



<https://creativecommons.org/licenses/by/4.0/>

ALGORITMOS SUPERVISADOS PARA LA PREDICCIÓN DEL ANCHO DE BANDA DE LAS APLICACIONES EN AMAZON WEB SERVICE DESDE UNA PYME RURAL

Supervised algorithms for application bandwidth prediction on Amazon Web Service from a rural SME¹

RAMIRO OSORIO DIAZ², MARTHA YANETH SEGURA RUIZ³, MAURICIO ALONSO VILLALBA⁴

Recibido:01 de mayo de 2023. Aceptado:13 de julio de 2023

DOI: <http://dx.doi.org/10.21017/rimci.2023.v10.n20.a138>

RESUMEN

Este artículo presenta una metodología para medir el comportamiento del ancho de banda realizando predicciones del tráfico de red que se conecta a la nube en empresas pequeñas y medianas de zonas rurales de difícil acceso en Colombia, con el fin de optimizar en el tiempo los recursos de red y asegurar la calidad del servicio en las aplicaciones web. Se realiza un estudio comparativo de tres algoritmos de redes neuronales que modelan una red neuronal multicapa, seleccionando el que tenga un error mínimo que se aproxime a cero, el algoritmo seleccionado se entrena a partir de una fuente de datos para predecir el tráfico de red que se conecta a la nube.

Es necesario analizar el comportamiento de la red para garantizar que las aplicaciones web que se encuentran en la nube transfiriendo información como datos, imágenes, sonidos, videos, entre otros, algunos en tiempo real y que generan alto tráfico, funcionen con calidad. Tener una visión general del tráfico que fluye a través de la red permite generar una planificación de la capacidad de la red cuando se administran recursos limitados como en el caso de pequeñas y medianas empresas de zonas rurales. Como producto del análisis de la investigación se desarrolla un prototipo de software libre que realiza las mediciones y predicciones en las áreas rurales. Los resultados de la implementación indican que el enfoque propuesto es superior a otros métodos de predicción en términos de precisión y predicción.

Palabras clave: Predicción; Pymes; Ancho de banda; Redes neuronales; Tráfico.

ABSTRACT

This article presents a methodology to measure the bandwidth behaviour by making predictions of the network traffic that connects to the cloud in small and medium enterprises in rural areas with difficult access in Colombia, in order to optimize network resources over time and ensure the quality of service in web applications. A comparative study of three neural network algorithms that model a multilayer neural network is performed, selecting the one that has a minimum error that approaches zero; the selected algorithm is trained from a data source to predict the network traffic that connects to the cloud.

- 1 Artículo producto del proyecto de investigación Predicción del tráfico de red de las aplicaciones web en la nube nativa basada en redes neuronales artificiales mediante software libre, ejecutado durante los años 2021 y 2022 desde el programa de Ingeniería en telecomunicaciones de Uniagustiniana, sede Bogotá.
- 2 Magister en telecomunicaciones móviles. Especialista en Redes de Alta Velocidad y Distribuidas. Ingeniero Civil. Docente Facultad de Ingeniería, Uniagustiniana. ORCID: <https://orcid.org/0000-0003-0007-9452> Correo electrónico: ramiro.osorio@uniagustiniana.edu.co
- 3 Magister en administración y dirección de empresas. Especialista en Diseño y Construcción de Soluciones Telemáticas. Especialista en Ingeniería de Software. Ingeniera de Sistemas. Docente Facultad de Ingeniería, Uniagustiniana. ORCID: <https://orcid.org/0000-0002-1572-1447> Correo electrónico: martha.segura@uniagustiniana.edu.co
- 4 Especialista en Docencia y Pedagogía Universitaria. Ingeniero de Sistemas. Docente Facultad de Ingeniería, Uniagustiniana. ORCID: <https://orcid.org/0000-0002-1850-1634> Correo electrónico: mauricio.alonso@uniagustiniana.edu.co

It is necessary to analyse network behaviour to ensure the quality of web applications in the cloud that transmit information such as data, images, sound, video, etc., some in real time, and that generate large volumes of traffic. Understanding the traffic flowing through the network enables network capacity planning when managing limited resources, such as in the case of small and medium-sized enterprises in rural areas. As a product of the research analysis, a free software prototype will be developed to perform the measurements and predictions in rural areas. The results of the implementation show that the proposed approach is superior to other forecasting methods in terms of accuracy and predictability.

Keywords: Forecasting; SMEs; Bandwidth; Neural Networks; Traffic.

I. INTRODUCCIÓN

EN LA actualidad las pequeñas y medianas empresas, que están ubicadas en zonas rurales, alejadas y de difícil acceso en Colombia, tienen recursos limitados con relación al software, conectividad e infraestructura que no permiten una alta productividad en sus procesos; generando pérdidas a largo plazo, por tal razón se deben desarrollar herramientas acordes a sus necesidades para una toma de decisiones acertada. La incursión de la inteligencia artificial en especial de los algoritmos de las redes neuronales permite diseñar herramientas basadas en Machine Learning para la optimización de sus recursos.

La investigación se basa en el estudio comparativo de tres algoritmos de redes neuronales, seleccionando uno, cuyo objetivo es aplicarlo como método para predecir el comportamiento del tráfico de la red (consumo de ancho de banda) desde la pequeña y mediana empresa (pyme), hacia la nube nativa, que permita modelar una red neuronal multicapa, con un mínimo de tres capas, mediante un prototipo basado en software libre que contribuya a la optimización de los recursos de red que soportan las aplicaciones web.

Menciona[1] respecto al tráfico de la red que “el tráfico producido en una red de computadoras no es más que una serie de tiempo, es decir un conjunto de valores de un mismo atributo censados en períodos regulares”. En los últimos años las redes neuronales artificiales se han utilizado para el análisis predictivo, lo cual esboza[2], “las redes neuronales artificiales, a través de su proceso de entrenamiento son capaces de determinar relaciones no lineales entre un conjunto de datos, asociando a patrones de entrada la o las salidas correspondientes. Es así, que algunos modelos de RNA (Redes Neuronales Artificiales) son utilizados para determinar proyecciones a partir de una fuente de datos; esta característica puede ser aprovechada para realizar predicciones, por ejemplo, para determinar ancho de banda disponible”.

Es necesario analizar el comportamiento de la red para garantizar que aplicaciones web que se encuentran en la nube en servidores como Amazon Web Service AWS, transfiriendo información como datos, imágenes, sonidos, videos, entre otros, algunos en tiempo real y que generan alto tráfico, funcionen con calidad. Tener una visión general del tráfico que fluye a través de la red permite generar una planificación de la capacidad de la red cuando se administran recursos limitados como en el caso de pequeñas y medianas empresas, así mismo[2] justifica la necesidad de predecir el tráfico de red, “se requiere un sistema de predicción de tráfico para propósitos de planeación y dimensionamiento, esto permitirá pronosticar las demandas de tráfico según períodos de tiempo anteriores”. Las aplicaciones que se desarrollan actualmente necesitan un mayor ancho de banda y una óptima calidad de servicio, las conexiones hacia la nube provocan que la red que debe soportar este tipo de tráfico cumpla con la demanda solicitada.

Según la revisión de literatura y proyectos relacionados no se encuentra un software a la medida para las pymes rurales que realice la predicción en el consumo del ancho de banda hacia la nube y se use como herramienta para la toma de decisiones en la planeación y optimización de recursos humanos y computacionales. Se pretende con este artículo identificar cual es el algoritmo más óptimo para la predicción usando algoritmos supervisados.

II. TRABAJADORES RELACIONADOS

Estudio sobre conectividad en zonas rurales en Colombia

En Colombia la brecha digital es una gran preocupación para las entidades étnico-territoriales debido a que los programas y planes de conectividad que se han intentado implementar desde el Estado en las comunidades rurales y las regiones periféricas del país aún no se ha reducido. Si bien

en el primer trimestre de 2020 el país avanzó enormemente en aspectos de conectividad e infraestructura con el aumento del acceso a internet fijo, que aumentó de 200.000 puntos conectados a 7,67 millones. Según el Ministro de Tecnologías de la Información y las Comunicaciones en el 2021 “23 departamentos y Bogotá registran velocidades de descarga superiores a los 10 Mbps. Hace un año solo 12 departamentos del país registraron velocidades de descarga promedio de 10 Mbps” [3]; pero incluso en los municipios de las zonas conectadas de la costa pacífica, la velocidad de descarga promedio no supera los 10 Mbps, según la cobertura mapa del proyecto nPerf; el cual mide la calidad de la conexión a internet.

En atención a lo dispuesto en el parágrafo 1 del artículo 31 de la Ley 1978 de 2019 -adicionado por el artículo 6 de la Ley 2108 de 2021-, el problema identificado por la Comisión de Regulación de Comunicaciones (CRC) es la “Deficiencia de

conectividad del servicio de Internet fijo en zonas rurales, apartadas y de difícil acceso” [4]. A continuación, se procederá a desarrollar las causas y consecuencias del referido problema, las cuales se enuncian en el esquema contenido en la Fig 1.

En Colombia se definen como las causas en materia de conectividad fija los desincentivos para la prestación del servicio de acceso a internet fijo, como también el alcance de las políticas públicas en materia de conectividad; las limitaciones de la demanda en la prestaciones del servicio de acceso a internet fijo, la baja oferta en la prestación del servicio de acceso a internet y dificultades para el despliegue de infraestructura; y las consecuencias que conllevan, son la baja apropiación del servicio de internet fijo, con esto el bajo aprovechamiento de los potenciales beneficios del uso de internet, como la imposibilidad de acceso a internet en los niveles de velocidad requeridos, debido a los altos costos en la prestación del servicio de internet fijo.

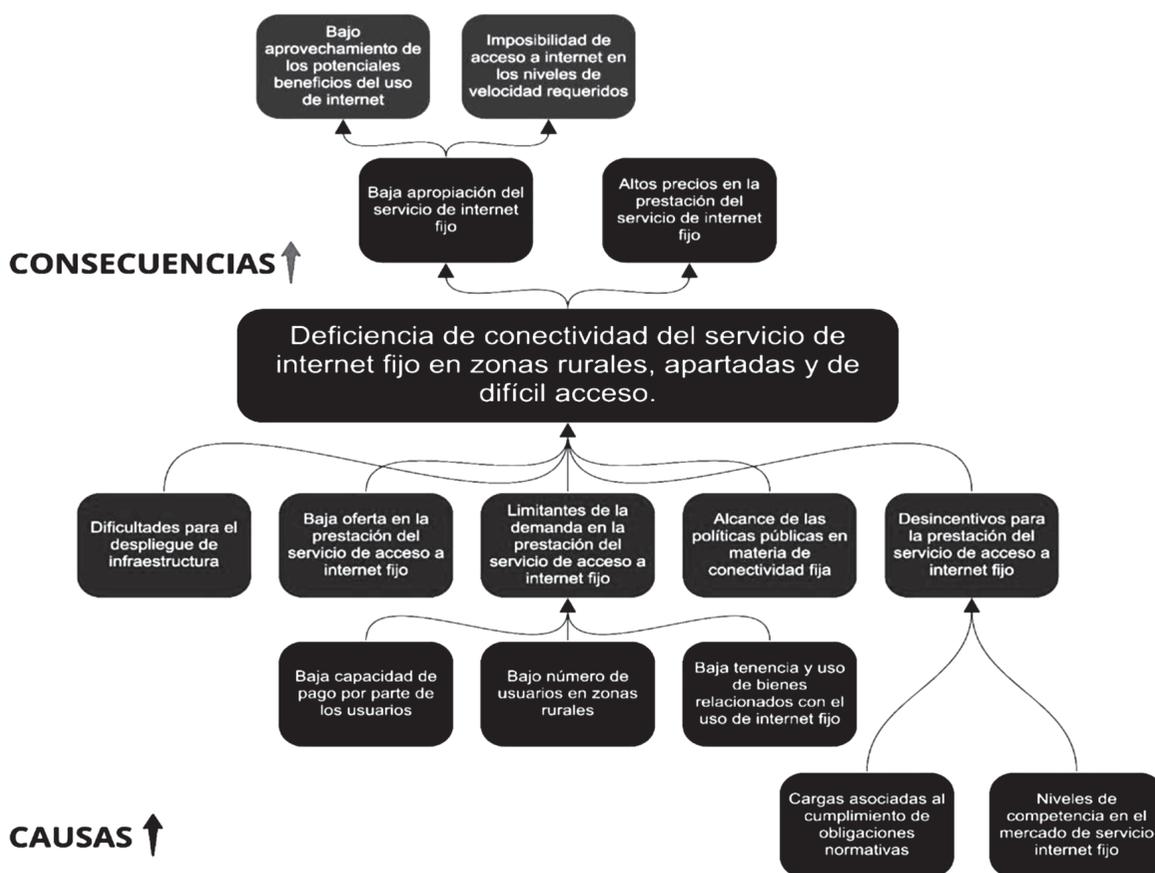


Fig. 1. Deficiencia de conectividad del servicio de Internet fijo en zonas rurales Fuente: Elaboración CRC [pág. 13][4]

El estudio permite evidenciar que las empresas en zonas rurales y de difícil acceso requieren gestionar de una manera eficiente los recursos limitados que poseen para optimizar las aplicaciones que se encuentran ubicadas en la nube.

Trabajos relacionados con algoritmos de Redes Neuronales Artificiales

En[5] se muestra cómo la tecnología de la informática y la comunicación en red está experimentando un rápido desarrollo, lo que lleva a un número creciente de aplicaciones y servicios disponibles en línea. A medida que hay más aplicaciones disponibles en línea, el tráfico de red se convierte en un problema importante ya que las altas cargas de red pueden limitar el acceso a los usuarios. En este documento, proponen un modelo de predicción del modelo de Promedio Móvil de Regresión Automática No Lineal (NARMA) para ayudar a los administradores de red a pronosticar el tráfico de Internet y planificar sus recursos en consecuencia. El estimador de perceptrón multicapa (MLP) se utilizó en este documento. El rendimiento del modelo se evaluó utilizando el error cuadrático medio (MSE), las pruebas de correlación y las pruebas de histograma residual con un buen acuerdo entre el modelo y los resultados reales.

En el trabajo de investigación[6], se realiza un estudio sobre la congestión del tráfico rodado que tiene un importante impacto socioeconómico y medioambiental. Una gestión dinámica y precisa del tráfico permitiría un uso más eficiente de la infraestructura vial y prevenir situaciones de congestión. Esta gestión sería posible si se dispusiera de una predicción precisa del estado del tráfico, para lo cual este artículo propone una nueva técnica de predicción de tráfico basada en redes neuronales convolucionales. A diferencia de las soluciones existentes, estas redes permiten explorar no sólo la evolución temporal del tráfico, sino también su evolución espacial. La red propuesta ha sido entrenada utilizando datos reales de tráfico, y su rendimiento ha sido comparado con soluciones actuales, en concreto con el de una red neuronal que solo procesa temporalmente los datos de tráfico. Los resultados demuestran que el procesado espacio temporal de los datos de tráfico mediante la red neuronal convolucional propuesta mejora la predicción del estado del tráfico

un 22% en términos de error absoluto y un 35% en términos de error relativo.

En el artículo[7] se presenta un estudio cuyo objetivo fue identificar los factores clave que influyen en la adopción del usuario de la tecnología de aprendizaje colaborativo basada en la nube en el contexto educativo. Basado en la Teoría Unificada de Aceptación y Uso de la Tecnología (UTAUT), la construcción de personalización se vinculó con la intención de comportamiento, la expectativa de rendimiento y la expectativa de esfuerzo. Esta investigación aplicó un nuevo enfoque metodológico que combina tanto el Fuzzy Analytic Hierarchy Process (FAHP) como el Structural Equation Modeling (SEM) para determinar el peso relativo y la importancia de los factores, así como para probar las hipótesis propuestas en el modelo de investigación. Utilizando un cuestionario de encuesta, se recolectó información de 150 estudiantes de cuatro universidades públicas malayas. Los hallazgos de FAHP demostraron que la expectativa de rendimiento, la influencia social y la personalización fueron los factores más importantes que predecían la intención conductual de adoptar tecnología de aprendizaje colaborativo basada en la nube desde el punto de vista de los expertos. Los resultados del SEM mostraron que la intención conductual de los usuarios se vio significativamente influenciada por la expectativa de rendimiento, la expectativa de esfuerzo, la influencia social y la personalización. Aunque la personalización ejerció una influencia directa sobre la intención conductual, su influencia indirecta a través de la expectativa de rendimiento y la expectativa de esfuerzo también fue considerable. Este estudio y sus hallazgos pueden servir como base para que los proveedores de servicios en la nube, el ministerio de educación y las instituciones educativas puedan tomar decisiones estratégicas y sólidas sobre la adopción de la tecnología basada en la nube en entornos educativos.

Trabajos relacionados en Computación en la nube

En un intento por reducir el costo del desarrollo de simulaciones comerciales basadas en la nube[8], la Plataforma de simulación de CloudSME (CSSP) se ha desarrollado como un enfoque genérico que combina un AppCenter con el flujo de trabajo del marco de acceso científico de WS-GRADe/gUSE y las capacidades basadas en la

nube de CloudBroker Platform. El documento presenta el CSSP y dos casos de estudio representativos de áreas claramente diferentes que ilustran cómo se pueden crear simulaciones comerciales basadas en nubes múltiples.

En la investigación[9], la clasificación de los paquetes de red es crucial en la detección de intrusos. Como los sistemas de detección de intrusos son la defensa principal de la infraestructura de las redes, deben adaptarse al aumento exponencial de las amenazas. A pesar del hecho de que los investigadores han ideado muchas técnicas de aprendizaje automático, esta área de investigación aún está lejos de encontrar sistemas perfectos con alta precisión de detección de paquetes maliciosos. El aprendizaje profundo es un subconjunto del aprendizaje automático y tiene como objetivo imitar el funcionamiento del cerebro humano en el procesamiento de datos para su uso en la toma de decisiones. Ya ha demostrado excelentes capacidades para tratar muchos problemas del mundo real, como el reconocimiento facial y los sistemas inteligentes de transporte. Este documento desarrolla una red neuronal profunda artificial para detectar paquetes maliciosos en el tráfico de red. La red neuronal profunda artificial se construye con cuidado y gradualmente para confirmar el número óptimo de neuronas de entrada y salida y el mecanismo de aprendizaje dentro de las capas ocultas. El rendimiento se analiza realizando varios experimentos en conjuntos de datos de tráfico de código abierto del mundo real utilizando métricas de clasificación bien conocidas. Los experimentos han mostrado resultados prometedores para la aplicación en el mundo real en la clasificación binaria del tráfico de red.

Trabajos relacionados con Tráfico de la red

En su trabajo[10] expone dos enfoques que emplean las técnicas de aprendizaje automático para permitir la predicción del tráfico en redes ópticas elásticas. Los resultados muestran que la aplicación de estrategias de adaptación tiene un rendimiento superior, que es una oportunidad futura para que los operadores de telecomunicaciones mejoren la eficiencia de sus arquitecturas de red.

En el trabajo de[11], se esboza como las aplicaciones web y móviles modernas tienen una carga de usuario muy variable desde diferentes ubica-

ciones geográficas que depende de la hora del día, las multitudes de flash y la tasa de crecimiento (alta o baja) de la carga del usuario. Independientemente de la ubicación de alojamiento de servidor de las aplicaciones, los usuarios de diferentes regiones geográficas necesitan aplicaciones altamente receptivas porque el mal rendimiento da como resultado clientes insatisfechos. Una encuesta reciente sugiere que las aplicaciones móviles pierden el 34% de sus usuarios debido al mal tiempo de respuesta. Para mejorar la experiencia del usuario de las aplicaciones, proponen la migración global en vivo (GLM) de máquinas virtuales, que permite migrar máquinas virtuales entre diferentes regiones geográficas a través de redes de área extensa (WAN). La configuración experimental consiste en una infraestructura de red habilitada para red definida por software (SDN) debido a su control extensivo sobre el enrutamiento de tráfico de red. El objetivo es colocar las máquinas virtuales en los sitios más cercanos a la región con una gran población de usuarios de aplicaciones respectivas. Sin embargo, existen altas probabilidades de congestión en enlaces entre sitios si se sigue una secuencia de migración aleatoria mientras se realizan migraciones de VM múltiples que pueden provocar fallas de migración. Formulan una Programación de enteros mixtos (MIP) y lo resuelven usando una técnica de sub gradiente de relajación lagrangiana. También proponen un sistema de control basado en retroalimentación con reguladores autoajustables para mejorar la secuencia de migración regularmente. Las extensas simulaciones muestran que la solución propuesta logra una mejora mínima de 38.09% en comparación con el algoritmo con mejor ajuste.

III. METODOLOGÍA

El experimento de predicción de tráfico realizado consistió en instalar aplicaciones web en la nube, a través de un servidor Linux en Amazon, simulando los servicios de red en una Pyme.

Se realizaron mediciones del consumo de ancho de banda de la red de una Pyme conectada a la nube. Se instala un software para medir el consumo de datos transmitidos desde el cliente web hacia los servicios que se encuentran configurados en la nube, así como los datos que recibe el cliente web desde el servidor, durante un periodo de un

mes. Se evalúan los datos capturados para realizar una depuración y obtener un consolidado aceptable para aplicar al modelo de red neuronal.

Al entrenar la red neuronal se espera que aprenda a predecir el tráfico mediante algoritmos de activación y optimización de pesos que maximizan o minimizan la función de error que mide el rendimiento de la red. El entrenamiento es un proceso iterativo hasta llegar a la solución óptima de la red neuronal. El algoritmo de entrenamiento depende de su configuración.

Como se muestra en la Fig. 1, el primer paso fue definir un set para entrenamiento de buena calidad, seguido se preparan los datos mediante un script, luego se estrena la red mediante un proceso iterativo que se va refinando, luego se genera el gráfico de rendimiento de la red, se comprueba el análisis de entrenamiento y se finaliza con una Tabla de resultados.

Se resume la Metodología a desarrollar en la presente investigación en la Fig. 2.[12]

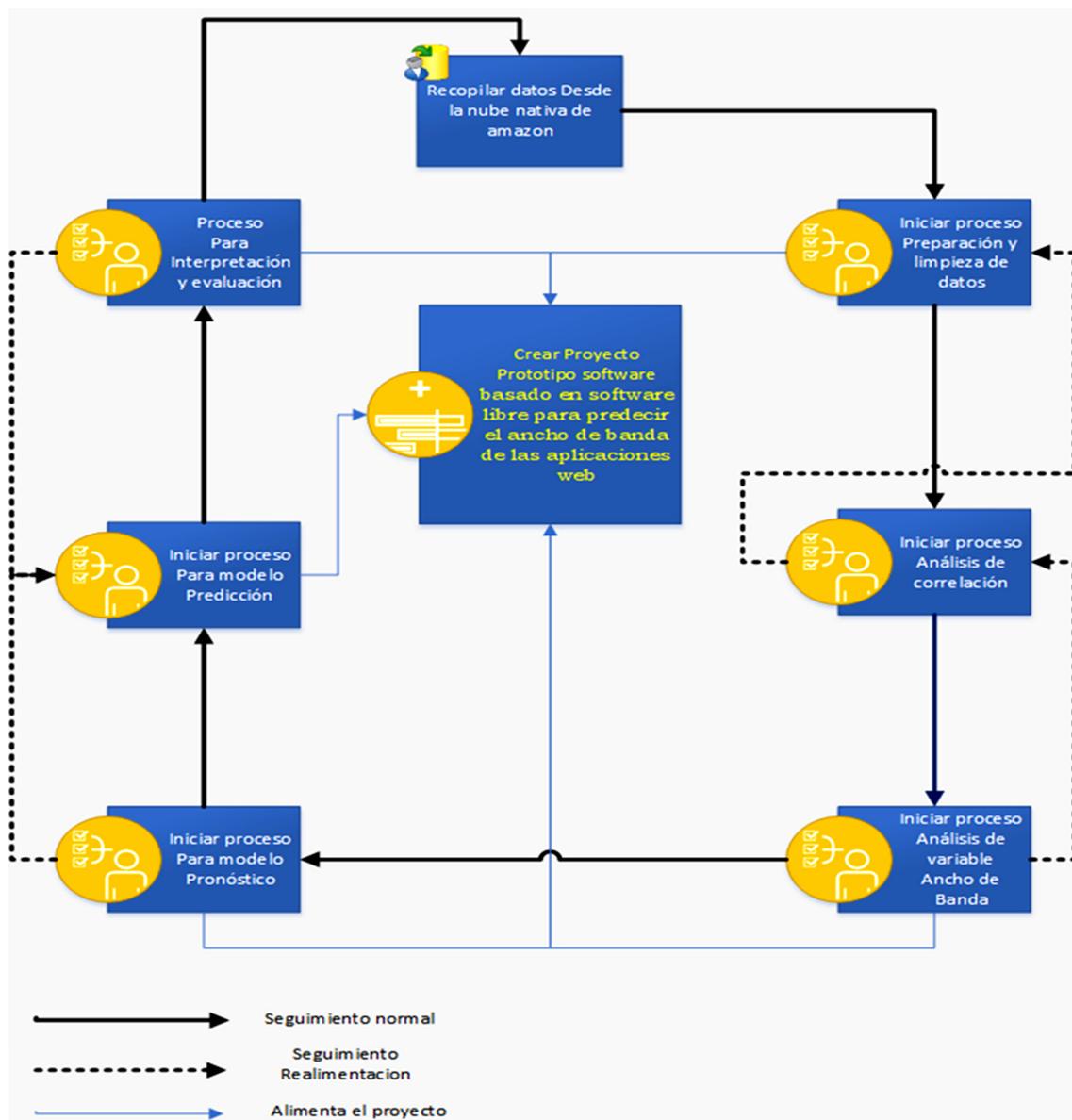


Fig. 2. Procedimiento para predicción del tráfico de red. Fuente. Elaboración Propia

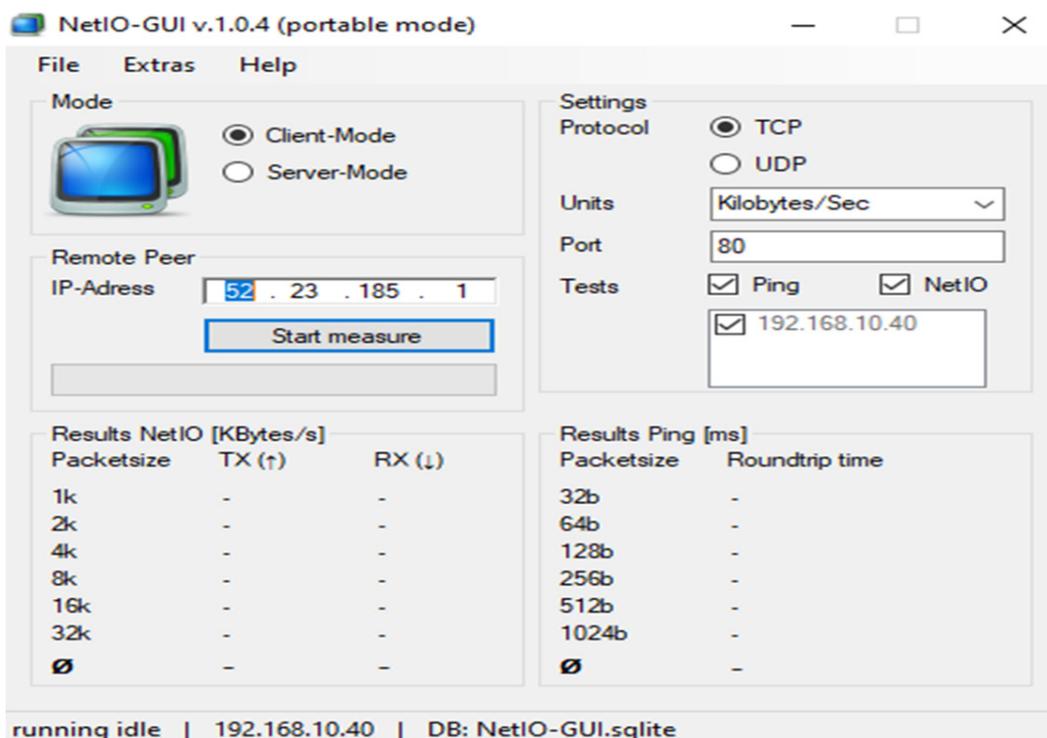


Fig. 3. Configuración NetIO-cliente

IV. RESULTADOS Y DISCUSIÓN

A. Evaluación de Set de Datos para el entrenamiento

El conjunto de datos utilizados para realizar el entrenamiento fue tomado con la herramienta NetIO, basados en el modelo cliente-servidor. Para la toma de la información relacionada se realizaron los siguientes pasos, frente a las aplicaciones web instaladas en la nube en Amazon Web Services (AWS).

i. Realizar la toma de datos con NETIO, como modelo cliente-servidor:

Para la toma de datos frente a las aplicaciones web en la nube se instalaron tres aplicaciones en AWS, las cuales fueron las siguientes: una página web en el puerto 80, una aplicación web tipo ERP en el puerto 8080 y, por último, una aplicación web para manejo de historia clínica en el puerto 9000.

Se realizó la toma de datos para cada puerto (80, 8080, 9000) desde un cliente (NetIO), Fig. 3, ubicado en la Pyme simulada, y el servidor un

equipo instalado en AWS (basado en Linux, herramienta NetIO-Server), como se observa en la Fig. 4, después se dispuso un solo set de datos que incluía todos los puertos antes mencionados, con un total de 670 instancias, como se muestra en la Fig. 5, donde se presenta un ejemplo del set de entrenamiento.

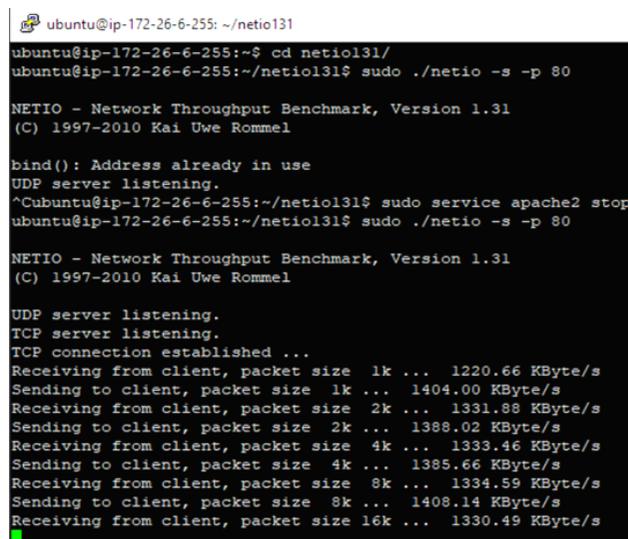


Fig. 4. Configuración NetIO-server.

Local Host	Remote Host	Timestamp	Protocol	TX 1k	TX 2k	TX 4k	TX 8k	TX 16k	TX 32k
192.168.10.40	52.23.185.1	9/07/2019 21:48	TCP	1127903	1180775	1182781	1177634	1187864	1193685
192.168.10.40	52.23.185.1	9/07/2019 21:49	TCP	1170745	1221721	1214320	1214554	1218139	1237893
192.168.10.40	52.23.185.1	9/07/2019 21:51	TCP	1270027	1205854	1303479	1317860	1315647	1327672
192.168.10.40	52.23.185.1	9/07/2019 21:53	TCP	1209227	1240832	1152924	1256527	1241855	1265449
192.168.10.40	52.23.185.1	9/07/2019 21:55	TCP	1155346	1205896	1186426	1118663	1211312	1132902
192.168.10.40	52.23.185.1	9/07/2019 21:57	TCP	1380270	1426472	1424762	1353616	1298952	1372516
192.168.10.40	52.23.185.1	9/07/2019 21:59	TCP	1136396	1172118	1176966	1171388	1075671	1197558
192.168.10.40	52.23.185.1	9/07/2019 22:01	TCP	1217034	1270733	1260129	1261823	1276734	809556
192.168.10.40	52.23.185.1	9/07/2019 22:02	TCP	1265792	1344465	1313361	1221523	1324875	1337786
192.168.10.40	52.23.185.1	9/07/2019 22:04	TCP	1219582	1272448	1155297	1263742	1194889	1197558
192.168.10.40	52.23.185.1	9/07/2019 22:06	TCP	1168793	1212336	1226090	1218776	1219567	1238293
192.168.10.40	52.23.185.1	9/07/2019 22:09	TCP	846031	912028	887192	918143	902555	969129
192.168.10.40	52.23.185.1	9/07/2019 22:11	TCP	887413	878076	992099	942969	658560	850281
192.168.10.40	52.23.185.1	9/07/2019 22:13	TCP	1001239	925475	942953	998028	963740	968663
192.168.10.40	52.23.185.1	9/07/2019 22:14	TCP	1272127	1308343	1316738	1324230	1321874	1082717
192.168.10.40	52.23.185.1	9/07/2019 22:16	TCP	1265189	1317791	1316540	1318210	1323013	1338220
192.168.10.40	52.23.185.1	9/07/2019 22:18	TCP	918143	954610	935450	1003089	999714	1034423
192.168.10.40	52.23.185.1	9/07/2019 22:20	TCP	1136595	1176775	1167241	1178903	1186708	1198530
192.168.10.40	52.23.185.1	9/07/2019 22:22	TCP	930988	1085697	1108714	1149273	1080474	1135470
192.168.10.40	52.23.185.1	9/07/2019 22:24	TCP	1207595	1260594	1267903	1264334	1262509	1257640
192.168.10.40	52.23.185.1	9/07/2019 22:25	TCP	1345446	1413989	1416479	1431670	1426282	1398059

Fig. 5. Segmento de Set de datos del total de 670 instancias

ii. Depurar el set de datos para realizar el entrenamiento

Se ajustó dataset, tomando en cuenta los campos TX_AVG (valor promedio, Upload), RX_AVG (valor promedio, Download), Número puerto; también se convirtió los datos Mb/seg.

B. Diseño del experimento y evaluación del algoritmo de entrenamiento

Se presenta la arquitectura de las redes neuronales artificiales multicapa, por medio de un algoritmo base desarrollado en Python, con el cual se realizan varios experimentos de simulación con los algoritmos de entrenamiento de redes neuronales (Fig. 6), Levenberg-Marquart (LM), Backpropagation (BP), y Grado descendente (GD); con función aprendizaje de adaptación Grado Descendente con Momento (GDM), y una función para medir Error Cuadrático Medio (MSE, Mean Squared Error), para identificar cuál de ellos presentaba un mejor resultado de entrenamiento y con error de aproximación cercano o aproximadamente igual a cero; con el cual se pueda predecir el ancho de banda de una red de datos.

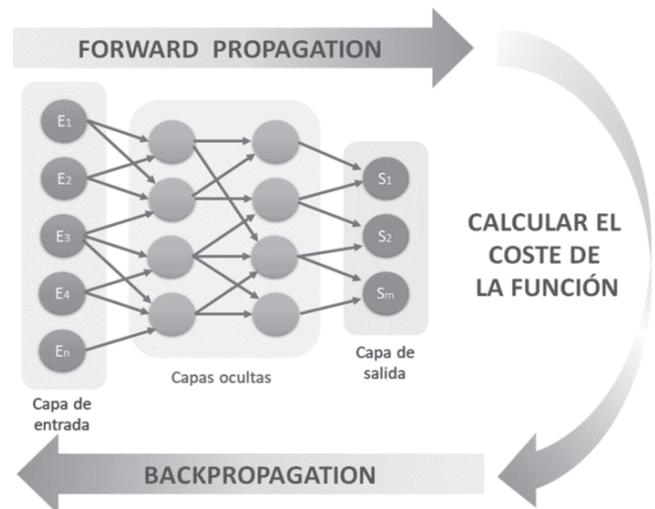


Fig. 6. Esquema de funcionamiento-red neuronal[13]

El algoritmo base está construido en Python como se muestra en la Fig. 7, incluyendo a NeuPy, es una biblioteca de Python para crear prototipos y construir redes neuronales. NeuPy usa Tensorflow como backend. admite muchos tipos diferentes de redes neuronales, desde un perceptrón simple hasta modelos de aprendizaje profundo[14]. Esto permite a los usuarios entrenar fácilmente redes

```
2
3 @author: raosdipro
4 """
5 #carga de las librerias usadas para el programa de python
6 from sklearn.model_selection import train_test_split
7 from sklearn import metrics
8 import numpy
9 from neupy import algorithms
10 from neupy.layers import *
11
12 numpy.random.seed(7)
13
14 # carga los dataset
15 dataset = numpy.loadtxt("data-fin-pl.csv", delimiter=",")
16 # dividido en variables de entrada (X) y salida (Y)
17 X = dataset[:,0:2]
18 Y = dataset[:,4]
19
20 #se defina los dataset de entrenamiento, test o validacion
21 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=7)
22
23 #se define el modelo de la red neuronal (2-->10-->1)
24 network = Input(2) >> Relu(10) >> Sigmoid(1)
25 #network.show()
26
27 #se define el algoritmo de entrenamiento
28 optimizer = algorithms.LevenbergMarquardt(network, verbose=True, show_epoch=5)
29
30 #se entrena la red con los dataset
31 optimizer.train(X_train, Y_train, X_test, Y_test, epochs=100)
32
33 #se grafica el entrenamiento con las perdidas (loss-->mse) frente a la epocas
34 optimizer.plot_errors()
35
36 #se realiza la prediccion de la red
37 y_predicted = network.predict(X, X_train)
38
39 #se calcula el modelo de efectividad o precision
40 score = metrics.accuracy_score(Y, y_predicted.round())
41 print("Validation accuracy: {:.2%}".format(score))
```

Fig. 7. Algoritmo base desarrollado en Python.

neuronales con arquitecturas construibles en GPU; también se usan las siguientes bibliotecas como son Scikit-Learn: es una biblioteca de código abierto que provee herramientas científicas para el análisis de datos y data mining (minería de datos). Emplea algoritmos de clasificación, regresión y agrupamiento; y opera de manera simultánea con librerías como NumPy[15], y Numpy: es una librería de Python orientada al cálculo vectorial y matricial. Proporciona una estructura de datos de matriz que garantizan cálculos eficientes con matrices[16].

Para la evaluación del algoritmo de entrenamiento más favorable para la predicción del uso del ancho de banda en una red de datos, entre una pyme rural frente a la nube nativa (AWS), se tuvo en cuenta una topología de red neuronal con una estructura de red de tres capas (2 Input, 10 capas ocultas y 1

de salida). La red neuronal utiliza el algoritmo Perceptrón multicapa de Sklearn que puede aprender tanto modelos no lineales como lineales.

Entradas: Tx_AVG (upstream) y Rx_AVG (Downstream)
Salidas: Ancho de banda (ANCHO)

i. Análisis y Resultados del experimento.

Con el algoritmo base desarrollado en Python, incluyendo las diferentes bibliotecas pertinentes aplicadas; y ejecutando los diferentes algoritmos de entrenamiento de redes neuronales propuesto en esta investigación, para determinar cuál de ellos es el más eficiente, en predecir el consumo del ancho de banda que utilizan las aplicaciones web (ubicadas en AWS), frente los clientes PYMES

(pequeñas y medianas empresas) las cuales están en una ubicación geográfica de tipo rural.

Para el entrenamiento de la red neuronal, se describe el procedimiento realizado en el código del algoritmo base en Python.

ii. Carga del Data set y sus respectivas librerías.

```
#carga de las librerías usadas para el programa de python
from sklearn.model_selection import train_test_split
from sklearn import metrics
import numpy
from neupy import algorithms
from neupy.layers import *

numpy.random.seed(7)

# carga los dataset
dataset = numpy.loadtxt("data-fin-p1.csv", delimiter=",")
# dividido en variables de entrada (X) y salida (Y)
X = dataset[:,0:2]
Y = dataset[:,4]
```

Fig. 8. Las entradas son TX_AVG, RX_AVG, y la salida es el ancho de banda.

iii. Se construyen los Dataset de entrenamiento, y de test o validación.

```
#se defina los dataset de entrenamiento, test o validacion
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=7)
```

En las tomas realizadas, se usó una relación del 20% para test y 80% entrenamiento.

iv. Modelo de la red neuronal.

```
#se define el modelo de la red neuronal (2-->10-->1)
network = Input(2) >> Relu(10) >> Sigmoid(1)
#network.show()
```

La función de activación de las neuronas para las capas ocultas y de salida, cuenta con:

- Identidad o lineal: Activación no operativa, útil para implementar cuellos de botella lineales.
- Logística: La función logística sigmoidea.
- tanh: La función hiperbólica tangencial.
- ReLu: La función de unidad lineal rectificadora.

Se aplicó la función tanh para las capas ocultas y Logística para la capa de salida, la cual presentó

mejor rendimiento para la predicción (se realizaron pruebas con las otras funciones, pero no mostraron una mejor predicción).

v. Se define el algoritmo de entrenamiento.

```
#se define el algoritmo de entrenamiento
optimizer = algorithms.LevenbergMarquardt(network, verbose=True,
show_epoch=5)
```

En esta línea se reemplazan los algoritmos, con los cuales se va a evaluar su eficiencia.

vi. Se entrena la red con los Dataset de entrenamiento y de test.

```
#se entrena la red con los dataset
optimizer.train(X_train, Y_train, X_test, Y_test, epochs=100)
```

Para los diferentes algoritmos que se van a evaluar.

vii. Se grafica el entrenamiento con las pérdidas, y las épocas.

```
#se grafica el entrenamiento con las pérdidas (loss-->mse)
frente a la épocas
optimizer.plot_errors()
```

Como se muestra en la Fig. 9.

viii. Se realiza la predicción de la red neuronal.

```
#se realiza la predicción de la red
y_predicted = network.predict(X, X_train)
```

ix. Se calcula la efectividad de predicción o precisión.

```
#se calcula el modelo de efectividad o precision
score = metrics.accuracy_score(Y, y_predicted.round())
print("Validation accuracy: {:.2%}".format(score))
```

Se resumen los datos de la evaluación en la Tabla I con base en los resultados obtenidos, a partir del código construido en Python.

Tabla. I. Resultados de evaluación.

Algorithm	MSE - Train	MSE-Validation	Accuracy
Backpropagation	0.022589	0.023255	97.76%
Grado descendiente	0.038066	0.075425	89.25%
Levenberg-Marquart	0.108209	0.104478	89.18%

Analizando la información se puede observar que el algoritmo de Backpropagation presentó la mejor eficiencia, para la predicción del ancho de banda.

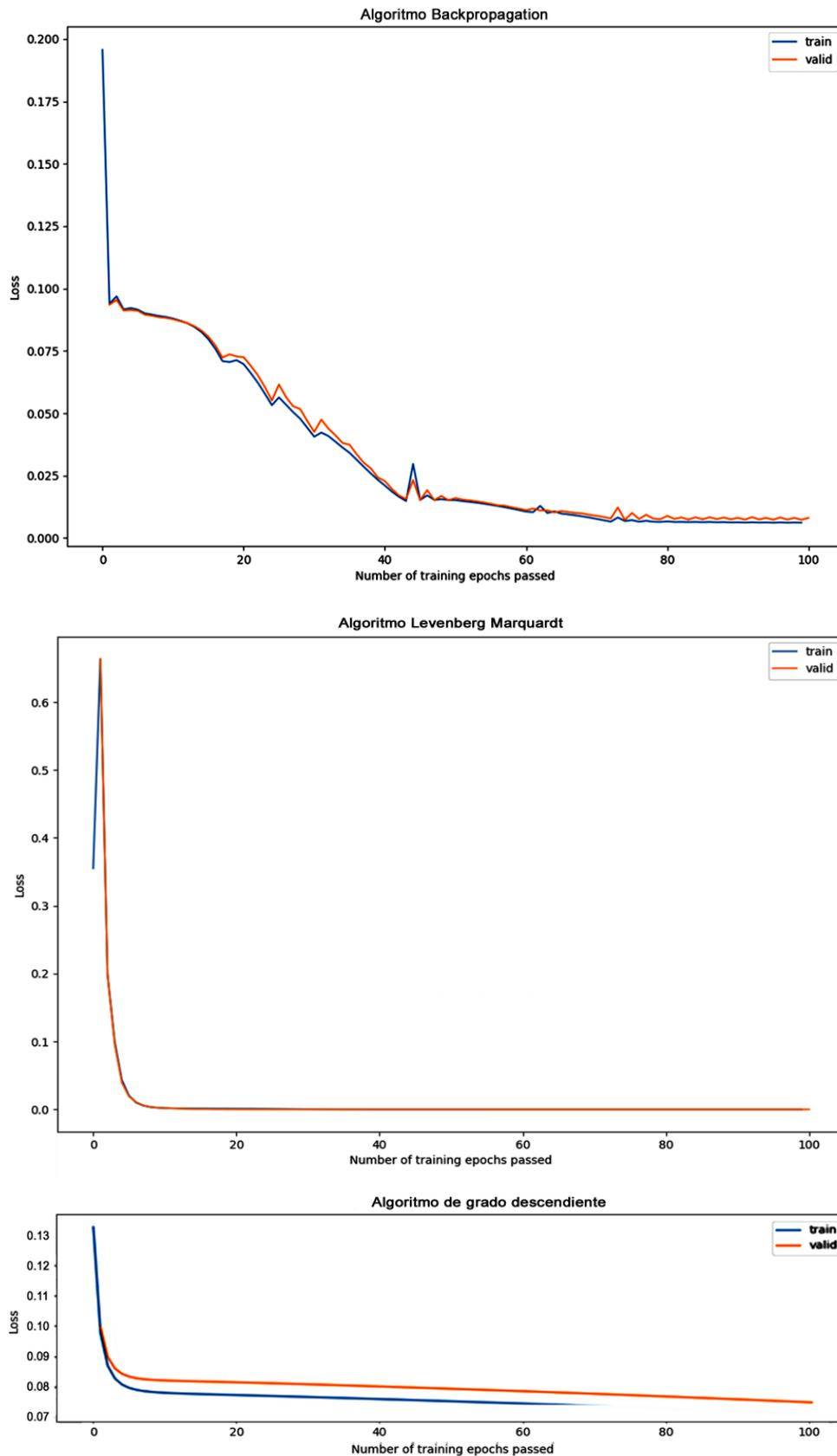


Fig. 9. Gráficas con los diferentes algoritmos [BP, GD, y LM], Loss vs epochs.

V. CONCLUSIÓN

Se utilizó Amazon Web Services (AWS), como nube nativa para la carga de aplicaciones web necesarias para determinar el comportamiento del consumo del ancho de banda, por la calidad de servicio, el rendimiento y los costos que ofrece al cliente.

Con la metodología propuesta, se realizó la predicción del consumo del ancho de banda de las aplicaciones web ubicadas en AWS. Terminadas las pruebas con los tres algoritmos se determinó que el algoritmo de entrenamiento de redes neuronales con mejor desempeño corresponde al Backpropagation; arrojando un porcentaje de asertividad del 97.76%, para modelo de red neuronal de 3 capas con una estructura de 2,10,1.

Las redes neuronales como modelo matemático son eficaces para la predicción, clasificación y categorización de información crítica, como es el tráfico de datos que se genera hacia la nube, en donde están ubicadas las aplicaciones web que requieren calidad de servicio y optimización de recursos humanos en una Pymes rurales.

REFERENCIAS

- [1] McCluskey, P., «Feedforward and recurrent neural networks and genetic programs for stock market and time series forecasting», Master of Science, Brown University. (1993).
- [2] Piedra, N., Chicaiza J., López J. and Garcia, J. «Study of the Application of Neural Networks in Internet Traffic Engineering», Hdl.handle.net, [Online]. Available: <http://hdl.handle.net/10525/1028>. (2008).
- [3] Colombia alcanzó 7,67 millones de conexiones fijas en el tercer trimestre de 2020. (n.d.). Retrieved May 6, 2023, from <https://www.larepublica.co/economia/colombia-alcanzo-767-millones-deconexiones-fijas-en-el-tercer-trimestre-de-2020-3120559>.
- [4] Z. De and D. Acceso, “PROMOCIÓN DE LA CONECTIVIDAD A INTERNET FIJO EN Página 2 de 81 PROMOCIÓN DE LA CONECTIVIDAD A INTERNET FIJO EN ZONAS DE DIFÍCIL ACCESO.
- [5] Sahrani, M. N., Zan, M. M. M., Yassin, I. M., Zabidi, A., & Ali, M. S. A. M. (2017). Artificial Neural Network non-linear auto regressive moving average (NARMA) model for internet traffic prediction. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(1-3), 145-149. Sancho, F. (23 de abril de 2017). Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente [Entrada de Blog]. Recuperado de: <http://www.cs.us.es/~fsancho/?e=165>.
- [6] Mena-Oreja, J., & Gozalvez, J. (2017). Predicción de la Velocidad del Tráfico Basada en Redes Neuronales Convolucionales.
- [7] Yadegaridehkordi, E., Nizam Bin Md Nasir, M. H., Fazmidar Binti Mohd Noor, N., Shuib, L., & Badie, N. (2018). Predicting the adoption of cloud-based technology using fuzzy analytic hierarchy process and structural equation modelling approaches. *Applied Soft Computing*, 66, 77-89. <https://doi.org/10.1016/J.ASOC.2017.12.051>.
- [8] Taylor, S. J. E., Kiss, T., Anagnostou, A., Terstyanszky, G., Kacsuk, P., Costes, J., & Fantini, N. (2018). The CloudSME simulation platform and its applications: A generic multi-cloud platform for developing and executing commercial cloud-based simulations. *Future Generation Computer Systems*, 88, 524-539. <https://doi.org/10.1016/j.future.2018.06.006>
- [9] Abdullah, S. A., & Al-Ashoor, A. (2020). An artificial deep neural network for the binary classification of network traffic. *International Journal of Advanced Computer Science and Applications*, 11(1), 402-408.
- [10] Aibin, M. (2018). Traffic prediction based on machine learning for elastic optical networks. *Optical Switching and Networking*, 30, 33-39. <https://doi.org/10.1016/j.osn.2018.06.001>
- [11] Nanda, S., & Hacker, T. J. (2017). TAG: Traffic-Aware Global Live Migration to Enhance User Experience of Cloud Applications. In 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 202-209). IEEE. <https://doi.org/10.1109/CloudCom.2017.13>
- [12] Desarrollo e innovación en ingeniería [recurso electrónico] / Edgar Serna M., ed. — 4a. ed. — Medellín: Instituto Antioqueño de Investigación, Pág 27-36 — (Ingeniería y ciencia), DOI: <http://doi.org/10.5281/zenodo.3387679>. (2019)
- [13] Calvo, D. <<https://www.diegocalvo.es/definicion-de-red-neuronal/>> [Artículo publicado el 12 de Julio de 2017]
- [14] <<http://neupy.com/pages/home.html/>>
- [15] <<https://platzi.com/blog/librerias-de-machine-learning-tensorflow-scikit-learnpythorch-y-keras/>> [Artículo publicado en 2018].
- [16] <<https://aprendeia.com/introduccion-a-numpy-python-1/>> [Artículo publicado el 21 de septiembre de 2018].