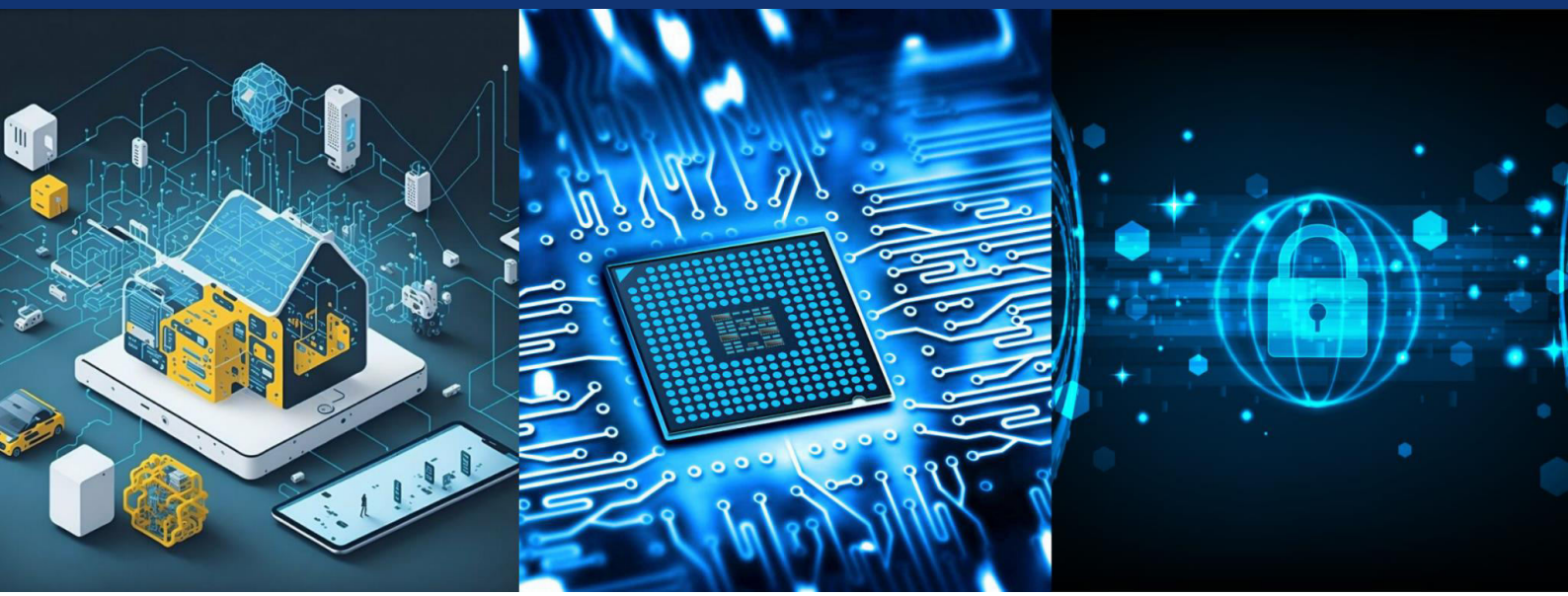
IJIRCCE

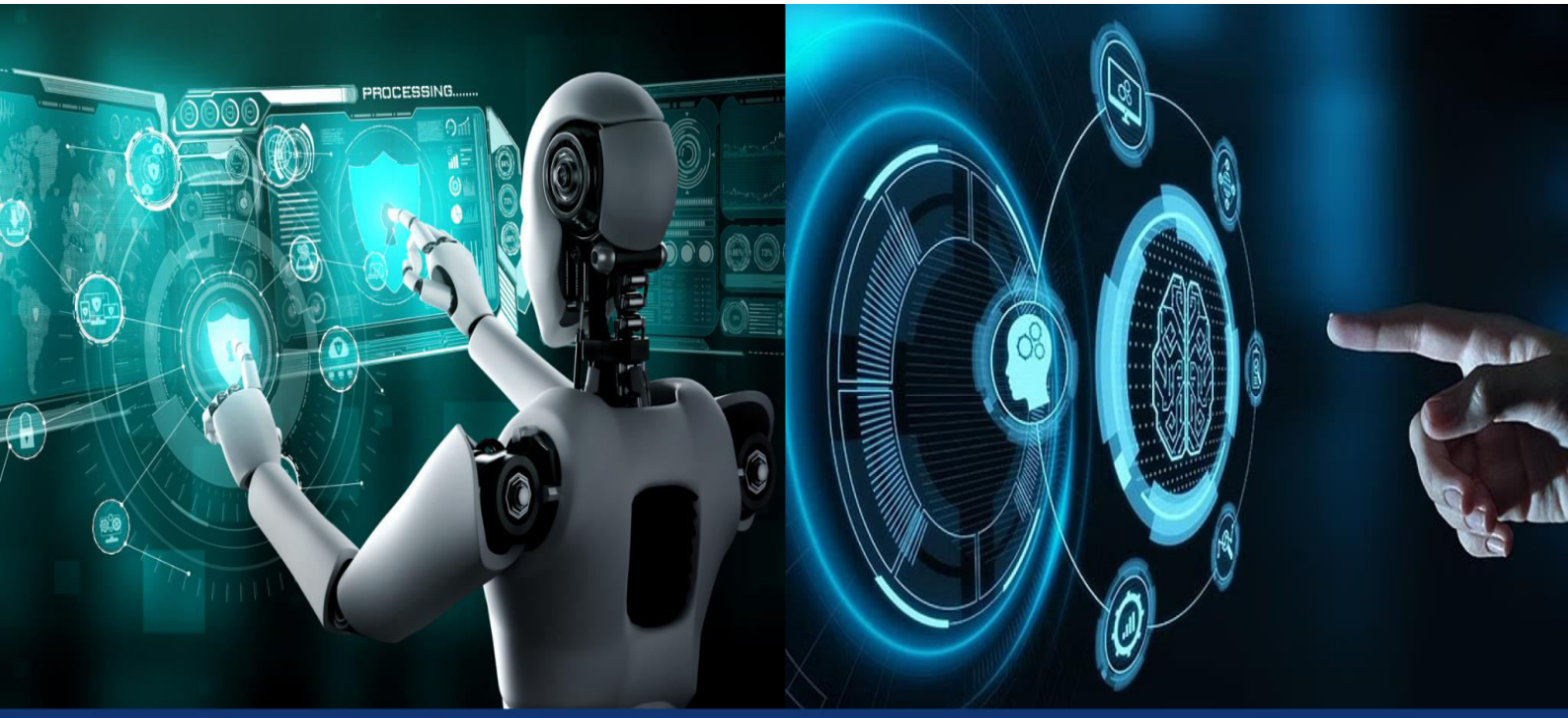
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 7, July 2026

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Application Performance and ISP Fault Prediction Using Network Traffic Analysis

Barige Abhishek, Dr. V. Uma Rani

Post Graduate Student, Department of Computer Science and Engineering, Data Science, Jawaharlal Nehru Technological University, Hyderabad, India

Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

ABSTRACT: The performance of modern web applications depends significantly on reliable network connectivity. Users often experience issues such as high latency, packet loss, buffering, and service interruptions, making it difficult to determine whether the problem originates from Internet Service Provider (ISP) networks or application-level faults. This paper presents a machine learning-based framework for Application Performance and ISP Fault Prediction using network traffic data captured through Wireshark. Key performance metrics including latency, packet loss, throughput, jitter, retransmissions, and packet count are extracted and analyzed. Machine learning models such as Random Forest and XGBoost are employed to classify network conditions into Normal, Network Fault, and Application Fault categories. A Streamlit-based dashboard provides interactive visualization and monitoring. Experimental results demonstrate improved fault diagnosis, reduced troubleshooting time, and enhanced network reliability.

KEYWORDS: Network Traffic Analysis, ISP Fault Prediction, Wireshark, XGBoost, Network Fault Detection.

I. INTRODUCTION

Network performance is becoming a crucial component in guaranteeing high-quality user experiences due to the growing use of cloud computing, online streaming, remote collaboration, and web-based applications. Issues including buffering, connection outages, service interruptions, and delayed application answers are frequently encountered by users. It is still difficult to determine if these issues are caused by application-level malfunctions or ISP network issues.

Traditional network monitoring tools provide raw traffic statistics and packet captures but lack intelligent fault diagnosis capabilities. Manual analysis of large volumes of network traffic data is time-consuming and often ineffective in rapidly identifying root causes. Machine Learning (ML) techniques provide an automated approach to analyze network traffic patterns and classify performance issues.

This paper proposes an intelligent fault prediction framework that utilizes Wireshark traffic data and machine learning algorithms to classify network conditions into Normal, Network Fault, and Application Fault categories. The system enables faster fault diagnosis and improved network performance management through an interactive visualization dashboard.

II. RELATED WORK

The growing dependence on internet-based applications and cloud services has increased the need for effective network performance monitoring and fault diagnosis. Network administrators often face challenges in identifying whether performance degradation is caused by ISP-related network issues or application-level problems. To address these challenges, several researchers have explored machine learning and network traffic analysis techniques for intelligent fault detection and performance monitoring. Parvat et al. proposed a machine learning-based framework for network traffic analysis and anomaly detection. The system analyzes large volumes of network traffic data and classifies traffic behavior using machine learning algorithms. Their study demonstrated that machine learning techniques can improve traffic analysis accuracy and assist administrators in identifying abnormal network conditions. However, the proposed approach primarily focuses on traffic classification and anomaly detection without distinguishing whether performance

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

issues originate from network faults or application-level problems. Furthermore, the framework lacks an automated root-cause analysis mechanism.

Hu and Zhou developed a Wireshark packet dissector for the DL/T 860 communication protocol to improve packet-level traffic analysis and protocol interpretation. The proposed system enhances network troubleshooting by providing detailed insights into communication packets and protocol fields. Although the approach improves packet inspection capabilities, it mainly focuses on protocol analysis rather than intelligent fault prediction. Additionally, the framework requires protocol-specific knowledge and does not provide automated classification of network performance issues.

III. PROPOSED ALGORITHM

A. *System Architecture*:

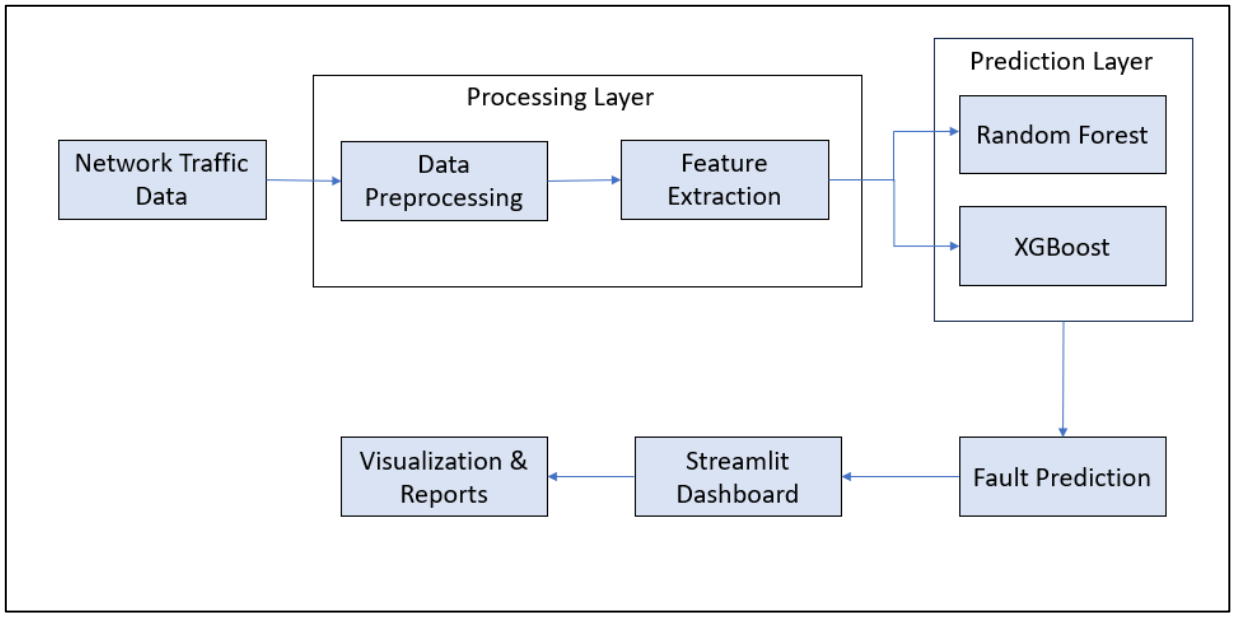


Fig. 1: Proposed Architecture for Application Performance and ISP Fault Prediction

B. *Design Considerations:*

- The proposed fault prediction model is trained and tested using Wireshark-captured network traffic datasets containing network performance metrics.
- The dataset includes both normal network behavior and faulty conditions related to ISP network issues and application-level performance degradation.
- Data preprocessing removes missing values, duplicate records, and inconsistent entries to improve data quality.
- Important network performance features such as latency, packet loss, throughput, jitter, retransmissions, packet count, maximum latency, and minimum latency are selected for analysis.
- Random Forest and XGBoost algorithms are employed due to their high classification accuracy and ability to handle large-scale network traffic data.
- The system classifies network conditions into three categories: Normal, Network Fault, and Application Fault.
- Prediction results are generated automatically based on extracted network performance metrics.
- A Streamlit-based dashboard provides real-time visualization, fault analysis, and performance monitoring.

C. *Description of the Proposed Algorithm:*

The proposed system aims to accurately identify the root cause of network performance degradation and classify whether the issue originates from ISP-related network faults or application-level problems. The algorithm consists of five major steps.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 1: Data Preprocessing:

The Wireshark network traffic dataset is loaded and preprocessed. Missing values, duplicate records, and irrelevant attributes are removed. The dataset is then normalized and transformed into a format suitable for machine learning analysis.

Let,

D = {x1, x2, x3, ..., xn}

where D represents the preprocessed network traffic dataset consisting of n traffic records.

Step 2: Feature Selection and Model Training:

Important network performance features are extracted from the dataset, including latency, packet loss, throughput, jitter, retransmissions, packet count, maximum latency, and minimum latency.

The extracted features are used to train Random Forest and XGBoost classification models.

Prediction = Mode(T1, T2, T3, ... , Tn) eq. (1)

where T1, T2, ... , Tn represent the predictions generated by individual decision trees.

Step 3: Network Condition Classification:

The trained machine learning model analyzes incoming network traffic records and classifies network conditions.

Fault Status = ML(X) eq. (2)

where ML represents the trained machine learning model and X represents the input feature vector.

If Fault Status = 0 → Normal

If Fault Status = 1 → Network Fault

If Fault Status = 2 → Application Fault

Step 4: Performance Analysis:

The system evaluates network performance using extracted metrics.

Performance Metrics = {Latency, Packet Loss, Throughput, Jitter, Retransmissions} eq. (3)

The analyzed metrics help identify the factors contributing to performance degradation and service quality issues.

Step5: Fault Prediction and Visualization:

After classification, the system generates fault prediction results and visualization reports.

Prediction Result = {Normal, Network Fault, Application Fault}

The Streamlit dashboard displays fault distributions, performance trends, prediction summaries, and downloadable reports.

These intelligent prediction and visualization mechanisms improve fault diagnosis accuracy, reduce troubleshooting time, and support proactive network management.

IV. PSEUDO CODE

Step 1: Start.

Step 2: Load Wireshark network traffic dataset.

Step 3: Perform data preprocessing.

- Remove missing values.
- Remove duplicate records.
- Handle inconsistent data.
- Normalize feature values.

Step 4: Split dataset into training and testing datasets.

- Latency
- Packet Loss
- Throughput
- Jitter
- Retransmissions
- Packet Count
- Maximum Latency
- Minimum Latency

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 5: Split dataset into training and testing datasets.
Step 6: Train Random Forest and XGBoost classifiers using training data.
Step 7: Evaluate model performance using.

- Accuracy
- Precision
- Recall
- F1-Score

Step 8 Upload new network traffic data for analysis.
Step 9: Apply trained model on input data.
Step10: Classify network condition.
IF Prediction = 0
Display "Normal"
ELSE IF Prediction = 1
Display "Network Fault"
ELSE
Display "Application Fault"
END IF
Step 11: Analyze performance metrics and identify root cause.
Step 12: Generate prediction report.
Step 13: Display results through Streamlit Dashboard.

- Fault Distribution
- Latency Analysis
- Packet Loss Analysis
- Throughput Analysis
- Prediction Results
- Performance Metrics

Step 14: Allow report download.
Step 15: End.

V. SIMULATION RESULTS

The Application Performance and ISP Fault Prediction System was developed with Python and assessed using network traffic data captured by Wireshark, which contained various metrics related to network performance. The dataset comprised entries that depicted typical network behavior, faults linked to ISPs, and issues at the application level. Techniques for data preprocessing and feature extraction were employed to derive significant attributes like latency, packet loss, throughput, jitter, retransmissions, packet count, as well as maximum and minimum latency. Machine learning algorithms, such as Random Forest and XGBoost, were utilized to train and test models for accurately categorizing network conditions into Normal, Network Fault, and Application Fault classes. The system's performance was evaluated using standard metrics, including Accuracy, Precision, Recall, and F1-Score. The experimental findings revealed a high level of classification accuracy and dependable fault prediction capabilities, facilitating the effective identification of the underlying causes of network performance issues.

To improve network monitoring and analysis, the suggested framework includes an interactive dashboard based on Streamlit in addition to fault prediction. Network performance data, fault distributions, latency trends, packet loss analysis, throughput fluctuations, and prediction outcomes are all displayed in real-time on the dashboard. Network traffic datasets can be uploaded, fault classification results can be viewed, and forecast reports can be downloaded for additional analysis. The overall findings show that in contemporary communication contexts, the suggested architecture successfully increases fault detection accuracy, shortens troubleshooting times, boosts network dependability, and facilitates proactive network performance management.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A. Evaluation and Interpretation of System Interfaces:

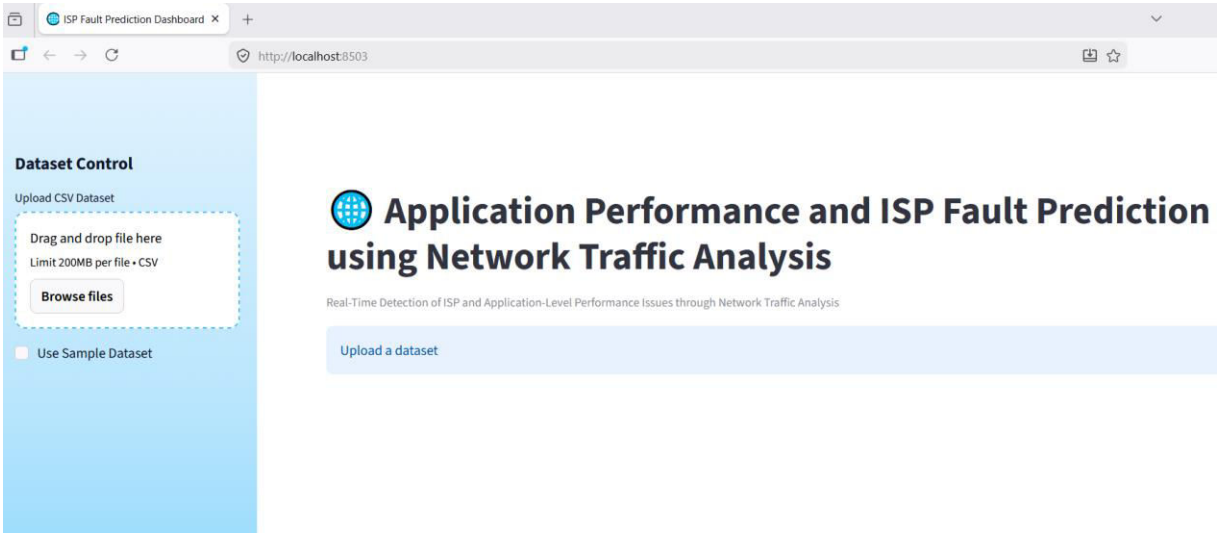


Fig.2. Home Page

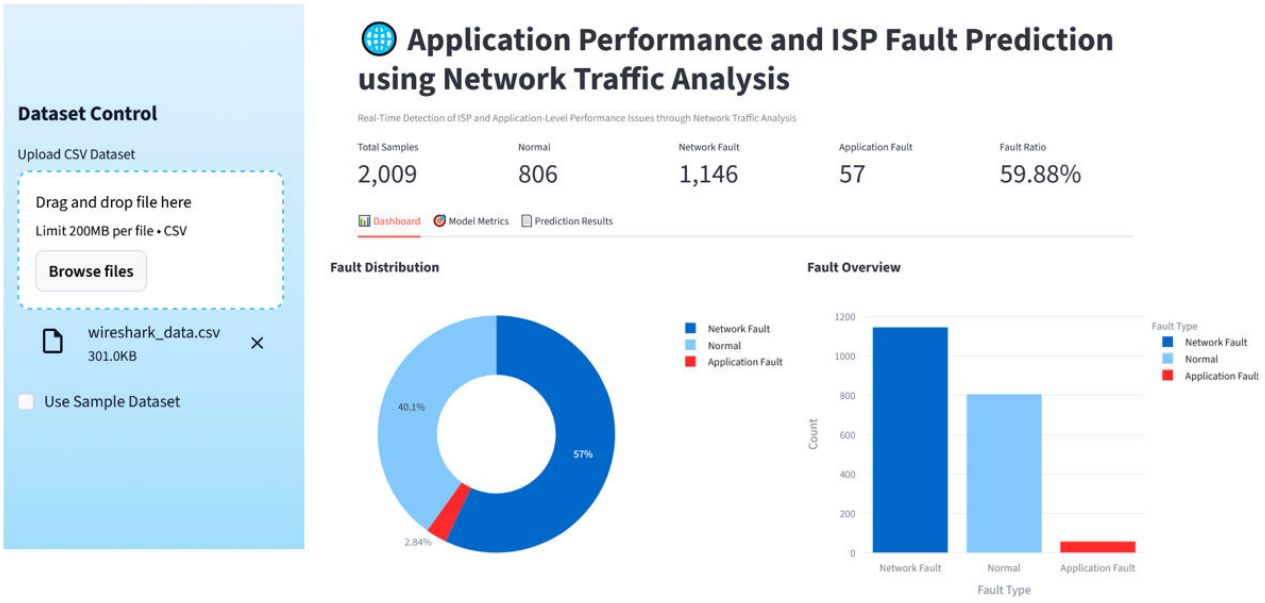


Fig.3. Fault Distribution and Fault Overview Analysis Dashboard

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

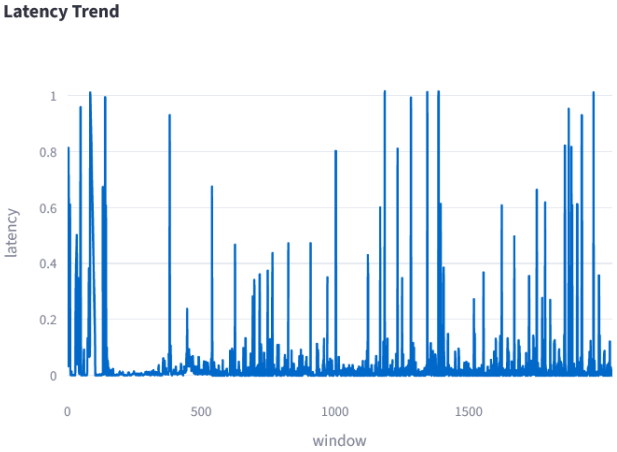


Fig.4. Latency Monitoring and Performance Analysis

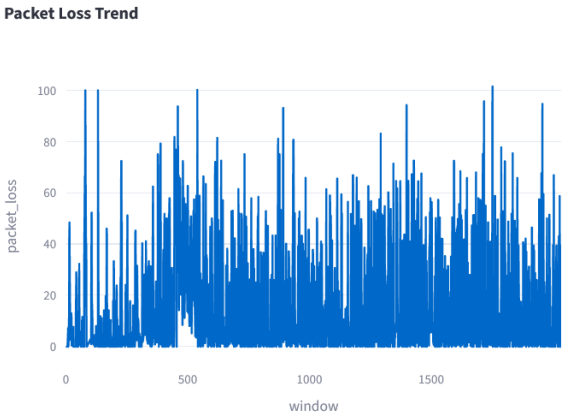


Fig. 5. Packet Loss Trend Analysis of Network Traffic Data

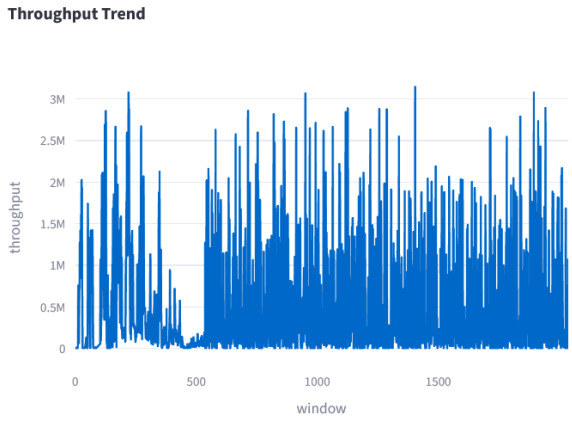


Fig.6 Throughput Trend

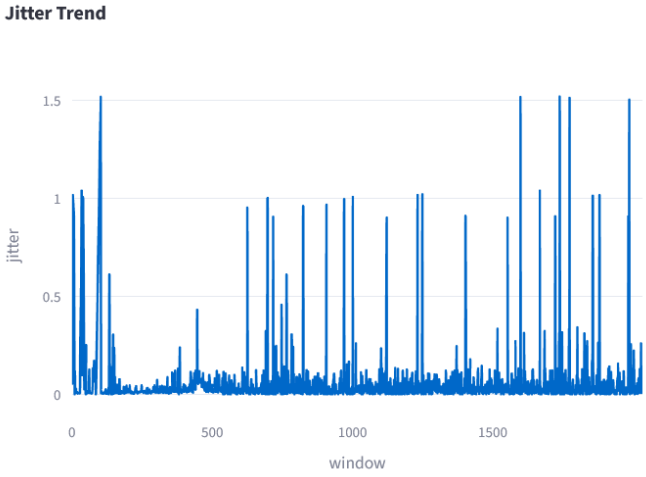


Fig. 7. Jitter Trend

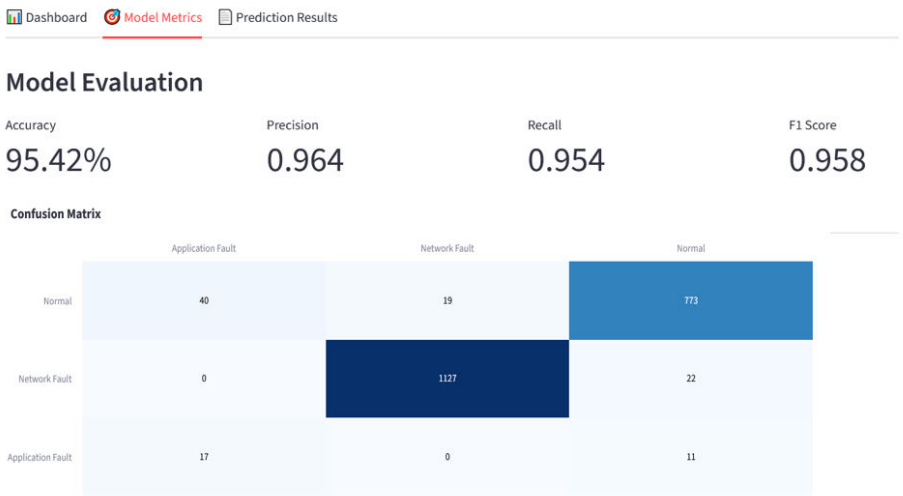


Fig. 8. Performance Evaluation Metrics of the Proposed Machine Learning Model

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The performance of the proposed Application Performance and ISP Fault Prediction System is evaluated using standard classification metrics including Accuracy, Precision, Recall, and F1-Score. These metrics help assess the effectiveness of the machine learning model in correctly classifying network conditions as Normal, Network Fault, or Application Fault.

Let:
TP = True Positives (correctly identified fault conditions)
TN = True Negatives (correctly identified normal conditions)
FP = False Positives (normal conditions incorrectly classified as faults)
FN = False Negatives (fault conditions incorrectly classified as normal)

Accuracy:
Accuracy represents the proportion of correctly classified instances to the total number of instances analyzed. It measures the overall effectiveness of the model in identifying network conditions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{eq. (5)}$$

Precision:
Precision is the ratio of correctly predicted fault instances to the total number of instances predicted as faults.

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{eq. (6)}$$

Recall :
Recall measures the proportion of actual fault instances that are correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{eq. (7)}$$

F1-Score:
The F1-Score is the harmonic mean of Precision and Recall and provides a balanced evaluation of the classifier's performance.

$$\text{F1Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{eq. (8)}$$

VI. CONCLUSION AND FUTURE WORK

The proposed Application Performance and ISP Fault Prediction System provides an intelligent approach for identifying and classifying network performance issues using machine learning and network traffic analysis. By analyzing Wireshark-captured traffic data and extracting key metrics such as latency, packet loss, throughput, jitter, and retransmissions, the system accurately distinguishes between Normal, Network Fault, and Application Fault conditions. The use of Random Forest and XGBoost algorithms improves prediction accuracy while reducing manual troubleshooting efforts. Furthermore, the Streamlit-based dashboard enables effective visualization and monitoring of network performance. Experimental results demonstrate the system's capability to enhance fault diagnosis, improve network reliability, and support proactive performance management. Future work includes integrating deep learning techniques, real-time traffic monitoring, automated alert generation, and larger datasets to further improve prediction accuracy, scalability, and practical deployment in real-world network environments.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

1. Parvat, T., Kolwankar, S., Lopes, N., Sawant, P., & Parchurkar, S., "Network Traffic Analysis Using Machine Learning," *Indian Journal of Computer Science*, Vol. 9, Issue 3, May–June 2024, pp. 32–41, DOI: 10.17010/ijcs/2024/v9/i3/174151.
2. Hu, X., & Zhou, Y., "Wireshark Packet Dissector for DL/T 860 Protocol," *2022 4th International Conference on Electrical Engineering and Control Technologies (CEECT)*, Shanghai, China, 2022, pp. 30–34, DOI: 10.1109/CEECT55960.2022.10030402.
3. Priya, N. S., Meyyappan, S., Balasubramanian, K. N., & Pruthiev, A. S., "Network Attack Detection Using Machine Learning," *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2022, pp. 342–346, DOI: 10.1109/ICACCS54159.2022.9785263.
4. Chauhan, R., & Kumar, S., "Packet Loss Prediction Using Artificial Intelligence Unified with Big Data Analytics, Internet of Things and Cloud Computing Technologies," *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, 2021, pp. 1–6, DOI: 10.1109/ISCON52037.2021.9702517.
5. Siswanto, A., Syukur, A., Kadir, E. A., & Suratin, "Network Traffic Monitoring and Analysis Using Packet Sniffer," *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, Rabat, Morocco, 2019, pp. 1–4, DOI: 10.1109/COMMNET.2019.8742369.
6. Arvind, S., Silveri, V. K., Potey, G., Nunavath, P., & Podishetty, R., "Network Traffic Virtualization Using Wireshark and Google Maps," *2023 International Conference on Distributed Computing, Electrical Circuits and Electronics (ICDCECE)*, India, 2023, pp. 1–6, DOI: 10.1109/ICDCECE57866.2023.10150823.
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.
8. McKinney, W., "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, 2010, pp. 56–61.
9. Hunter, J. D., "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, Vol. 9, No. 3, 2007, pp. 90–95.
10. Chen, T., & Guestrin, C., "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details