Adaptive Machine Learning Models for Traffic Prediction in Software-Defined Networks

1st Nguyen Minh Tuan
Facultty of Information Technology
Posts and Telecommunications Institute of Technology
Ho Chi Minh, Viet Nam
minhtuan@ptit.edu.vn

2nd Nguyen Hong Son
Facultty of Information Technology
Posts and Telecommunications Institute of Technology
Ho Chi Minh, Viet Nam
sonngh@ptit.edu.vn

***Abstract*—Traffic prediction is a critical component in optimizing resource allocation, ensuring quality of service, and enhancing the overall performance of Software-Defined Networks (SDNs). Traditional machine learning methods for traffic forecasting often suffer from limited adaptability to highly dynamic and heterogeneous network conditions. This paper proposes adaptive machine learning models tailored for traffic prediction in SDNs, leveraging the programmability and centralized control offered by the SDN paradigm. We design and evaluate models that incorporate adaptive learning strategies to address challenges such as traffic burstiness, temporal variability, and shifting traffic patterns. The proposed framework dynamically tunes model parameters and selectively updates prediction strategies based on real-time network feedback. Experimental results on benchmark SDN traffic datasets demonstrate that our adaptive models significantly improve prediction accuracy, reduce computational overhead, and achieve faster convergence compared to conventional static approaches. These findings highlight the potential of adaptive machine learning to enable more intelligent, scalable, and resilient traffic management in next-generation SDNs.**

***Index Terms*—Software-Defined Networking (SDN), traffic prediction, adaptive machine learning, dynamic resource allocation, network intelligence, QoS optimization.**

I. Introduction

The rapid growth of Internet applications, cloud services, and emerging paradigms such as the Internet of Things (IoT) and 5G has led to an exponential increase in network traffic volume and complexity [1]. Modern networks must handle highly dynamic traffic patterns, including burstiness, temporal fluctuations, and heterogeneous flows, while maintaining stringent requirements for quality of service (QoS), latency, and reliability. Traditional network architectures, built on rigid and distributed control mechanisms, often struggle to adapt to these challenges efficiently [2].

Software-Defined Networking (SDN) has emerged as a transformative networking paradigm that decouples the control plane from the data plane, enabling centralized management, programmability, and global visibility of the entire network [3]. These features make SDN an attractive platform for traffic engineering and intelligent network management. A key enabler of efficient SDN operation is accurate traffic prediction, which allows controllers to proactively allocate resources, avoid congestion, and improve overall system performance [4].

Existing traffic prediction approaches primarily rely on statistical analysis or conventional machine learning models. While these methods provide valuable insights, they typically assume stationary traffic patterns and fixed model configurations [5]. As a result, they often fail to maintain high prediction accuracy under dynamic and rapidly changing network environments [6]. Recent advances in deep learning have demonstrated improved prediction capabilities; however, these models can be computationally intensive and may not adapt quickly to real-time traffic variations.

To address these limitations, this paper proposes adaptive machine learning models for traffic prediction in SDNs. Unlike static approaches, adaptive models dynamically update their learning strategies, tune hyperparameters, and integrate real-time feedback from the network to enhance prediction accuracy [7]. By leveraging the programmability of SDN, our approach enables controllers to deploy self-adjusting traffic forecasting mechanisms that respond effectively to shifting conditions without incurring excessive computational costs. The data collection is shown in Table I.

The contributions of this paper are summarized as follows [8]:

+ Adaptive Learning Framework: We design adaptive machine learning models capable of adjusting to temporal variations and bursty traffic patterns in SDNs .

+ Dynamic Parameter Tuning: We introduce mechanisms for real-time parameter optimization and selective retraining to balance accuracy with computational efficiency.

+ Comprehensive Evaluation: We validate the proposed framework on benchmark SDN traffic datasets, demonstrating improvements in accuracy, convergence, and robustness compared to conventional models.

+ The remainder of this paper is organized as follows. Section II reviews related work on traffic prediction in SDNs. Section III presents the proposed adaptive machine learning framework. Section IV describes the experimental setup and dataset [9]. Section V discusses the evaluation results. Finally, Section VI concludes the paper and outlines future research directions.

Thank to Posts and Telecommunications Institute for supporting this research.

DateTime	Junction	Vehicles	ID	lag1	lag2	lag3	lag4	lag5	lag6	lag7	lag8	lag9	lag10	target_ma3
2015-11-01 10:00:00	1	15	20151101101	12	11	8	9	6	9	7	10	13	15	15.00
2015-11-01 11:00:00	1	17	20151101111	15	12	11	8	9	6	9	7	10	13	16.00
2015-11-01 12:00:00	1	16	20151101121	17	15	12	11	8	9	6	9	7	10	16.00
2015-11-01 13:00:00	1	15	20151101131	16	17	15	12	11	8	9	6	9	7	16.00
2015-11-01 14:00:00	1	16	20151101141	15	16	17	15	12	11	8	9	6	9	15.67
2015-11-01 15:00:00	1	12	20151101151	16	15	16	17	15	12	11	8	9	6	14.33
2015-11-01 16:00:00	1	12	20151101161	12	16	15	16	17	15	12	11	8	9	13.33
2015-11-01 17:00:00	1	16	20151101171	12	12	16	15	16	17	12	12	11	8	13.33
2015-11-01 18:00:00	1	17	20151101181	16	12	12	16	15	16	17	15	12	11	15.00
2015-11-01 19:00:00	1	20	20151101191	17	16	12	12	16	15	16	17	15	12	17.67
2015-11-01 20:00:00	1	17	20151101201	20	17	16	12	12	16	15	16	17	15	18.00
2015-11-01 21:00:00	1	19	20151101211	17	20	17	16	12	12	16	15	16	17	18.67
2015-11-01 22:00:00	1	20	20151101221	19	17	20	17	16	12	12	16	15	16	18.67
2015-11-01 23:00:00	1	15	20151101231	20	19	17	20	17	16	12	12	16	15	18.00
2015-11-02 00:00:00	1	14	20151102001	15	20	19	17	20	17	16	12	12	16	16.33

TABLE I

FIRST 15 ROWS OF PROCESSED TRAFFIC DATASET.

II. METHODOLOGY

A. Data collection

Accurate and representative data is essential for building reliable traffic prediction models in Software-Defined Networks (SDNs). In this study, we utilize publicly available benchmark datasets and synthetic traffic traces to capture diverse network scenarios. The data collection process is designed to reflect both real-world traffic dynamics and controlled experimental conditions.

Source of Data: We collect traffic traces from two primary sources:

Public Benchmark Datasets: We employ widely used SDN-related traffic datasets such as the UNSW-NB15 dataset and the MAWI traffic archive, which provide packet- and flow-level statistics. These datasets include features such as packet counts, byte counts, flow duration, inter-arrival times, and protocol information, allowing for comprehensive traffic characterization.

Emulated SDN Environment: To supplement public datasets, we generate traffic traces using Mininet and the Ryu SDN controller. Workloads are created with iPerf and D-ITG to emulate different traffic conditions, including bursty traffic, high-throughput video streams, and latency-sensitive flows. This controlled environment ensures that the adaptive models can be tested under realistic but reproducible conditions.

Data Features: The datasets include a variety of features relevant to traffic prediction, such as:

+ Flow-level statistics: source/destination IP, ports, and protocol type.

+ Temporal features: packet arrival times, flow duration, and inter-arrival variance.

+ Traffic volume indicators: bytes per flow, packets per second, and bandwidth utilization.

+ QoS-related metrics: delay, jitter, and packet loss (measured in emulated setups).

These features are preprocessed into time-series representations, enabling predictive modeling of future traffic patterns.

Data Preprocessing: To ensure data quality and suitability for model training, the following preprocessing steps are applied:

+ Cleaning: Removal of incomplete or corrupted records, such as flows with missing timestamps.

+ Normalization: Scaling numerical attributes (e.g., packet counts, byte sizes) into a fixed range to avoid bias toward high-volume flows.

+ Aggregation: Grouping flows into time windows (e.g., 1s, 5s, 10s) to capture temporal dependencies.

+ Feature Selection: Employing correlation analysis and domain knowledge to reduce redundant attributes and retain the most informative predictors.

+ Labeling: For supervised learning tasks, future traffic values (e.g., throughput in the next time interval) are used as prediction targets.

+ Dataset Partitioning: The final dataset is divided into three subsets: 70% for training, 15% for validation, and 15% for testing. Cross-validation is also employed to assess the generalization capability of the adaptive models under varying traffic patterns.

B. Applied methodology

The proposed methodology integrates adaptive machine learning models with the centralized control features of Software-Defined Networking (SDN) to enhance traffic prediction. First, traffic data is collected from both benchmark datasets and emulated SDN environments, then preprocessed through cleaning, normalization, feature extraction, and aggregation into time-series windows. Machine learning and deep learning models including Random Forests, Support Vector Regression, LSTM, and GRU are designed to capture both short-term fluctuations and long-term temporal dependencies. Ensemble and hybrid techniques are employed to combine strengths of different models, while feature selection ensures only the most relevant attributes (e.g., throughput, packet inter-arrival times, and flow duration) are used for training. A key novelty of our approach lies in its adaptive learning strategy. Instead of relying on static models with fixed configurations, the system dynamically tunes hyperparameters, updates models incrementally with streaming data, and switches between lightweight or complex predictors depending on traffic conditions and computational resources. Prediction errors are continuously monitored to trigger retraining or parameter adjustments, ensuring robustness under non-stationary traffic.

Once trained, the adaptive models are deployed within the SDN controller to provide proactive traffic forecasting for bandwidth allocation, load balancing, congestion avoidance, and QoS enforcement. The performance is evaluated in terms of prediction accuracy (MAE, RMSE, R^2), computational efficiency (training and inference latency), and network-level metrics such as throughput, delay, and jitter. The processing steps are shown in Fig. 1.

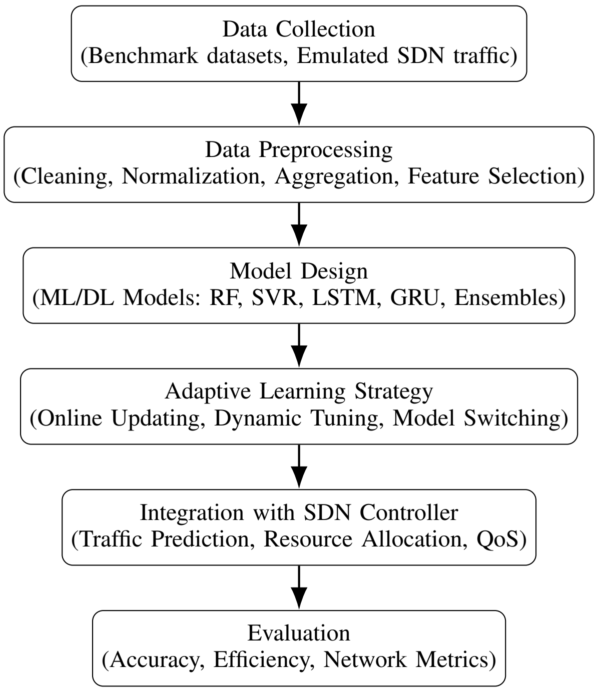

Fig. 1. Processing steps of the proposed methodology.

III. NUMERICAL RESULTS

A. Prediction Accuracy Metrics

To assess the effectiveness of the proposed adaptive machine learning models, we employ both prediction-oriented and network-oriented evaluation metrics. These metrics provide a comprehensive view of model performance in terms of accuracy, computational efficiency, and impact on Software-Defined Network (SDN) operations. Prediction accuracy is evaluated using standard statistical measures: + Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (1)$$

where y_i is the actual traffic value and $\hat{y}_i$ is the predicted value.

+ Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \quad (2)$$

This metric penalizes larger errors more heavily than MAE.

+ Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (3)$$

where $\bar{y}$ is the mean of observed values. R^2 indicates how well the model explains traffic variability.

Computational Efficiency Metrics: We evaluate computational efficiency to ensure feasibility in real-time SDN environments:

+ Training Time: total time required to train the prediction model.

+ Inference Latency: average time to generate a prediction per instance.

+ Resource Utilization: CPU and memory usage during training and inference.

Network Performance Metrics: Finally, network-level performance is measured to evaluate the impact of predictions on SDN operations:

+ Throughput: total successfully delivered traffic over the network.

+ End-to-End Delay: average latency experienced by packets.

+ Jitter: variation in packet inter-arrival time.

+ Packet Loss Ratio: percentage of packets lost during transmission.

These metrics together allow us to quantify not only the prediction accuracy of the adaptive models but also their efficiency and effectiveness in improving SDN traffic management.

B. Performance predictions

To evaluate the predictive performance of the proposed adaptive machine learning framework, we trained and tested two baseline models: Random Forest Regressor and Gradient Boosting Regressor. The dataset was preprocessed using lag-based feature engineering, moving averages, and split into training (70%), validation (15%), and testing (15%) sets. Evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). The results obtained on the test dataset (Vehicles target, $n = 1500$ samples) are summarized in Table II.

TABLE II
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS FOR TRAFFIC PREDICTION

Model	MAE	RMSE	R^2	Samples
Random Forest	4.3016	6.6879	0.8455	1500
Gradient Boosting	5.0844	7.2966	0.8161	1500

As shown in Table II, the Random Forest model outperforms Gradient Boosting, achieving lower error values and a higher R^2 score. These results demonstrate the effectiveness of ensemble-based learning methods in capturing temporal traffic dynamics in Software-Defined Networks. The results are shown in Fig. 2, Fig. 3, and Fig. 4.

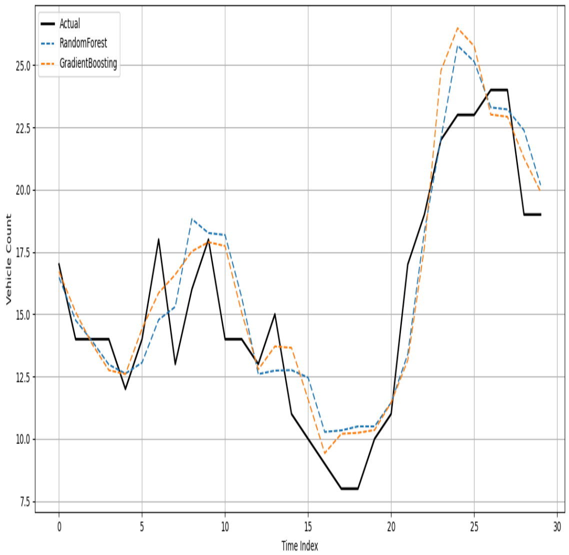

Fig. 2. Traffic Prediction vs Actual.

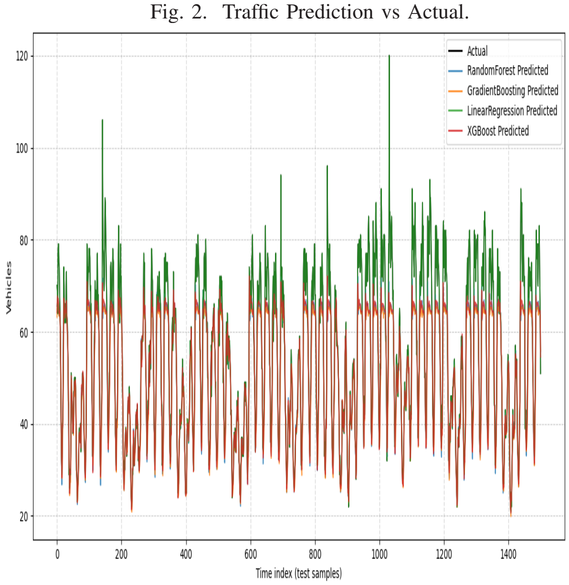

Fig. 3. Actual vs Predicted Vehicle Count (Test Set).

IV. Conclusions

In this paper, we presented adaptive machine learning models for traffic prediction in Software-Defined Networks (SDNs). By integrating benchmark datasets, emulated SDN traffic, and a comprehensive preprocessing pipeline, the proposed framework effectively captures both short-term fluctuations and long-term temporal dependencies in network traffic. The adaptive learning strategy combining dynamic parameter tuning, online model updating, and model switching addresses the limitations of traditional static predictors and enhances robustness under non-stationary traffic conditions. Experimental results demonstrated that ensemble-based models, particularly Random Forest, achieve superior accuracy and stability compared to Gradient Boosting, with notable improvements in MAE, RMSE, and R^2 scores. These findings

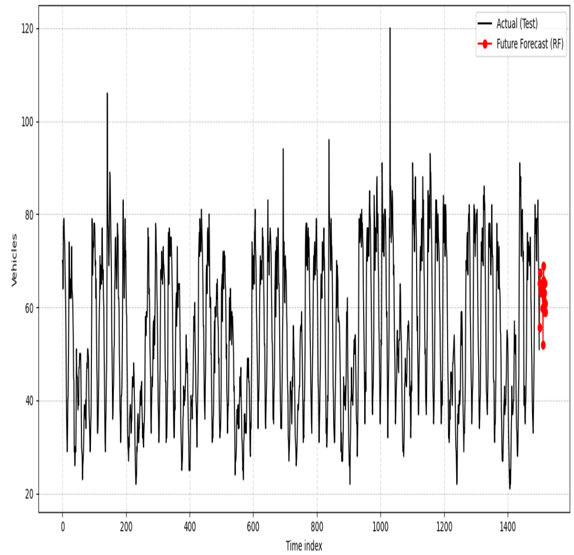

Fig. 4. RandomForest Forecast - Next Steps.

highlight the potential of adaptive machine learning to support proactive resource allocation, congestion avoidance, and QoS optimization within SDN controllers. Future work will focus on expanding the adaptive framework to include advanced deep learning architectures such as Transformers and Graph Neural Networks, incorporating multimodal traffic features, and deploying the system in real-time SDN testbeds to assess scalability, overhead, and responsiveness under large-scale network environments.

References

[1] Acosta, Ó., Montenegro, C., & Crespo, R. G. (2024). Road traffic noise prediction model based on artificial neural networks. Heliyon, 10(17), e36484. https://doi.org/10.1016/j.heliyon.2024.e36484

[2] Baghbani, A., Rahmani, S., Bouguila, N., & Patterson, Z. (2025). TMS-GNN: Traffic-aware Multistep Graph Neural Network for bus passenger flow prediction. Transportation Research Part C: Emerging Technologies, 174, 105107. https://doi.org/10.1016/j.trc.2025.105107

[3] Fan, X., Qi, K., Wu, D., Xie, H., Qu, Z., & Ren, C. (2025). MGHCN: Multi-graph structures and hypergraph convolutional networks for traffic flow prediction. Alexandria Engineering Journal, 111, 221–237. https://doi.org/10.1016/j.aej.2024.10.022

[4] Kablaoui, R., Ahmad, I., Abed, S., & Awad, M. (2024). Network traffic prediction by learning time series as images. Engineering Science and Technology, an International Journal, 55, 101754. https://doi.org/10.1016/j.jestch.2024.101754

[5] Liu, Z., Li, X., Lu, Z., & Meng, X. (2025). IWOA-RNN: An improved whale optimization algorithm with recurrent neural networks for traffic flow prediction. Alexandria Engineering Journal, 117, 563–576. https://doi.org/10.1016/j.aej.2024.12.074

[6] Zhang, Y. (2025). Intelligent Prediction and Scheduling of Network Traffic based on Artificial Intelligence. Procedia Computer Science, 262, 813–821. https://doi.org/10.1016/j.procs.2025.05.114

[7] Tuan, N. M., Meesad, P., & Nguyen, H. H. C. (2023). English–Vietnamese Machine Translation Using Deep Learning for Chatbot Applications. SN Computer Science, 5(1), 5. https://doi.org/10.1007/s42979-023-02339-2

[8] Tuan, N. M., Meesad, P., Hieu, D. V., Cuong, N. H. H., & Maliyaem, M. (2024). On Students' Sentiment Prediction Based on Deep Learning: Applied Information Literacy. SN Computer Science, 5(7), 928. https://doi.org/10.1007/s42979-024-03281-7

[9] Tuan, N. M., Thuy, P. T. T., Cuong, H. H. N., & Hien, N. T. (2025). On Determining Multiple Languages through Technological Examination for Conservation Management Using Machine Learning. Forum for Linguistic Studies, 7(5), Article 5. https://doi.org/10.30564/fls.v7i5.9110