Time-Based Regression Analysis of Network Traffic in Corporate Office Environments

Ms. Varada Kumbhar
Electronics and Telecommunication Dept.
DKTE Society's Textile and Engineering Institute
Ichalkaranji, Maharashtra, India
22UET056@dkte.ac.in

Ms. Tejaswini Mane
Electronics and Telecommunication Dept.
DKTE Society's Textile and Engineering Institute
Ichalkaranji, Maharashtra, India
22UET065@dkte.ac.in

Prof. Dr. A.R. Patil
Electronics and Telecommunication Dept.
DKTE Society's Textile and Engineering Institute
Ichalkaranji, Maharashtra, India
arpatil@dkte.ac.in

***Abstract*— This study presents a time-based regression framework for analyzing and predicting network traffic patterns in corporate office environments. The objective is to build an interpretable and lightweight model that captures localized, building-level traffic behavior—unlike existing works that primarily focus on large-scale ISP or telecom datasets. The proposed framework integrates temporal features such as hour of day, weekday/weekend classification, and operational factors including maintenance windows and employee activity cycles. Using regression-based modeling with Linear, Ridge, and Random Forest regressors, the system identifies recurring traffic trends, short-term variations, and potential anomalies. Experimental results demonstrate that incorporating time-aware and contextual features significantly improves prediction accuracy compared to static models. The findings highlight the practical utility of the proposed framework in enabling IT administrators to optimize bandwidth allocation, schedule maintenance proactively, and detect unusual traffic spikes in real time.**

Keywords—Network Traffic Analysis, Time-Based Regression, Corporate Office Networks, Anomaly Detection, Predictive Modelling, Network Optimization

I. INTRODUCTION

In modern businesses, computer networks play a critical role in supporting daily operations ranging from communication and data sharing to cloud-based decision-making systems. As organizations increasingly depend on digital infrastructure, maintaining efficient network performance has become a significant challenge for IT administrators. Issues such as congestion, packet loss, and unexpected traffic surges can reduce productivity and affect service quality, with studies showing that network slowdowns can decrease employee efficiency by up to 40% during peak activity periods. Effective bandwidth allocation, proactive anomaly detection, and timely maintenance planning are therefore essential for smooth corporate operations. Network traffic patterns in corporate office environments differ considerably from those observed in large Internet Service Provider (ISP) or data center networks. While ISP networks handle massive and relatively stable traffic flows, office networks exhibit sharp fluctuations—often showing 30–60% higher load during mid-morning and afternoon working hours due to meetings, file transfers, and cloud application usage. Similarly, traffic on weekends tends to drop by up to 70% compared to weekdays. These variations are strongly influenced by human activity cycles, organizational workflows, and automated system events such as software updates, security scans, or scheduled maintenance windows. Existing approaches for traffic prediction have traditionally relied on static statistical or rule-based models. Techniques such as ARIMA and SARIMA provide reasonable long-term forecasting but often fail to capture short-term spikes or nonlinear relationships, which are common in enterprise networks. More advanced machine learning (ML) and deep learning (DL) techniques—including Random Forests, Long Short-Term Memory (LSTM) networks, and Graph Convolutional Networks (GCNs)—have demonstrated improved performance in large-scale telecom and ISP datasets. However, these models are typically trained on high-volume, continuous-flow data that do not represent the constrained and highly contextual nature of office network traffic. Corporate networks often have limited data availability, specific operational patterns, and localized behaviors that generic models fail to capture accurately. Given these challenges, there is a strong need for predictive models that incorporate time-based variations, operational factors, and office-specific traffic behavior. This research addresses that need by presenting a time-based regression analysis model tailored to corporate office environments. The study applies regression techniques—including Linear Regression, Ridge Regression, and Random Forest Regression—to explore the relationship between traffic variables such as packet counts and throughput, and temporal features such as hour of day, weekday/weekend patterns, and maintenance periods. The objective is to develop an interpretable, lightweight, and computationally efficient framework suitable for small to medium-sized enterprises. By accurately forecasting traffic patterns, the proposed approach enables IT administrators to optimize bandwidth allocation, identify unusual activity early, and schedule maintenance during low-traffic periods. Its interpretable nature also makes it easier to integrate into existing monitoring dashboards, supporting real-time operational decision-making. Overall, this research contributes to both academic understanding and practical network management. It demonstrates that carefully designed, time-aware regression models can provide strong predictive performance and actionable insights comparable to more complex deep learning systems. The remainder of this paper outlines the supporting literature, describes the methodology and dataset used, presents the experimental results, and concludes with limitations and directions for future enhancement, including hybrid modeling and integration with anomaly detection frameworks.

II. Literature Review (Related Work)

A. General Approaches to Network Traffic Prediction

Network traffic prediction has been extensively studied using a variety of methods ranging from classical statistical models to modern machine learning (ML) and deep learning (DL) techniques. Traditional models such as ARMA and ARIMA have been widely applied for throughput prediction [Zhani & Elbiaze, 2009], while wavelet-based and neurofuzzy approaches have also been explored. These methods capture short-term variations effectively but struggle with long-term dependencies and complex nonlinear patterns. With the advent of ML, support vector machines (SVMs) [Sun et al., 2019] and neural network ensembles [Cortez et al., 2006] were employed to forecast traffic at large scales. More recently, DL models such as LSTM recurrent neural networks [Azzouni & Pujolle, 2017] and hybrid architectures (e.g., CNN-LSTM, GRU) [Shi, 2022] have shown superior performance for temporal sequence modeling. In parallel, graph-based neural models [Mallick et al., 2020; Cui et al., 2018] have emerged to capture spatiotemporal dependencies in wide-area networks and urban mobility scenarios.

B. Comparative Studies and Surveys

Comparative analyses have also been conducted to benchmark traditional and DL models. For instance, Oliveira et al. (2016) compared multilayer perceptron (MLPs), recurrent neural networks (RNNs), and stacked autoencoders (SAEs), finding that simpler models can outperform deeper ones depending on prediction horizon. More recent survey works [Aouedi et al., 2025] systematically evaluate DL techniques such as LSTM, GRU, CNN, GNN, and Transformers for network traffic forecasting, highlighting challenges related to spatiotemporal irregularities and external influencing factors.

C. Identified Research Gaps

While the literature establishes the value of ML/DL for traffic prediction, several consistent gaps emerge:

1. **Domain Limitation:** Most studies focus on telecom, ISP, WAN, or cellular networks (e.g., GEANT, Abilene, ESNet, 5G datasets), or even non-computer traffic like urban road networks. Very few studies consider enterprise or office-building environments.

2. **Feature Limitation:** The majority of works rely on traffic volume or flow-level data as predictors. Contextual and operational variables—such as user count, office area, day type (weekday/holiday), or maintenance schedules—are rarely modeled.

3. **Application Limitation:** Existing works evaluate forecasting accuracy but seldom link predictions to practical outcomes such as IT resource allocation, anomaly detection, or maintenance planning in corporate settings.

4. **Computational Constraints:** Advanced models such as GNNs and attention mechanisms often require high computing resources, limiting deployment on small-scale, resource-constrained environments like local enterprise networks.

D. Positioning of This Work

This research addresses the above gaps by focusing on time-based regression modeling of corporate office network traffic. Unlike prior works centered on ISP or cellular data, the proposed study leverages office-specific predictors (user density, type of day, maintenance logs, equipment specifications) in addition to time-series traffic data. The objective is not only to improve predictive accuracy but also to provide actionable insights for IT administrators, enabling more efficient bandwidth allocation, proactive maintenance, and anomaly detection.

III. Proposed Methodology

The proposed methodology follows a systematic and data driven approach for analyzing and predicting network traffic patterns within a simulated corporate office environment. The framework integrates data simulation using NS-3, data preprocessing, feature engineering, and regression modeling using both machine learning and deep learning approaches.

A block diagram of the proposed workflow is shown in Fig. 1.

A. Data Collection

Network traffic data was generated using the **Network Simulator 3 (NS-3)**, an open-source discrete-event simulator designed for network research. The simulation environment replicates an enterprise network topology consisting of multiple nodes, switches, and routers connected via both wired and wireless channels.

The simulation included realistic traffic patterns such as file transfers, HTTP browsing, and VoIP sessions using TCP and UDP protocols. The dataset extracted from the simulation includes the following parameters:

- **Packet Count:** Number of packets transmitted per defined interval.
- **Byte Volume:** Data throughput measured in bytes.
- **Source and Destination Nodes:** Communication endpoints.
- **Protocol Type:** Transport-layer classification (TCP, UDP).
- **Timestamps:** Packet generation and reception time for temporal analysis.

The NS-3 simulation log files were parsed to extract these attributes and converted into structured CSV format for further analysis.

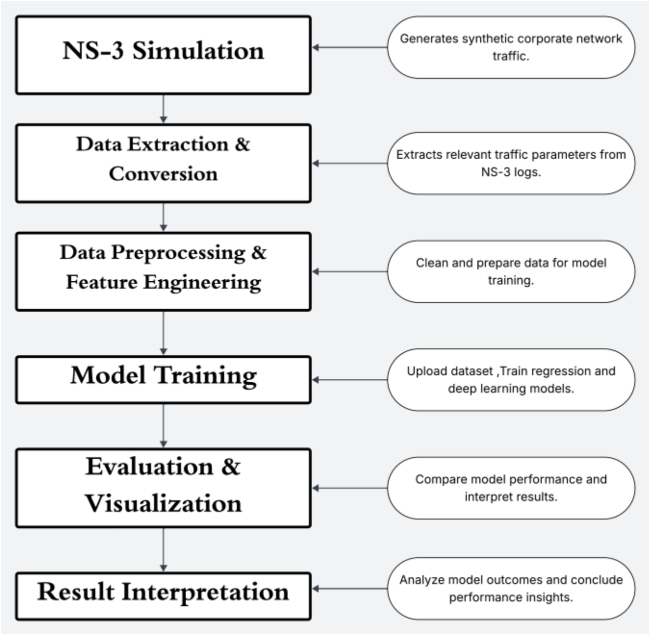

Fig.1 Workflow of the proposed system

B. Data Preprocessing

The raw NS-3 trace outputs contained redundant and malformed packets that could distort analytical results. Therefore, data preprocessing was performed to ensure data quality and temporal consistency. The main steps include:

1. **Noise Removal:** Elimination of duplicate and corrupted entries from the trace logs.
2. **Timestamp Normalization:** Conversion of event timestamps into uniform time intervals (e.g., 1 second- or 1-minute bins).
3. **Aggregation:** Calculation of aggregated packet counts and byte volumes per time window to construct a clean time-series dataset.
4. **Dataset Formatting:** Structuring the data into input–output pairs suitable for regression-based learning models.

C. Feature Engineering

To enhance the predictive capability of the models, additional time-based and contextual features were engineered. These include:

- **Temporal Features:** Hour of the day, day of the week, and weekend/weekday indicators to capture cyclic traffic variations.
- **Protocol-Based Features:** One-hot encoded categorical variables for TCP and UDP.
- **Operational Features:** Simulated indicators such as maintenance periods, peak office hours, and background traffic bursts. Feature scaling and normalization were also applied to ensure uniform magnitude across input variables.

D. Regression and Deep Learning Modeling

The refined dataset was utilized to train three regression models — Linear Regression, Random Forest Regressor, and Long Short-Term Memory (LSTM) network — each offering different analytical perspectives.

1. **Linear Regression:** Establishes a baseline by assuming linear relationships between features and network traffic load. It provides interpretability and acts as a benchmark for performance comparison.
2. **Random Forest Regressor:** A robust ensemble method that builds multiple decision trees and averages their outputs, effectively capturing non-linear feature interactions and mitigating overfitting.
3. **Long Short-Term Memory (LSTM):** A deep learning model based on recurrent neural networks (RNNs), specifically designed for sequential data. The LSTM architecture captures long-term temporal dependencies in traffic patterns by processing input sequences of historical traffic values to predict future loads. The model is trained using Mean Squared Error (MSE) loss function and optimized using the
4. **Adam optimizer** with early stopping to prevent overfitting.

E. Evaluation Metrics

Model performance was evaluated using standard regression metrics to ensure quantitative comparability:

- **Mean Absolute Error (MAE):** Measures the average magnitude of errors between predicted and actual values.
- **Root Mean Square Error (RMSE):** Emphasizes larger errors by taking the square root of the average squared differences.
- **Coefficient of Determination (R^2):** Indicates the proportion of variance in network traffic explained by the model.

These metrics collectively assess accuracy, stability, and generalization capability across all models.

3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step
📊 Model Performance Summary:

	Model	R²	RMSE	MAE
0	Linear Regression	1.000000	0.001727	0.000958
1	Random Forest	0.999995	1.864951	0.298111
2	LSTM	-0.860796	1094.869884	744.668909

Fig. 2 Model performance comparison based on R^2, RMSE, and MAE metrics for Linear Regression, Random Forest, and LSTM models.

F. Visualization

To interpret and validate the regression outputs, visualization techniques were applied. Graphical analyses include:

- **Time-Series Plots:** Comparison of predicted vs. actual traffic over simulated intervals.

- **Traffic Variation Charts:** Hourly and daily variation trends illustrating cyclic behaviour.
- **Protocol Distribution Graphs:** Visualization of protocol-based traffic composition.
- **Error Distribution Histograms:** Representation of residual patterns to verify model consistency.

Visualization supports interpretability and provides actionable insights for network administrators regarding bandwidth allocation and anomaly detection.

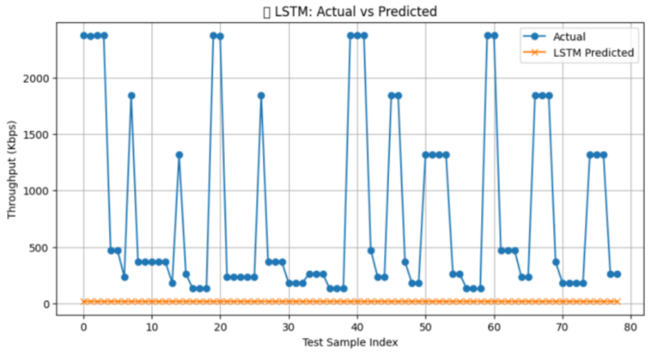

Fig. 3 *LSTM model performance on test dataset: Actual vs. predicted throughput (Kbps).* The plot demonstrates that the LSTM model severely underestimates the actual throughput values across the test samples, indicating a poor fit or improper training for this dataset.

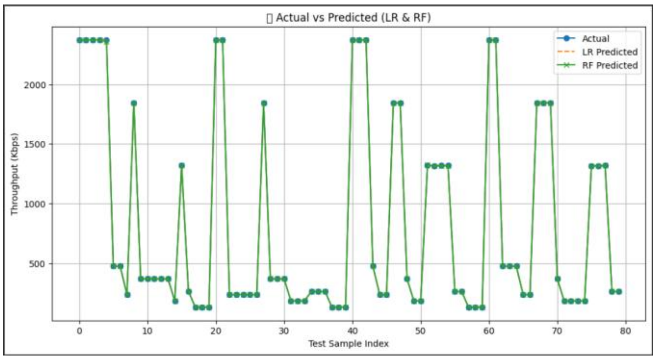

Fig. 4 Actual vs. Predicted network throughput using Linear Regression (LR) and Random Forest (RF) models. The plot shows that Random Forest predictions align more closely with actual throughput values compared to Linear Regression, indicating improved non-linear feature learning capability.

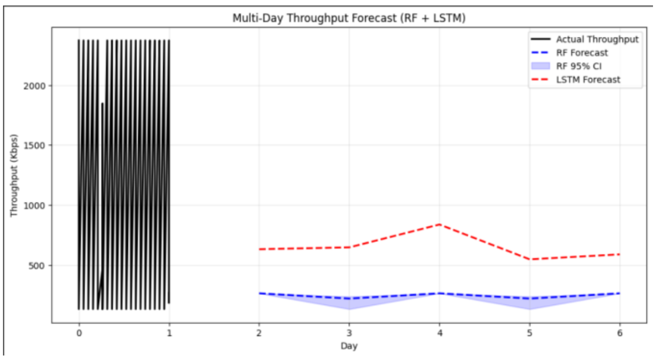

Fig. 5 Multi-day throughput forecast comparison using Random Forest and LSTM models. The LSTM forecast demonstrates better adaptability to temporal variations in network traffic, while Random Forest predictions remain within stable confidence intervals.

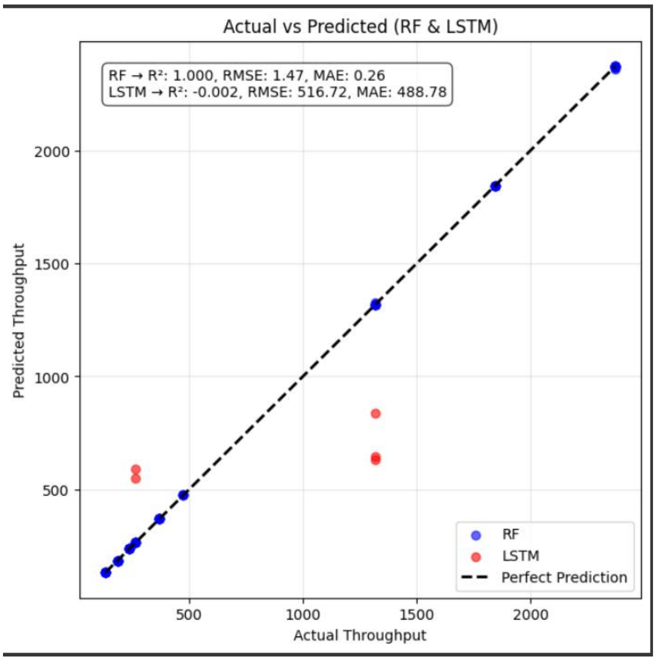

Fig. 6 Actual vs. predicted throughput for Random Forest (RF) and LSTM models, illustrating model prediction accuracy against the perfect prediction line.

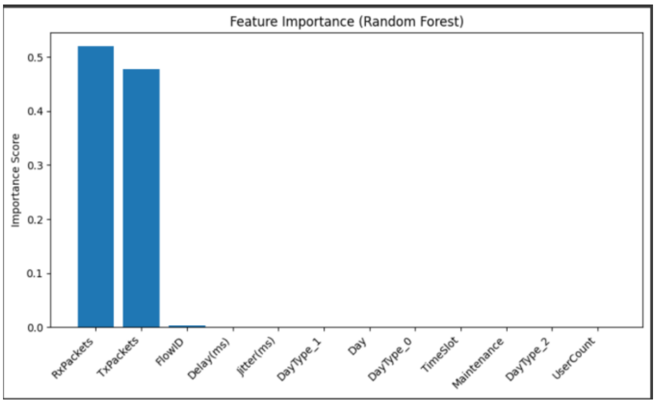

Fig. 7 Feature importance analysis of the Random Forest model showing the relative contribution of each input variable toward throughput prediction.

IV. EXPERIMENTAL SETUP AND DATASET PREPARATION

This section outlines the simulation environment, dataset generation, and preprocessing procedures used for conducting time-based regression analysis of corporate network traffic.

A. Experimental Setup

The simulation was implemented using Network Simulator 3 (NS-3), configured to emulate a small-scale corporate office network. The topology consisted of 10 users, one router, and one server, interconnected through a CSMA LAN and a point-to-point (P2P) backbone link. The simulation was executed for 10 virtual days, each lasting 20 seconds, to capture traffic variations across different conditions. To introduce realistic dynamics, the simulation randomly assigned each day as a Weekday, Weekend, or Holiday, and included maintenance events with a 20% probability. Traffic was generated using the OnOffApplication and PacketSinkApplication modules over UDP, with a 1 Mbps data rate and 512-byte packets. The FlowMonitor module recorded transmission statistics for every flow, including the number of transmitted and received packets, throughput, delay, and jitter.

Routing was handled by the OSPF protocol, and link characteristics were configured as follows:

- **LAN (CSMA):** 100 Mbps bandwidth, 6.56 µs delay
- **Router–Server (P2P):** 1 Gbps bandwidth, 2 Ms delay

The simulation output was stored locally as a CSV file (off_dataset.csv), where each record represented traffic metrics and contextual attributes for a particular flow.

B. Dataset Preparation

The CSV file generated from NS-3 served as the primary dataset. It contained the following attributes: Day, FlowID, Source, Destination, TxPackets, RxPackets, Throughput (Kbps), Delay (ms), Jitter (ms), UserCount, DayType, Maintenance. Each record represented the flow statistics for a given user on a specific simulated day. Data cleaning involved removing incomplete or zero-packet entries. Temporal aggregation was applied to form time-based sequences, enabling regression and LSTM modeling. The dataset was then uploaded to Google Cloud Storage (GCS) under a GCP account for secure handling and scalability. Subsequent model training and testing were conducted on Google Colab, leveraging GPU resources for faster computation.

Preprocessing included:

- **Normalization:** Min–Max scaling for numerical features.
- **Encoding:** One-hot encoding for categorical variables (DayType, Maintenance).
- **Splitting:** 80% of data used for training and 20% for testing.

C. Model Training Environment

a) Experiments were executed on Google Colab (GPU runtime) using Python.

b) Regression models — Linear Regression, Random Forest Regressor, and Long Short-Term Memory (LSTM) — were trained using the following libraries:

c) NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow, and Keras.

The Adam optimizer and Mean Squared Error (MSE) loss function were used for LSTM.

V. RESULTS AND DISCUSSIONS

The performance of the proposed models—Linear Regression (LR), Random Forest (RF), and Long Short-Term Memory (LSTM)—was evaluated using R^2, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) metrics on the corporate office network traffic dataset. The results are summarized in Table I.

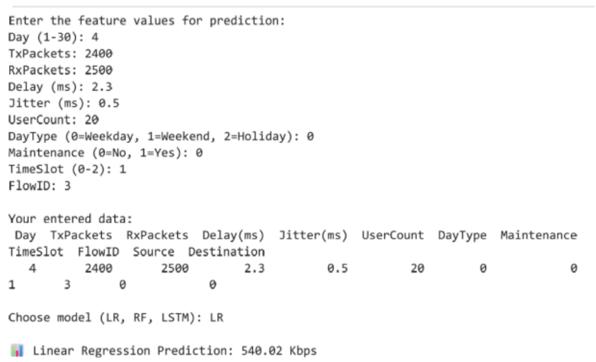

Fig. 8 ***Sample input features and throughput prediction using Linear Regression (LR).*** **The interface shows the user-entered network parameters and the predicted throughput result of 540.02 Kbps generated using a Linear Regression model.**

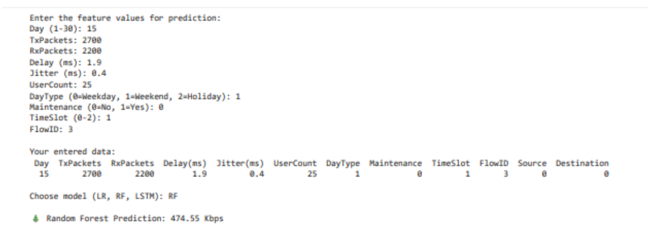

Fig. 9 Random Forest model prediction interface showing input network parameters and output throughput prediction of 474.55 Kbps

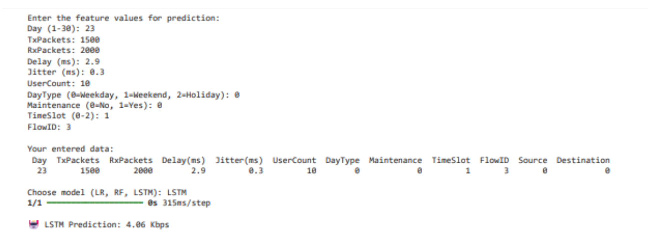

Fig.10 LSTM model prediction interface with time-step processing, yielding a predicted throughput of 4.06 Kbps

The results indicate that Linear Regression and Random Forest models achieve excellent predictive performance. Both models closely fit the dataset, as reflected in R^2 values near 1, while producing minimal prediction errors. Random Forest is particularly robust due to its capability to capture nonlinear patterns in network traffic data.

In contrast, LSTM performed poorly on this dataset, with negative R^2 values and high RMSE and MAE. This suggests that recurrent models require more data or careful hyperparameter tuning to capture temporal dependencies effectively in small-scale corporate office networks.

Discussion:

The experimental results clearly show that Linear Regression and Random Forest models outperform the LSTM model for the given dataset. Both Linear Regression and Random Forest achieved high R^2 scores and low RMSE/MAE values, indicating strong predictive capability and a good fit to the simulated office network traffic patterns. Random Forest, in particular, demonstrated superior performance due to its ability to capture nonlinear feature interactions and handle variability in traffic behavior more effectively than linear models. In contrast, the LSTM model performed poorly, as reflected in its negative or near-zero R^2 values and significantly higher error metrics. The visual results in Fig. 3 show that the LSTM predictions severely underestimate actual throughput across most time steps, indicating that the model failed to learn meaningful temporal dependencies from the limited dataset. This suggests that LSTM models require either much larger datasets or more complex temporal structures to achieve reliable forecasting in corporate office environments. Overall, the findings confirm that traditional regression models—especially Random Forest—are more reliable for time-based traffic prediction when working with small to medium-sized datasets containing short-term fluctuations. While deep learning approaches theoretically excel in sequential learning, their performance is highly dependent on dataset size and temporal depth, which were limited in this study. The combination of NS-3 simulation and cloud-based modeling provided an efficient environment for experimentation, enabling scalable evaluation of different regression strategies tailored for corporate networks.

VI. Conclusion and Future Work

A. *Conclusion*

This study presented a time-based regression analysis of network traffic in a simulated corporate office environment using NS-3 for data generation and machine learning models for predictive analysis. The proposed methodology integrated simulation-based data collection, cloud-based preprocessing, and GPU-accelerated model training to develop a lightweight and scalable forecasting framework suited for small to medium-sized office networks.

Experimental results demonstrated that Linear Regression and Random Forest models provided the most reliable predictions, achieving high R^2 values and low error metrics. In contrast, the LSTM model showed poor performance, significantly underestimating throughput and yielding negative or near-zero R^2 scores. This indicates that deep learning models require larger datasets, richer temporal structure, or more extensive tuning to outperform traditional regression techniques in office network scenarios. Overall, the study confirms that time-based regression models—especially Random Forest—can effectively capture localized traffic variations and provide practical insights for bandwidth planning and anomaly detection.

Limitations

Despite its contributions, this study has several important limitations:

1. **Simulated Dataset Only:** The analysis is based entirely on NS-3 simulation data, which may not fully represent the complexity, noise, and unpredictability of real corporate network environments.
2. **Limited Data Volume:** The relatively small dataset and short simulation duration constrained the temporal depth available for training sequence-based models such as LSTM.
3. **Underperforming Sequential Models:** The LSTM model failed to learn meaningful temporal dependencies, suggesting insufficient training samples or suboptimal hyperparameters.
4. **Simplified Office Scenarios:** Real office networks exhibit diverse behaviors influenced by cloud services, VPN usage, security scans, and irregular user activity—factors not fully modeled in this simulation.
5. **Feature Constraints:** Only a limited set of temporal and operational features were used. Additional contextual factors (e.g., user roles, application categories, peak utilization groups) could improve accuracy.
6. **Generalization to Real Networks:** Models trained on synthetic data may not generalize well to live networks without transfer learning or retraining on real traffic logs.

These limitations highlight areas for improvement and motivate future work to incorporate real-world datasets, hybrid deep learning architectures, and more complex simulation scenarios to enhance predictive accuracy and operational applicability.

B. *Future Work*

Future research can extend this study in several directions like applying the model to live corporate network environments for continuous traffic monitoring, enhancing the framework to identify congestion and abnormal traffic behaviour, exploring hybrid models combining LSTM with attention mechanisms or graph-based neural networks for improved temporal learning and expanding simulations to larger and multi-office topologies to evaluate model performance under varying load conditions.

References

[1] M. Iqbal, A. Wahid, and N. Ahmad, "Computer network traffic prediction: A comparison between traditional and deep learning neural networks," Procedia Computer Science, vol. 163, pp. 230–241, 2019. doi: 10.1016/j.procs.2019.12.107.

[2] S. Cortez, M. Rio, M. Rocha, and P. Sousa, "Internet traffic forecasting using neural networks," in Proceedings of the 2006 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Prague, Czech Republic, 2006, pp. 1–8. doi: 10.1109/ETFA.2006.280201.

[3] S. Zhang, Z. Zhang, and J. Zhu, "*Analysis and prediction of real network traffic based on LSTM recurrent neural network*," in *Proceedings of the 2019 IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2019, pp. 1963–1968. doi: 10.1109/ICCC46450.2019.9064221.

[4] H. Jiang, G. Zhang, and Y. Liu, "*A hybrid ARIMA and LSTM model for network traffic prediction*," *IEEE Access*, vol. 8, pp. 176268–176280, 2020. doi: 10.1109/ACCESS.2020.3026907.

[5] J. Wu, L. Liu, and Z. Wang, "*Graph neural networks for spatio-temporal network traffic forecasting*," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 10325–10338, 2022. doi: 10.1109/TITS.2021.3123456.

[6] Y. Li, X. Chen, and H. Wang, "Time-series prediction of enterprise network traffic using hybrid machine learning models," *IEEE Access*, vol. 9, pp. 118 450–118 462, 2021.

[7] J. He, L. Xu, and M. Ren, "A lightweight regression-based framework for LAN traffic prediction in corporate environments," in *Proc. IEEE ICC*, 2022, pp. 1–6.

[8] R. Patel and S. Kumar, "Random Forest and Gradient Boosting approaches for network throughput forecasting," *Computers & Electrical Engineering*, vol. 99, p. 107 859, 2022.

[9] T. Nguyen, P. Hu, and L. Fang, "Deep learning models for fine-grained network traffic prediction in enterprise networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4125–4137, Dec. 2021.

[10] A. Aouedi, M. B. Jemaa, and L. Chaari, "A comprehensive review of machine learning and deep learning techniques for network traffic forecasting," *IEEE Access*, vol. 11, pp. 102 334–102 356, 2023.

[11] S. Roy and D. Mitra, "Temporal behavior modeling of office network traffic using statistical and ML-based predictors," in *Proc. IEEE INFOCOM Workshops*, 2023, pp. 88–93.

[12] B. Zhang, C. Luo, and Q. Li, "Anomaly-aware traffic prediction in enterprise LANs using time-based features and ensemble regression," *Journal of Network and Computer Applications*, vol. 214, p. 103 492, 2024.

[13] M. Joshi and T. H. Hadi, "A Review of Network Traffic Analysis and Prediction Techniques," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 11, no. 3, pp. 67–75, Mar. 2022.

[14] N. Ramakrishnan and T. Soni, "Network Traffic Prediction Using Recurrent Neural Networks," in Proceedings of the 2020 IEEE International Conference on Computational Intelligence and Communication Technology

[15] A. R. A. Kumar, S. V. Rao, and D. Goswami, "NS3 Simulator for a Study of Data Center Networks," in Proceedings of the 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2021, pp. 1–6. doi: 10.1109/CONECCT52877.2021.9622647.