



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

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



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

AI Powered Network Traffic Prediction and Proactive QoS Optimization using Digital Twin

M. Divya Sri, R. Harshini, Dr. S. Senthil Kumar

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Madurai,
Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Madurai,
Tamil Nadu, India

Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Madurai,
Tamil Nadu, India

ABSTRACT: Network congestion can cause increased latency, packet loss, jitter, and reduced throughput, negatively affecting the quality of service (QoS) experienced by users. This paper presents an AI-powered framework for network traffic prediction and proactive QoS optimization using a digital twin. Network traffic is captured using Wireshark and processed with Python and PyShark to derive traffic and QoS-related metrics. The cleaned observations are organized into a dataset and stored for historical analysis using MySQL. A time-series learning model, such as a Long Short-Term Memory (LSTM) network, is intended to learn traffic patterns and predict possible future QoS degradation. The predicted condition is then represented in a digital-twin layer, where candidate optimization actions can be examined before recommendations are presented through a web dashboard. The proposed workflow connects real-network observations, historical data, prediction, simulation, and proactive decision support. Quantitative performance claims are not made here; prediction accuracy and the effectiveness of optimization actions must be established through systematic evaluation on the collected dataset.

KEYWORDS: Network traffic prediction, Quality of Service (QoS), Digital twin, Wireshark, PyShark, LSTM, congestion prediction, proactive optimization.

I. INTRODUCTION

Modern computer networks support applications that require reliable and timely data delivery, including video communication, cloud services, online learning, and real-time monitoring. When traffic demand approaches or exceeds available network capacity, congestion may occur. Its effects can include higher latency, increased packet loss and jitter, and reduced throughput. Reactive troubleshooting detects these issues after service quality has already degraded. A predictive approach can instead identify patterns that may indicate an upcoming QoS problem and give network operators time to consider preventive actions.

This project, “AI Powered Network Traffic Prediction and Proactive QoS Optimization using Digital Twin,” proposes a data-driven workflow for anticipating network degradation. Wireshark is used to capture traffic from a real network. Python and PyShark support packet-data processing and extraction of relevant observations. After cleaning and metric calculation, the dataset is saved for historical analysis in MySQL. A machine-learning time-series model is planned to predict future traffic conditions, while a digital-twin layer provides a representation in which potential responses can be examined before recommendations are displayed.

The main objectives are: (i) collect and preprocess network traffic observations, (ii) derive useful traffic and QoS indicators, (iii) build a historical dataset suitable for time-series prediction, (iv) predict possible congestion or QoS degradation, and (v) connect predictions to a digital-twin-based decision-support workflow. The remainder of this paper reviews related concepts, describes the proposed methodology, discusses the current implementation workflow and evaluation plan, and concludes with future work.

II. LITERATURE SURVEY

Network performance measurement provides the foundation for assessing service quality. IETF specifications define metrics for packet loss and IP packet delay variation, supporting consistent interpretation of loss and delay-related behaviour [1], [2]. These measures motivate monitoring multiple indicators rather than relying on a single traffic statistic. Throughput, latency, jitter, packet loss, bandwidth utilization, and traffic load can provide complementary evidence of changing network conditions.

Machine learning has been applied to time-series forecasting in domains where the next state depends on patterns across previous observations. Long Short-Term Memory (LSTM), introduced by Hochreiter and Schmidhuber, is a recurrent neural-network architecture designed to learn dependencies over sequences [3]. For network prediction, sequential records can represent traffic volume or QoS metrics over time. Model performance depends on capture quality, the chosen prediction horizon, feature selection, and a timeaware evaluation procedure.

Digital-twin concepts provide a virtual representation of a physical system that can be used to observe state, explore scenarios, and support decisions. In a network-management setting, a digital twin can combine measured network conditions with predicted future states and allow candidate actions to be assessed before they are recommended for the live network. The proposed project combines this idea with packet-derived observations and AI-based forecasting. Unlike a system that only reports current measurements, the intended workflow aims to connect historical data, futurecondition prediction, and proactive recommendations.

A key practical consideration is that predictions should be evaluated against observations that occur later in time than the training data. Randomly mixing adjacent time-series records between training and testing can produce overly optimistic results. Therefore, this project should use chronological train, validation, and test partitions and compare its chosen model against a simple baseline before drawing conclusions about predictive benefit.

III. METHODOLOGY

The proposed system is organized as a pipeline: Real Network → Wireshark Capture → Raw Network Data → Python/PyShark Processing → Data Cleaning and QoS Metrics → MySQL Historical Dataset → AI-Based Prediction → Digital Twin and Scenario Analysis → Optimization Recommendation → Web Dashboard.

A. Network Traffic Collection

Traffic is captured from the target network using Wireshark. Captures should cover different periods and, where possible, different traffic conditions so that the dataset represents more than one operating state. Capture duration, sampling interval, interface, and any filtering rules should be recorded to make later analysis reproducible. Packet captures may contain sensitive information; collection and storage should therefore be limited to authorized traffic and only the fields needed for analysis.

B. Preprocessing and Metric Extraction

Python and PyShark are used to read the capture data and convert relevant packet observations into structured records. The preprocessing stage checks field consistency, handles missing or invalid values, standardizes data types, and organizes observations into time intervals. Candidate features include packet count, bytes transferred, traffic rate, bandwidth utilization, latency, jitter, packet loss, throughput, and a derived network-load indicator. The exact availability and calculation of each metric depend on the fields present in the captures and on whether suitable timing and packet-pair information are available. Metrics that cannot be measured reliably from the available capture should be excluded or explicitly marked as unavailable rather than estimated without evidence.

C. Dataset Storage and Historical Data

After cleaning and metric calculation, records are assembled into a time-indexed dataset and stored in MySQL for retrieval and analysis. Each record should include a timestamp or time-window identifier and the selected features. Historical data can be accumulated through additional authorized captures collected at different times, then processed through the same pipeline so that feature definitions remain consistent. Duplicate records, inconsistent units, missing timestamps, and capture gaps should be checked before model training.

D. Traffic and QoS Prediction

The prediction stage uses a sequence of previous time-window observations to estimate a future target, such as traffic load or a QoS indicator. An LSTM model is a candidate because it can learn temporal relationships from ordered data. The input sequence length, prediction horizon, feature scaling, model architecture, and training parameters must be selected through validation rather than assumed in advance. The model should be trained only on the training period, with preprocessing parameters fitted on that period and then reused for validation and testing. A persistence or movingaverage baseline should be included to determine whether the learned model improves on a simple forecast.

E. Digital Twin and Proactive Recommendation

The digital-twin layer represents the monitored network state using current metrics and predicted future conditions. When a prediction indicates a possible degradation, the system can evaluate candidate responses such as reviewing traffic distribution, checking available capacity, prioritizing critical application traffic, or investigating a potential bottleneck. These are decision-support recommendations; they should not automatically change a production network unless the action is separately validated and explicitly authorized. The dashboard is intended to display recent metrics, predicted trends, warning conditions, and recommended next steps in an understandable form.

F. Evaluation Plan

The predictive component should be evaluated with a chronological holdout set. For continuous targets, suitable measures include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). If congestion is formulated as a classification task, precision, recall, F1-score, and a confusion matrix can be reported. Evaluation should state the prediction horizon, dataset size and period, feature set, baseline, and model settings. The optimization component should separately assess whether recommendations correspond to improved simulated or measured QoS; prediction accuracy alone does not demonstrate that optimization is effective.

IV. EXPERIMENTAL RESULTS

The project workflow has reached the data-cleaning and dataset-creation stages: traffic observations captured with Wireshark have been processed, metrics have been calculated, and a dataset has been created and saved. The next stage is to combine observations collected at different times into a consistent historical dataset and use it to train and evaluate a future-congestion prediction model. The proposed database layer is MySQL.

At this stage, no verified numerical prediction scores, baseline comparison, or measured QoS improvement values are available to report. Accordingly, this paper does not claim a particular accuracy percentage or a proven reduction in congestion. The experimental evaluation should be completed after the historical dataset and prediction pipeline are finalized. Results should include a dataset summary, chronological split details, MAE/RMSE or classification metrics as appropriate, comparison with a baseline, and representative examples showing predicted versus observed values.

A further evaluation step is to test the digital-twin workflow using clearly defined scenarios, such as increased traffic load or a predicted rise in delay. For each scenario, the study should record the initial state, the proposed recommendation, the simulated or measured outcome, and any constraints. This separates the evidence for forecasting from the evidence for proactive optimization and avoids treating a recommendation as successful without measurement.

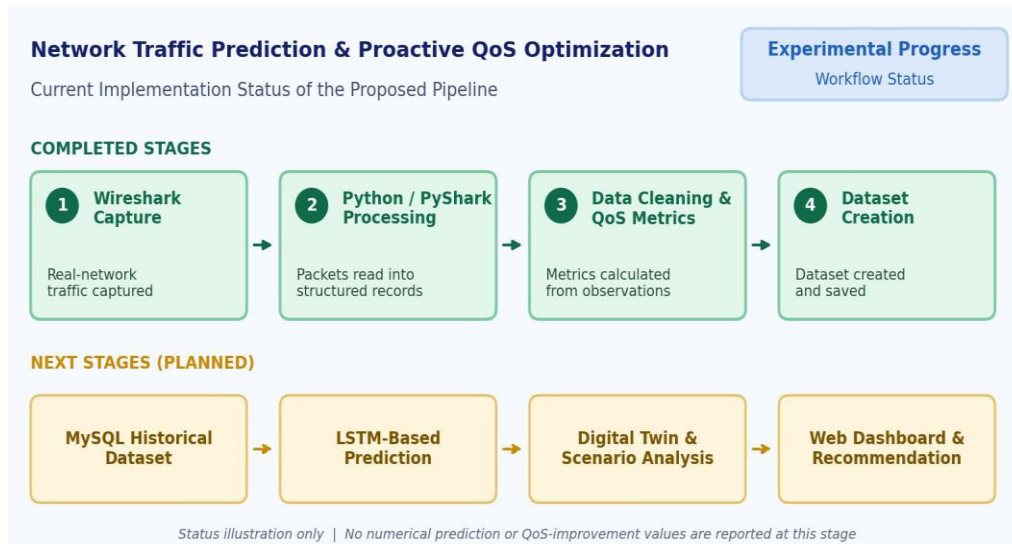


Fig.1. Current implementation status of the proposed workflow (illustrative; no measured values)

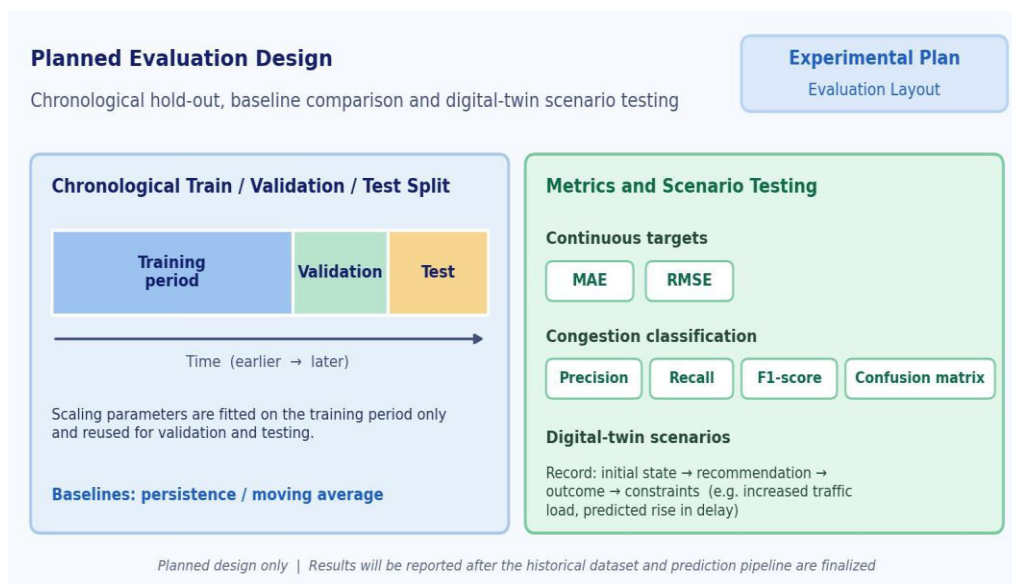


Fig.2. Planned evaluation design (illustrative; no measured values)

V. CONCLUSION

This paper presented a proposed framework for AI-powered network traffic prediction and proactive QoS optimization using a digital twin. The workflow integrates Wireshark-based traffic capture, Python/PyShark preprocessing, QoS metric extraction, MySQL historical storage, time-series prediction, digital-twin scenario analysis, and dashboard-based recommendations. The data preparation and dataset-creation stages form the basis for the next phase: building a consistent historical dataset and evaluating a temporal prediction model. Future work will train and compare the selected model with simple baselines, report results on a chronological test set, and assess whether digital twin recommendations improve QoS under controlled scenarios. Quantitative claims will be made only after those experiments are completed.

REFERENCES

1. G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Packet Loss Metric for IPPM," RFC 2680, Internet Engineering Task Force, September 1999.
2. C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)," RFC 3393, Internet Engineering Task Force, November 2002.
3. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
4. W. Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud. AddisonWesley Professional, 2016. [5] Wireshark Foundation, "Wireshark User's Guide." Available: https://www.wireshark.org/docs/wsug_html_chunked/. Accessed October 2026.
5. PyShark contributors, "PyShark: Python wrapper for tshark, allowing Python packet parsing using Wireshark dissectors." Available: <https://github.com/KimiNewt/pyshark>. Accessed October 2026.
6. Oracle, "MySQL 8.4 Reference Manual." Available: <https://dev.mysql.com/doc/refman/8.4/en/>. Accessed October 2026.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details