

Peña, M., da Silva, J., Flebes, O., & Calderón, C. (2018). Sistema para detección y aislamiento de fallas. *Revista Cubana de Ciencias Informáticas, 12*(2), 58-73.

Schetinin, V., Fieldsend, J., Partridge, D., Coats, T., Krzanowski, W., Everson, R., . . . Hernandez, A. (2007). Confident Interpretation of Bayesian Decision Tree Ensembles for Clinical Applications. *IEEE Transactions on Information Technology in Biomedicine , 11*(3), 312-319.

Tian, Z. (2019). Chaotic characteristic analysis of network traffic time series at different time scales. *Chaos, Solitons & Fractals, 130*, 1-16.

Wilson, M. (01 de 04 de 2019). *What is MTR & How to Use to Troubleshoot & Test your Connections*. Obtenido de https://www.pcwdld.com/what-is-mtr-and-howto-troubleshoot-connecti