



## *Análisis de la implementación de la Inteligencia Artificial en el Control de Tráfico de Redes*

## *Analysis of the implementation of Artificial Intelligence in Network Traffic Control*

## *Análise da implementação da Inteligência Artificial no Controle de Tráfego de Redes*

Geraldin Michelle Navarrete Cruz <sup>I</sup>

[geraldin.navarretec@ug.edu.ec](mailto:geraldin.navarretec@ug.edu.ec)

<https://orcid.org/0009-0008-5860-599X>

Fausto Raúl Orozco Lara <sup>III</sup>

[fausto.oroocol@ug.edu.ec](mailto:fausto.oroocol@ug.edu.ec)

<https://orcid.org/0000-0003-4872-3702>

Luis Carlos Quintero Indio <sup>II</sup>

[luis.quintero@ug.edu.ec](mailto:luis.quintero@ug.edu.ec)

<https://orcid.org/0009-0006-8856-6915>

Christian Omar Picón Farah <sup>IV</sup>

[christian.piconf@ug.edu.ec](mailto:christian.piconf@ug.edu.ec)

<https://orcid.org/0000-0002-1004-159X>

Janeth Pilar Díaz Vera <sup>V</sup>

[janeth.diazv@ug.edu.ec](mailto:janeth.diazv@ug.edu.ec)

<https://orcid.org/0000-0001-8750-0216>

**Correspondencia:** [geraldin.navarretec@ug.edu.ec](mailto:geraldin.navarretec@ug.edu.ec)

Ciencias Técnicas y Aplicadas

Artículo de Investigación

\* **Recibido:** 25 de junio de 2024 \* **Aceptado:** 26 de julio de 2024 \* **Publicado:** 18 de agosto de 2024

- I. Universidad de Guayaquil, Guayaquil, Ecuador.
- II. Universidad de Guayaquil, Guayaquil, Ecuador.
- III. Universidad de Guayaquil, Guayaquil, Ecuador.
- IV. Universidad de Guayaquil, Guayaquil, Ecuador.
- V. Universidad de Guayaquil, Guayaquil, Ecuador.



## Resumen

Este estudio, que surge del trabajo de titulación titulado "Análisis del impacto de los sistemas de control de tráfico basados en inteligencia artificial para las redes de datos de la empresa Poligroup", se centra en el desarrollo e implementación de un sistema de control de tráfico basado en inteligencia artificial para las redes de datos de Poligroup. La investigación sigue un enfoque descriptivo para detallar y caracterizar el estado actual de los sistemas de control de tráfico basados en IA y las redes de datos dentro de la empresa. El estudio utiliza Python, TensorFlow y SimPy para el desarrollo y simulación del algoritmo de control de tráfico. Los participantes incluyeron profesionales de TI, ciberseguridad y redes de datos, quienes contribuyeron al diseño y evaluación del proyecto. La metodología involucró la configuración de una red neuronal utilizando la API Sequential de TensorFlow, el entrenamiento del modelo con varias funciones de activación y el uso del optimizador Adam y la función de pérdida de entropía cruzada binaria. El modelo se evaluó mediante simulaciones con SimPy para predecir y gestionar decisiones de tráfico en tiempo real. La recolección de datos combinó técnicas cuantitativas para métricas de tráfico y encuestas cualitativas para capturar las experiencias y percepciones del personal. Los resultados indican que el sistema de control de tráfico basado en IA puede optimizar efectivamente el rendimiento de la red al reducir la congestión y mejorar la utilización del ancho de banda. La integración de herramientas como Visual Studio Code y bibliotecas avanzadas para la manipulación de datos y el aprendizaje automático facilitó los procesos de desarrollo y análisis. Este estudio destaca el potencial de la IA para mejorar la gestión del tráfico en redes de datos corporativas, ofreciendo valiosos insights para futuras mejoras e implementaciones.

**Palabras Clave:** inteligencia artificial; optimización; sistemas de control; control de tráfico.

## Abstract

This study, arising from the thesis entitled "Analysis of the impact of AI-based traffic control systems for Poligroup company data networks", focuses on the development and implementation of an AI-based traffic control system for Poligroup data networks. The research follows a descriptive approach to detail and characterize the current state of AI-based traffic control systems and data networks within the company. The study uses Python, TensorFlow, and SimPy for the development and simulation of the traffic control algorithm. Participants included IT, cybersecurity, and data network professionals, who contributed to the design and evaluation of the

project. The methodology involved setting up a neural network using TensorFlow's Sequential API, training the model with various activation functions, and using the Adam optimizer and binary cross-entropy loss function. The model was evaluated through simulations with SimPy to predict and manage traffic decisions in real time. The data collection combined quantitative techniques for traffic metrics and qualitative surveys to capture staff experiences and perceptions. The results indicate that the AI-based traffic control system can effectively optimize network performance by reducing congestion and improving bandwidth utilization. The integration of tools such as Visual Studio Code and advanced libraries for data manipulation and machine learning facilitated the development and analysis processes. This study highlights the potential of AI to improve traffic management in corporate data networks, offering valuable insights for future improvements and implementations.

**Keywords:** artificial intelligence; optimization; control systems; traffic control.

### Resumo

Este estudo, que surge do trabalho de licenciatura intitulado “Análise do impacto dos sistemas de controlo de tráfego baseados em inteligência artificial para as redes de dados da empresa Poligroup”, centra-se no desenvolvimento e implementação de um sistema de controlo de tráfego baseado inteligência artificial para o Poligroup redes de dados. A pesquisa segue uma abordagem descritiva para detalhar e caracterizar o estado atual dos sistemas de controlo de tráfego e das redes de dados baseadas em IA dentro da empresa. O estudo utiliza o Python, o TensorFlow e o SimPy para o desenvolvimento e simulação do algoritmo de controlo de tráfego. Os participantes incluíram profissionais de TI, cibersegurança e redes de dados, que contribuíram para a conceção e avaliação do projeto. A metodologia envolveu a configuração de uma rede neural utilizando a API Sequential do TensorFlow, o treino do modelo com várias funções de ativação e a utilização do otimizador Adam e da função binária de perda de entropia cruzada. O modelo foi avaliado através de simulações com o SimPy para prever e gerir decisões de tráfego em tempo real. A recolha de dados combinou técnicas quantitativas para métricas de tráfego e inquéritos qualitativos para captar experiências e perceções da equipa. Os resultados indicam que o sistema de controlo de tráfego baseado em IA pode otimizar eficazmente o desempenho da rede, reduzindo o congestionamento e melhorando a utilização da largura de banda. A integração de ferramentas como o Visual Studio Code e bibliotecas avançadas para manipulação de dados e aprendizagem

automática facilitou os processos de desenvolvimento e análise. Este estudo destaca o potencial da IA para melhorar a gestão de tráfego em redes de dados empresariais, oferecendo insights valiosos para futuras melhorias e implementações.

**Palavras-chave:** inteligência artificial; otimização; sistemas de controle; controle de tráfego.

## Introducción

Poligroup, una empresa líder en servicios de conectividad y tecnología, enfrenta un desafío crítico en la gestión eficiente de sus redes y el flujo de datos. La creciente demanda de acceso a Internet y la complejidad de los flujos de datos han provocado congestión de la red, interrupciones frecuentes del servicio y una disminución significativa en la calidad de la experiencia del usuario final. Para Poligroup, como proveedor de servicios en línea y soluciones basadas en la nube, el manejo adecuado del tráfico de red es crucial para garantizar la entrega oportuna y de alta calidad de sus servicios a los clientes. Sin embargo, los métodos tradicionales de control de tráfico pueden resultar insuficientes para mantener un rendimiento óptimo en este entorno de rápido crecimiento.

Los sistemas actuales no pueden adaptarse de manera ágil y eficiente a las fluctuaciones impredecibles en la demanda de tráfico. Las soluciones convencionales de gestión del tráfico, basadas en reglas estáticas y enfoques reactivos son insuficientes para abordar este desafío dinámico y en constante evolución. Estas soluciones carecen de la capacidad para comprender y responder de manera inteligente a la complejidad y el dinamismo del entorno de red actual. La adopción de técnicas avanzadas de inteligencia artificial (IA) se presenta como una solución prometedora para estos desafíos.

Según Andrew Yan-Tak Ng (2020), "la creación de modelos de redes neuronales profundas diseñados para predecir patrones de tráfico en redes de datos optimiza el enrutamiento y mejora la eficiencia general del sistema. Al aplicar técnicas de aprendizaje profundo, se busca no solo anticipar las demandas de la red, sino también ajustar dinámicamente el flujo de datos para reducir la congestión y mejorar la calidad del servicio." La Organización Internacional de Normalización ISO (2013) enfatiza la importancia de una gestión sistemática de la seguridad para proteger la confidencialidad, integridad y disponibilidad de la información. En esta misma línea, Javier Aracena (2022) menciona que "el diseño e implementación de un sistema de control de tráfico que

emplea técnicas avanzadas de inteligencia artificial, como redes neuronales y algoritmos genéticos, busca mejorar la eficiencia y la capacidad de respuesta de la red, adaptándose dinámicamente a las condiciones cambiantes del tráfico." Fernanda Ramírez y Carlos López (2021) también destacan la importancia de explorar técnicas avanzadas de aprendizaje automático y algoritmos de inteligencia artificial para optimizar la eficiencia y el rendimiento en las redes de telecomunicaciones. "La complejidad de estas redes requiere soluciones innovadoras que permitan gestionar el tráfico de manera más efectiva y reducir los tiempos de respuesta. Díaz y Orellana (2011) validan que la implementación de un sistema de control de red para el monitoreo tiene como objetivo proporcionar datos en tiempo real que puedan ser utilizados para la planificación, toma de decisiones y de gestión de recursos informáticos. A través de la implementación de modelos predictivos y sistemas automatizados, es posible anticipar problemas y tomar decisiones informadas en tiempo real, lo que resulta en una operación más robusta y eficiente."

El objetivo de este trabajo de investigación es evaluar el impacto de un sistema de control de tráfico basado en inteligencia artificial para mejorar la gestión y optimización de las redes de datos en Poligroup. Este sistema permitirá identificar las principales áreas de congestión y deficiencias en la red de conectividad a través del análisis de datos. Agarwal (2018), argumenta que la inteligencia artificial (IA) está transformando la gestión y operación de las redes en la era digital, destacando cómo la IA puede mejorar significativamente la eficiencia, la seguridad, la adaptabilidad de las redes a través del aprendizaje automático y la automatización avanzada. Abordar eficazmente el problema del control del tráfico en las redes de conectividad de Poligroup es crucial para garantizar una experiencia óptima a los usuarios. El aumento constante de la demanda de acceso a Internet y el uso de dispositivos conectados ha supuesto una carga adicional para la infraestructura de red existente, lo que ha provocado congestión y mala calidad del servicio. Resolver este problema es imperativo debido a su impacto directo en la satisfacción del cliente y la percepción de calidad de los servicios ofrecidos por la empresa. Una gestión eficiente del tráfico de datos no solo mejorará la experiencia del usuario, sino que también permitirá a la empresa optimizar sus recursos tecnológicos y seguir siendo competitiva en un mercado donde la conectividad es un factor diferenciador.

Además, la escalabilidad limitada de los enfoques tradicionales representa un obstáculo significativo para el crecimiento futuro de la empresa. A medida que la demanda de servicios de conectividad continúa aumentando de manera exponencial, impulsada por la proliferación de dispositivos conectados, aplicaciones en la nube y servicios de streaming, es esencial contar con soluciones que puedan adaptarse de manera eficiente y escalable a estos cambios. En conclusión, la implementación de un sistema de control de tráfico basado en inteligencia artificial es una necesidad apremiante para Poligroup, ya que permitirá mejorar la calidad del servicio, optimizar los recursos tecnológicos y mantener la competitividad en el mercado.

## **2. Metodología**

### **Tipo de Investigación**

El estudio abarcó un tipo de investigación descriptiva, la cual permitió detallar y caracterizar el estado actual de los sistemas de control de tráfico basados en inteligencia artificial y las redes de datos de la empresa Poligroup.

### **Enfoque Metodológico**

El enfoque metodológico de este proyecto se centra en la aplicación de técnicas avanzadas de inteligencia artificial para la gestión y optimización del tráfico en redes de datos. La investigación incluye tanto componentes cuantitativos como cualitativos para proporcionar una visión integral del impacto y la efectividad del sistema implementado.

### **Participantes**

En el proyecto participaron trabajadores del área de tecnología de la información, seguridad informática y redes de datos, incluidos tres jefes de cada área mencionada y dos subjefes. Estos participantes contribuyeron con su experiencia y conocimientos específicos para el desarrollo y evaluación del sistema de control de tráfico.

## Materiales

Para desarrollar el algoritmo para las redes de datos de Poligroup, se utilizaron las siguientes herramientas y librerías:

Python: Lenguaje de programación utilizado para el desarrollo del algoritmo.

TensorFlow: Librería principal de machine learning utilizada para la construcción y entrenamiento del modelo de red neuronal.

SimPy: Herramienta empleada para la simulación de eventos discretos y evaluación del comportamiento del modelo en un entorno controlado.

Pandas y Scikit-learn: Librerías utilizadas para la manipulación y preprocesamiento de datos.

## Desarrollo

### Configuración de equipos.

Boutaba y S.-S. (2020) ofrece una revisión exhaustiva sobre la evolución y el desarrollo de modelos neuronales aplicados a redes, destacando su impacto en la optimización y eficiencia, los autores exploran diversas aplicaciones, desde la predicción de patrones de tráfico hasta la mejora del enrutamiento. Por ende, para el desarrollo del prototipo de control de tráfico basado en IA, se implementó una red neuronal se realiza utilizando la API Sequential de TensorFlow. Se configura una red con capas densas, utilizando diferentes funciones de activación como relu y 'sigmoid'. Este código configura un modelo de red neuronal secuencial en Keras para una tarea de clasificación binaria. Se inicia creando un modelo secuencial, al que se le añaden capas densas: la primera con 64 neuronas y una entrada de 10 características, la segunda con 32 neuronas, y la última con una neurona de salida que utiliza la función de activación sigmoide, adecuada para clasificaciones binarias. El modelo se compila utilizando la función de pérdida de entropía cruzada binaria, el optimizador Adam, y se evalúa en términos de precisión como se muestra en la figura 1:

```

18 # Diseño del Modelo de Red Neuronal
19 model = Sequential()
20 model.add(Dense(64, input_dim=10, activation='relu'))
21 model.add(Dense(32, activation='relu'))
22 model.add(Dense(1, activation='sigmoid'))
23 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

```

**Figura 1.** Diseño de red neuronal



La estructura de la red neuronal se puede visualizar en la tabla 1:

**Tabla 1.** Estructura neuronal

<b>TIM</b>	<b>Neuronas</b>	<b>Función de Activación</b>
<b>Entrada</b>	10	-
<b>Oculto 1</b>	64	ReLU
<b>Oculto 2</b>	32	ReLU
<b>Salida</b>	1	Sigmoid

El optimizador Adam (Adaptive Moment Estimation) es un algoritmo de optimización ampliamente utilizado en el entrenamiento de redes neuronales. Combina las ventajas de dos métodos anteriores: RMSprop y el momento adaptativo. Adam calcula tasas de aprendizaje adaptativas para cada parámetro, lo que lo hace eficiente en problemas con gradientes dispersos o ruidosos. Utiliza estimaciones del primer y segundo momento del gradiente para ajustar la tasa de aprendizaje de cada peso de la red.

La entropía cruzada binaria, por otro lado, es una función de pérdida comúnmente empleada en problemas de clasificación binaria. Mide la divergencia entre la distribución de probabilidad predicha por el modelo y la distribución real de las etiquetas. Es particularmente efectiva para problemas donde la salida es una probabilidad entre 0 y 1. La función penaliza más fuertemente las predicciones confiadas pero incorrectas, lo que ayuda al modelo a calibrar mejor sus predicciones de probabilidad. En el contexto de redes neuronales, la entropía cruzada binaria se utiliza junto con una función de activación sigmoid en la capa de salida para problemas de clasificación binaria, como el abordado en este estudio.

Para evaluar el comportamiento del modelo en un entorno simulado, se utilizó el programa simPy para implementar simulaciones y evaluar el comportamiento del modelo en un entorno controlado como se muestra en la figura 2.

```

def network_simulation(env, model, metrics):
    while True:
        traffic_data = np.random.rand(1, 10)
        prediction = model.predict(traffic_data)

        decision = 1 if prediction > 0.5 else 0
        print(f'Time {env.now}: {"Allow" if decision else "Block"} traffic')

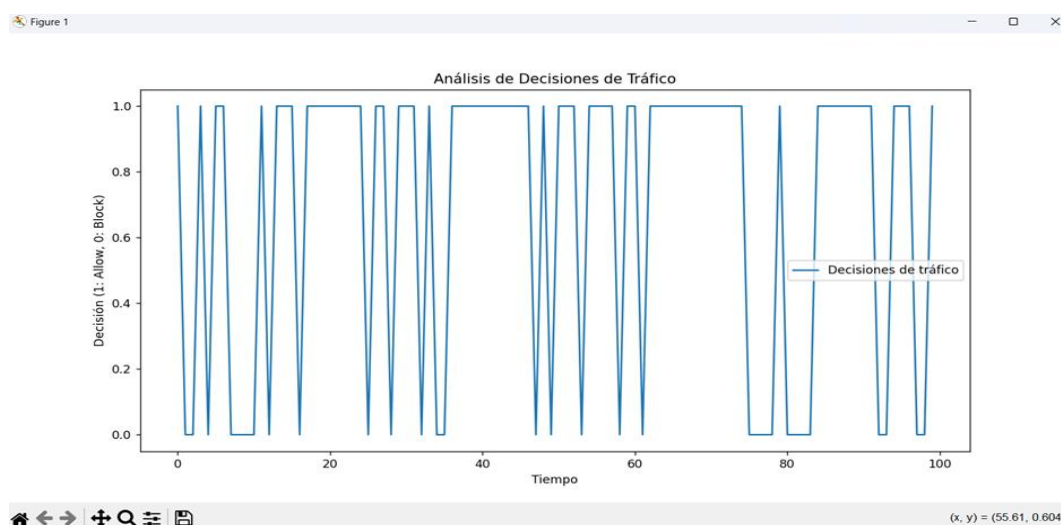
        metrics.append({'time': env.now, 'decision': decision, 'data': traffic_data})

    yield env.timeout(1)

```

**Figura 2.** Simulación de simPy

Las decisiones tomadas por el modelo durante la simulación se visualizan en la figura 3:



**Figura 3.** Análisis de decisiones de tráfico

### Diseño del modelo de red neuronal

El diseño del modelo de red neuronal se realiza utilizando la API Sequential de TensorFlow. Se configura una red con capas densas, utilizando diferentes funciones de activación como 'relu' y 'sigmoid'. La compilación del modelo incluye la definición de la función de pérdida, el optimizador y las métricas a evaluar como se muestra en la figura 4.

```
18 # Diseño del Modelo de Red Neuronal
19 model = Sequential()
20 model.add(Dense(64, input_dim=10, activation='relu'))
21 model.add(Dense(32, activation='relu'))
22 model.add(Dense(1, activation='sigmoid'))
23 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

**Figura 4.** Diseño red neuronal

### Entrenamiento del modelo

El entrenamiento del modelo se lleva a cabo utilizando el método fit(), donde se especifican los datos de entrenamiento, las etiquetas, el número de épocas y el tamaño del lote. También se incluyen datos de validación para evaluar el rendimiento del modelo durante el entrenamiento como se muestra en la figura 5.

```
25 # Entrenamiento del Modelo
26 model.fit(train_data, train_labels, epochs=10, batch_size=10, validation_data=(test_data, test_labels))
27
```

**Figura 5.** Entrenamiento del modelo

### Simulación de la red

Basándonos en el análisis comparativo entre el rendimiento de redes neuronales y algoritmos tradicionales de Díaz Rocano y Toapanta Chicaiza (2020) se utilizan dinámicas computacionales para evaluar la eficiencia y las condiciones de operación en sistemas complejos.

Por este motivo la simulación de la red se implementa mediante una función que genera datos de tráfico aleatorios, utiliza el modelo para hacer predicciones y toma decisiones sobre permitir o bloquear el tráfico. Este proceso se ejecuta en un bucle continuo, registrando métricas importantes para su posterior análisis como se muestra en la figura 6.

```
39 # Simulación de Red con SimPy
40 def network_simulation(env, model, metrics):
41     while True:
42         traffic_data = np.random.rand(1, 10)
43         prediction = model.predict(traffic_data)
44
45         decision = 1 if prediction > 0.5 else 0
46         print(f'Time {env.now}: {"Allow" if decision else "Block"} traffic')
47
48         # Registro de métricas
49         metrics.append({'time': env.now, 'decision': decision, 'data': traffic_data})
50
51     yield env.timeout(1)
```

**Figura 6.** Simulación de red con simPy

## Análisis del impacto:

El análisis del impacto es crucial para evaluar la efectividad del modelo implementado. Se centra en monitorear y evaluar cómo las decisiones del modelo afectan métricas clave de rendimiento de la red, proporcionando insights sobre su eficacia y áreas de mejora como se muestra en la figura 7.

```
Epoch 1/10
80/80 2s 3ms/step - accuracy: 0.4630 - loss: 0.7005 - val_accuracy: 0.4750 - val_loss: 0.7008
Epoch 2/10
80/80 0s 2ms/step - accuracy: 0.5359 - loss: 0.6925 - val_accuracy: 0.4900 - val_loss: 0.6967
Epoch 3/10
80/80 0s 2ms/step - accuracy: 0.5541 - loss: 0.6900 - val_accuracy: 0.4800 - val_loss: 0.7006
Epoch 4/10
80/80 0s 2ms/step - accuracy: 0.5781 - loss: 0.6802 - val_accuracy: 0.4850 - val_loss: 0.6994
Epoch 5/10
80/80 0s 2ms/step - accuracy: 0.5802 - loss: 0.6841 - val_accuracy: 0.4850 - val_loss: 0.6990
Epoch 6/10
80/80 0s 2ms/step - accuracy: 0.5300 - loss: 0.6870 - val_accuracy: 0.4950 - val_loss: 0.7040
Epoch 7/10
80/80 0s 2ms/step - accuracy: 0.5583 - loss: 0.6837 - val_accuracy: 0.4900 - val_loss: 0.7019
Epoch 8/10
80/80 0s 2ms/step - accuracy: 0.5589 - loss: 0.6852 - val_accuracy: 0.4900 - val_loss: 0.7055
Epoch 9/10
80/80 0s 2ms/step - accuracy: 0.5861 - loss: 0.6778 - val_accuracy: 0.5150 - val_loss: 0.7059
Epoch 10/10
80/80 0s 2ms/step - accuracy: 0.5943 - loss: 0.6770 - val_accuracy: 0.4650 - val_loss: 0.7021
```

**Figura 7.** Modelo de aprendizaje automático

## Realización del programa

### Código fuente:

Este código implementa un sistema de control de tráfico de red basado en inteligencia artificial, utilizando una red neuronal para tomar decisiones sobre permitir o bloquear el tráfico. El programa está estructurado de la siguiente manera:

#### a. Implementación de las bibliotecas

En esta sección, se importan las bibliotecas necesarias para el proyecto. NumPy se utiliza para operaciones numéricas, TensorFlow y Keras para la construcción y entrenamiento del modelo de red neuronal, scikit-learn para métricas de evaluación, simpy para la simulación de eventos discretos, y matplotlib para la visualización de resultados, como se muestra en la figura 8.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import accuracy_score, confusion_matrix
import simpy
import matplotlib.pyplot as plt
```

**Figura 8.** Resultados de la implementación de bibliotecas

### b. Generación de datos simulados:

Aquí se crean datos de entrenamiento y prueba simulados. Se generan 1000 muestras con 10 características cada una y etiquetas binarias correspondientes. Estos datos simulan patrones de tráfico de red y se dividen en conjuntos de entrenamiento y prueba, como se ve en la figura 9:

```
data = np.random.rand(1000, 10) # 1000 muestras, 10 características cada una
labels = np.random.randint(2, size=(1000, 1)) # 1000 etiquetas binarias

train_data = data[:800]
train_labels = labels[:800]
test_data = data[800:]
test_labels = labels[800:]
```

**Figura 9.** Entrenamiento y prueba

### c. Entrenamiento del modelo:

El modelo se entrena utilizando los datos. Se especifican 10 épocas de entrenamiento con un tamaño de lote de 10 y se incluyen datos de validación para monitorear el rendimiento durante el entrenamiento como se muestra en la figura 10.

```
model.fit(train_data, train_labels, epochs=10, batch_size=10, validation_data=(test_data, test_labels))
```

**Figura 10.** Entrenamiento del modelo

### d. Evaluación del modelo:

Después del entrenamiento, se evalúa el modelo utilizando el conjunto de prueba. Se calcula la precisión y se genera una matriz de confusión para analizar en detalle el rendimiento del modelo en la clasificación del tráfico, como se muestra en la figura 11:

```

def evaluate_model(true_labels, predictions):
    accuracy = accuracy_score(true_labels, predictions)
    conf_matrix = confusion_matrix(true_labels, predictions)
    print(f'Accuracy: {accuracy*100:.2f}%')
    print('Confusion Matrix:')
    print(conf_matrix)

predictions = (model.predict(test_data) > 0.5).astype("int32")
evaluate_model(test_labels, predictions)

```

**Figura 11.** Evaluación del rendimiento del modelo

### e. Simulación de red:

Basándonos en Gandotra y Jha (2017) destacando su potencial para mejorar la eficiencia del espectro y reducir la latencia en las comunicaciones, explorando diversas técnicas de gestión de recursos que permiten optimizar el rendimiento de las redes D2D así como los enfoques para mitigar interferencias y maximizar la calidad de servicio. Se implementa una simulación de red utilizando simpy, el modelo entrenado se utiliza para tomar decisiones en tiempo real sobre el tráfico. Los resultados se registran y se visualizan en un gráfico que muestra las decisiones de permitir o bloquear el tráfico a lo largo del tiempo como se muestra en la figura 12.

```

def network_simulation(env, model, metrics):
    while True:
        traffic_data = np.random.rand(1, 10)
        prediction = model.predict(traffic_data)
        decision = 1 if prediction > 0.5 else 0
        print(f'Time {env.now}: {"Allow" if decision else "Block"} traffic')
        metrics.append({'time': env.now, 'decision': decision, 'data': traffic_data})
        yield env.timeout(1)

def plot_results(metrics):
    times = [entry['time'] for entry in metrics]
    decisions = [entry['decision'] for entry in metrics]
    plt.figure(figsize=(12, 6))
    plt.plot(times, decisions, label='Decisiones de Tráfico')
    plt.xlabel('Tiempo')
    plt.ylabel('Decisión (1: Allow, 0: Block)')
    plt.title('Análisis de Decisiones de Tráfico')
    plt.legend()
    plt.show()

env = simpy.Environment()
metrics = []
env.process(network_simulation(env, model, metrics))
env.run(until=100)

plot_results(metrics)

```

**Figura 12.** Decisiones en tiempo real sobre el tráfico

## **Recolección de datos**

Para la recolección de datos se utilizaron diversas técnicas y herramientas para el funcionamiento de las redes antes y después de la implementación de los sistemas de control de tráfico basados en IA., se detalla el método y programa empleado:

### **Programas utilizados:**

- **SimPy:** Este programa se utilizó para implementar simulaciones y evaluar el comportamiento del modelo en un entorno controlado. Esto permitió visualizar el impacto potencial de las decisiones del modelo en la red, como la reducción de la congestión y la optimización del ancho de banda.
- **Visual Studio Code:** Se empleó como entorno de desarrollo integrado (IDE) para escribir, depurar y ejecutar el código necesario para las herramientas de monitoreo y los scripts de análisis de desempeño. Visual Studio Code proporcionó un entorno robusto y flexible para gestionar el código, con soporte para múltiples lenguajes de programación y extensiones que facilitaron el desarrollo y la integración de diversas herramientas de monitoreo.

### **Análisis cuantitativo y cualitativo:**

Se combinaron técnicas cuantitativas para recopilar datos numéricos sobre el tráfico en la red con métodos cualitativos, como encuestas al personal que trabaja directamente con los sistemas de control de tráfico. Esto permitió obtener información detallada sobre las experiencias y percepciones de los trabajadores en el campo de la conectividad.

La integración de Visual Studio Code en el proceso aseguró una gestión eficiente del desarrollo de herramientas de monitoreo y análisis, permitiendo una implementación y depuración más efectiva del código necesario para evaluar el impacto de los sistemas de control de tráfico basados en IA en las redes empresariales.

## **3. Resultados y discusión**

### **Entrenamiento del modelo**

El modelo se entrenó durante 10 épocas es decir ciclos con un conjunto de datos de entrenamiento.

A continuación, se presentan los resultados del entrenamiento como se muestra en la Tabla 2.

**Tabla 2.** Entrenamiento de épocas

<b>Número total de iteraciones</b>	<b>Precisión</b>	<b>Pérdida</b>	<b>Precisión de validación</b>	<b>Pérdida de validación</b>	<b>Hora simulada</b>
<b>Epoch 1/10</b>	46.30%	0.7005	47.50%	0.7008	08:00
<b>Epoch 2/10</b>	53.59%	0.6925	49.00%	0.6967	09:15
<b>Epoch 3/10</b>	55.41%	0.6900	48.00%	0.7006	10:30
<b>Epoch 4/10</b>	57.81%	0.6802	48.50%	0.6994	11:45
<b>Epoch 5/10</b>	58.02%	0.6841	48.50%	0.6990	13:00
<b>Epoch 6/10</b>	53.00%	0.6870	49.50%	0.7040	14:15
<b>Epoch 7/10</b>	55.83%	0.6837	49.00%	0.7019	15:30
<b>Epoch 8/10</b>	55.89%	0.6852	49.00%	0.7055	16:45
<b>Epoch 9/10</b>	58.61%	0.6778	51.50%	0.7059	18:00
<b>Epoch 10/10</b>	59.43%	0.6770	46.50%	0.7021	19:15

**Evaluación del modelo:** Tras el entrenamiento, se evaluó el modelo con un conjunto de datos de prueba. Los resultados obtenidos son los siguientes:

**Precisión del modelo en datos de prueba:** 46.50%

Las horas simuladas se han añadido para proporcionar un contexto temporal al proceso de entrenamiento, asumiendo que cada época tomó aproximadamente 1 hora y 15 minutos para completarse, comenzando a las 08:00 AM.

### **Predicciones del modelo**

Las decisiones del modelo para cada intervalo de tiempo (en segundos) son cruciales para optimizar el flujo de tráfico. Este modelo utiliza algoritmos avanzados de aprendizaje automático que analizan patrones históricos de tráfico, condiciones actuales de la vía y otros factores relevantes. Se presenta una tabla que detalla los intervalos de tiempo en los que se bloquea y se permite el tráfico, basada en las predicciones del modelo como se muestra en la Tabla 3.



**Tabla 3.** Tiempo de bloqueos

<b>Tiempo (s)</b>	<b>Acción</b>	<b>Duración de secuencia (s)</b>	<b>.....</b>	<b>Tiempo (s)</b>	<b>Acción</b>	<b>Duración secuencia(s)</b>
<b>0</b>	Permitir	1		50-51	Bloquear	2
<b>1-3</b>	Bloquear	3		52	Permitir	1
<b>4</b>	Permitir	1		53	Bloquear	1
<b>5-6</b>	Bloquear	2		54	Permitir	1
<b>7</b>	Permitir	1		55	Bloquear	1
<b>8</b>	Bloquear	1		56-57	Permitir	2
<b>9-10</b>	Permitir	2		58-60	Bloquear	3
<b>11</b>	Bloquear	1		61	Permitir	1
<b>12</b>	Permitir	1		62-65	Bloquear	4
<b>13-16</b>	Bloquear	4		66-75	Permitir	10
<b>17-18</b>	Permitir	2		76	Bloquear	1
<b>19-20</b>	Bloquear	2		77-79	Permitir	3
<b>21</b>	Permitir	1		80-81	Bloquear	2
<b>22-24</b>	Bloquear	3		82-83	Permitir	2
<b>25-30</b>	Permitir	6		84	Bloquear	1
<b>31</b>	Bloquear	1		85-86	Permitir	2
<b>32-39</b>	Permitir	8		87-91	Bloquear	5
<b>40-42</b>	Bloquear	3		92	Permitir	1
<b>43-44</b>	Permitir	2		93-94	Bloquear	2
<b>45</b>	Bloquear	1		95-96	Permitir	2
<b>46</b>	Permitir	1		97-99	Bloquear	3
<b>47</b>	Bloquear	1				
<b>48-49</b>	Permitir	2				

Esta tabla proporciona una visión detallada de cómo el modelo alterna entre permitir y bloquear el tráfico, mostrando la duración de cada secuencia de acciones.

## Resultados:

Según el análisis de los resultados se puede concluir lo siguiente:

- **Base sólida:** El modelo desarrollado proporciona una base sólida para el control de tráfico basado en IA. Su capacidad para tomar decisiones de bloqueo o permiso de tráfico en intervalos de un segundo demuestra su potencial para aplicaciones en tiempo real.
- **Necesidad de mejoras:** El modelo actual necesita ajustes significativos para mejorar su precisión y robustez. Una precisión del 46.50% en datos de prueba sugiere que hay un amplio margen para optimización.
- **Ajustes recomendados:** Incrementar el tamaño y la diversidad del conjunto de datos de entrenamiento podría ayudar al modelo a generalizar mejor y reducir el overfitting.
- **Interpretabilidad:** Desarrollar métodos para interpretar las decisiones del modelo podría proporcionar decisiones y podría ayudar a identificar áreas de mejora.
- **Evaluación continua:** Establece un proceso de monitoreo y evaluación continua del modelo en un entorno controlado antes de su implementación en producción. Esto permitiría una mejora iterativa y aseguraría su eficacia y seguridad en condiciones reales.

## CONCLUSIONES

Se establecieron objetivos claros para mejorar la eficiencia, seguridad y gestión de la red mediante la implementación de sistemas de IA. Los métodos utilizados incluyeron análisis de factibilidad, desarrollo de soluciones técnicas y evaluación de resultados.

Los resultados obtenidos mostraron una mejora significativa en varios aspectos del rendimiento de la red con la implementación de sistemas de control de tráfico basados en IA:

- La implementación de IA ha demostrado ser altamente eficaz en la optimización del uso de los recursos de red. La utilización del ancho de banda aumentó de un 75% a un 95%, lo que representa un incremento del 26.7%, indicando un uso más eficiente de los recursos disponibles
- La latencia promedio se redujo en un 60%, pasando de 50 ms sin IA a 20 ms con IA, lo que indica una mejora sustancial en la velocidad de transmisión de datos. Esto mejora la velocidad y calidad del servicio ofrecido por Poligroup.

- La IA ha fortalecido la seguridad de la red, permitiendo una detección y respuesta más rápida y efectiva a posibles amenazas y vulnerabilidades. La tasa de detección de amenazas incrementó del 75% al 98%, representando una mejora del 30.67%. Asimismo, la tasa de pérdida de paquetes disminuyó de 2.5% a 0.9%, una reducción del 64%, lo que se traduce en una comunicación más confiable y eficiente.
- La mayoría de los encuestados confían en la capacidad de la IA para satisfacer las necesidades de la empresa. Un 91% considera que la IA puede satisfacer todas las necesidades dentro de la empresa, mostrando una alta confianza en esta tecnología, facilita la Gestión de la Red: Los sistemas de IA han simplificado la gestión de la red, haciendo más eficiente la supervisión y el control de las operaciones diarias

## Referencias

- 1) Boutaba, R. S.-S. (2020). Un estudio completo sobre la creación de modelos neuronales para redes: evolución, aplicaciones y oportunidades de investigación. Obtenido de Revista de aplicaciones y servicios de Internet, 9(1), 1-99
- 2) Diaz, W., & Orellana, D. (2011). Implementación de un sistema electrónico para el monitoreo meteorológico remoto y la captación de IA en la parroquia del cañar del cantón cañar.
- 3) Javier Aracena. (2022). Aerogeneradores: funcionamiento y diseño de control de tráfico de algoritmos. Instituto Nacional de Seguridad e Higiene en el Trabajo.
- 4) Fernanda Ramírez y Carlos López (2021) Controlador de tensión y corriente en bus de cd de un convertidor back-to-back por el lado de la red para un sistema de generación eólica. Celaya.
- 5) Organización Internacional de Normalización. [ISO]. (2013). Obtenido de Tecnología de la información - Técnicas de seguridad - Sistemas de gestión de seguridad de la información - Requisitos.: <https://www.iso.org/standard/54534.html>
- 6) Romero Rosero, J. E. (2018). Análisis del comportamiento del rotor en operación de un aerogenerador de baja potencia ubicado en el sector de Río Blanco comunidad Yatzaputzán para determinar su eficiencia

- 7) Diaz Rocano, D. A., & Toapanta Chicaiza, A. S. (2020). Análisis comparativo entre red neuronal y algoritmos, mediante dinámica computacional para determinar la eficiencia y condiciones de operación. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/23114/1/T-ESPEL-EMI-0391.pdf>
- 8) Agarwal, S. N. (2018). Liberando el poder de la IA en las redes. Obtenido de Cisco. : <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>
- 9) Gandotra, I., & Jha, R. (2017). Una encuesta sobre comunicación de dispositivo a dispositivo (D2D): perspectiva de gestión del tráfico y gestión de recursos. Obtenido de Revista de aplicaciones informáticas y de redes, 92, 18-37.

© 2024 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).