INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

A Consensus-based Traffic Prediction for Time Series Data in SDN

Shahul Hamead Haja Moinudeen[1]*

1 Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India

OPEN ACCESS

Received: 24-06-2024

Accepted: 02-08-2024

Published: 23-08-2024

Citation: Moinudeen SHH (2024) A Consensus-based Traffic Prediction for Time Series Data in SDN . Indian Journal of Science and Technology 17(34): 3488-3495. https://doi.org/10.17485/IJST/v17i34.2061

* **Corresponding author**.

shahulhameadh@ssn.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: This research proposes a methodology for improving real-time traffic prediction by constructing dataset from the live traffic in Software-Defined Networks (SDNs). **Methods:** This work applies time-series based machine learning models such as XGBoost Regressor, LSTM, and Seasonal ARIMA, on the collected and preprocessed traffic data from SDN switches. **Findings:** Simulation shows that the percentage of error parameters that depict accuracy and goodness of fit for traffic predictions are still weak and needs improvement, as these predictions have the potential to compose new policies for the dynamic environment. **Novelty:** The consensus method is applied to forecast future network traffic across multiple time slots as consensus strengthens the decisions.

Keywords: Software defined networking; Traffic prediction; Consensus; Machine learning; Time series data

1 Introduction

Software-Defined Networks (SDN) have revolutionized network administration by enabling dynamic control and programmability. However, to fully realize their potential, intelligent decision-making processes are essential. Machine Learning facilitates adaptive network rules by analyzing data and making predictions. Leveraging a container-based strategy, this work not only aims to bridge the gap between data-driven analytics and policy composition in SDNs but also to offer a Software-as-a-Service (SaaS) platform. SDN is a network architecture that separates the control plane (which makes decisions about where traffic should be sent) from the data plane (which sends the traffic). SDN introduces a centralized controller that communicates with the network devices, enabling dynamic and programmable control over network resources. This separation enhances network flexibility, scalability, and efficiency, allowing for more efficient management and optimization of data traffic [Figure 1]. The Northbound interface in SDN serves as the communication link between the applications or business logic layer and the SDN controller. The Southbound interface in SDN facilitates communication between the SDN controller and the data plane that constitutes network devices, such as switches and routers. In general, the policies of the network are composed at the control plane and the data plane obeys the decision of controller. A set of conditional predicates that lead to an action are called policies. These policies are the result of careful consideration of the data collected from related parameters by monitoring the network at regular intervals. The collected data can be further used for predicting the behavior of network traffic. Therefore, these predicted values of the parameters can be used to frame a policy that will ensure proper functioning of the network. The data collection and predictions are seen as crucial steps for any policy making in SDN.

1.1 Motivation

In the rapidly evolving landscape of SDN, the challenge of dynamically composing policies that align with real-time demands has emerged as a critical bottleneck. The absence of a robust solution to intelligently predict future network traffic and translate these insights into adaptive policies hinders the full potential of SDN environments. Obtaining timely, accurate, and comprehensive information about network performance has emerged as a critical challenge. This is particularly true for Network Service Providers (NSPs) as well as a broader spectrum of individuals and entities engaged in network operations. As data volumes surge and networks grow in complexity, the need for monitoring of key parameters like traffic demand and transmission delay becomes paramount. Conventional monitoring approaches often fall short, struggling to provide a holistic view of the network in real-time. This lag in information availability can lead to suboptimal decision-making, potentially compromising the Quality of Service (QoS) commitments made to customers. Additionally, the diverse array of parameters relevant to network operations further complicates the monitoring process, necessitating a sophisticated solution capable of efficiently aggregating and presenting this multidimensional data. In most studies, data has been collected manually and has been used to determine policies. Raw data cannot be made into a policy directly. The degree of accuracy of prediction achieved by the model and its goodness of fit directly demonstrates the strength of the model. This field of research has opened the scope towards improving the accuracy and fitness parameters of the model. The following paragraph shortly discusses the works in literature which use machine learning based models for traffic prediction.

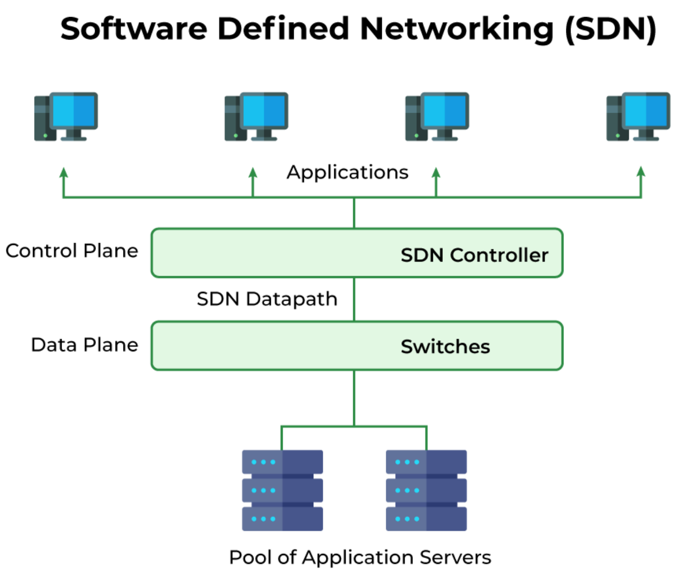

Fig 1. Software-Defined Network Architecture

1.2 Related works

Real time traffic forecasting by combining model-based prediction and data-driven prediction techniques was proposed in [1]. The advanced Machine Learning techniques like ARIMA, KNN and Random Forest were applied for traffic prediction in [2]. An idea for policy composition using predicate logic specific to the ONOS controller was proposed in [3]. Canellas et al. defined a policy model and a policy lifecycle. Within this, each policy has a policy type and a policy priority. Each policy also has a set of policy conditions and policy actions. Traffic prediction for time-series data has been discussed in [4] towards optimizing resource constraints. Deep learning techniques were introduced in [5] to improve QoS parameters. A survey paper by Junfeng Xie et al. [6] serves as a comprehensive guide to the existing literature exploring the intersection of machine learning and SDN. The application of SDN traffic prediction in educational scenario is discussed in [7]. Graph based convolution neural networks was applied for traffic prediction in [8]. The work by Wang et al. [9] explores the usage of LSTM to use non-linear time series data to predict network traffic linked with autocorrelation characteristics. They were able use LSTM in combination with DNN to forecast traffic with MAPE (mean absolute percentage error) of less than 5% across all datasets. They concluded that LSTMs have good performances irrespective of the granularity. The drawback with the approach is that they do not consider the discovery of linear factors in network traffic. This leads us to a quest for combining the different models for the discovery and consideration of both linear and non-linear factors in network traffic. With the same considerations in mind, Yang et al. in 2021[(10)] proposed an ARIMA-SA-BP neural network approach. This approach yielded a MAPE (Mean Average Percentage Error) 9.9614% which was less than the individual ARIMA, BPNN and LSTM models further leading us to think about combining different approaches to traffic prediction. In the realm of Software-defined Networking (SDN), the need for effective traffic classification mechanisms is paramount to enable dynamic and autonomous network control. While various machine learning classification algorithms exist, recent studies reveal the limitations of individual approaches in achieving optimal classification outcomes. Consequently, the adoption of ensemble algorithms, combining multiple classifiers, has emerged as a promising strategy to enhance prediction accuracy in SDN environments. The paper by Yang et al. in 2021[(11)] employs ensemble methods to classify ingress traffic flows into Elephant Flows and Mice flows. Elephant flows represent large, long-lived data transfers characterized by high bandwidth consumption, posing challenges for network resource allocation and congestion management in SDNs. Conversely, mice flows are characterized by short, sporadic bursts of data traffic, often requiring low-latency delivery and efficient handling to prevent network bottlenecks and ensure timely delivery of critical data. Nyaramneni et al. in 2023[(12)] used consensus methods to predict SDN traffic. Their work proves that we can apply these methods for the regression task of predicting traffic. Latif et al. in the year 2022[(13)] proposed an ML as a service module in their methodology to implement Anomaly Detection in SDNs. NSFaaS 2023[(14)] identified cloud orchestration using containers in an as-a-service model as they provide flexibility, scalability and efficiency. They provide Network Slice Federation as a service. Giannopoulos et al. identified the cruciality of as-a-Service models by providing a framework for Monitoring-as-a-Service for NSaaS[(15)]. Monitoring is an important tool that needs to be offloaded from the client to help them stay lightweight and this is ensured through the as-a-Service model[(15)]. The analysis of time-series data to draw temporal inferences in SDN is still in it's infancy and not much work has been done. However, Gao et al. in 2022[(16)] proposed a method to tackle the challenge of predicting temporal patterns of traffic volumes using time-series based ML models. Their work mostly focuses on the variation of traffic volume over time and does not involve other traffic data. There is only a single parameter of traffic volume. Gao et al.[(16)] forecasts only for one timeslot whereas we aim to forecast for multiple timeslots ahead. Shaygan et al. discusses systematic review of artificial techniques used in traffic predictions[(17)].

1.3 Research Gap

Although all these works in the literature were good enough in accomplishing traffic predictions for performing networking functions, the lack of accuracy is still disputed. The error parameters like Mean Squared Error (MSE) and Mean Absolute Error (MAE) depict the quality of accuracy, and R-Squared error (R2) conveys the quality of fitness of the prediction algorithms. The MSE and MAE percentages are expected to be less whereas the R2 is expected to be high for demonstrating good accuracy and fitness respectively. In most of the works, LSTM, XG-Boost and SARIMA models are widely used for traffic predictions, but they indicate vaguely higher MAE and MSE, and smaller R2 values which is a research gap. Reducing the MSE, MAE towards improving accuracy and increasing R2 parameters towards improving fitness is ongoing research in the field of machine learning. We intended to improve the accuracy of our traffic prediction and to confirm it by measuring improvement in these error parameters.

1.4 Contributions

The contributions of our work are as follows:

1. We have defined a methodology to construct a time-series dataset by collecting live traffic data from an SDN environment for various timeslots and pre-processing it.
2. Using traffic information across timeslots, we have proposed a consensus-based prediction of the future 'n' timeslots which takes value majority of XG-Boost, SARIMA and LSTM based models. This prediction aids in assessing the state of traffic in an SDN and new policies can be created out of this. A short description of consensus model is described in the following paragraph.

Consensus: Classic Consensus algorithms solve the problem of omitting uncertainty by counting the common result among various results, as consensus takes value majority, it strengthens accuracy and fitness which positively impact the error parameters. Hence, we have applied consensus to the forecast of individual prediction algorithms in our work to fill the potential research gap.

2 Methodology

In the system architecture, real-time demand and traffic data are acquired through northbound APIs of the SDN controller, followed by a process where a machine learning model makes predictions using diverse parameters. The predictions provide a complete representation of the data landscape. Temporal properties are extracted from the predicted data which becomes a powerful tool for informed policy determination, allowing stakeholders to assess the impact of strategies across various dimensions. The proposed system architecture comprises several interconnected modules designed to effectively predict network traffic, analyze temporal patterns, and to help compose dynamic policies in SDN. This is illustrated in Figure 2.

Firstly, a data collection module operates within the Mininet network environment and the ONOS controller to gather real-time network traffic data from various network nodes at predefined time slots. This data includes essential metrics such as packets sent, packets received, packets lost, bytes sent, and bytes received for each network node. Secondly, the raw data undergoes preprocessing to handle missing values, normalize the data, and convert it into a suitable format for model training. We construct the dataset by following these two steps.

Subsequently, the dataset is split into 70% for training and 30% for validating the performance of our consensus model respectively. The model training module implements Long Short-Term Memory (LSTM), XGBoost, and Seasonal Autoregressive Integrated Moving Average (SARIMA) models to predict network traffic metrics. These models are trained using the preprocessed data to capture temporal patterns and relationships within the network traffic data. Hyperparameter optimization techniques are employed to fine-tune the models and improve prediction accuracy. To obtain robust and accurate predictions, the consensus prediction module combines the outputs of the LSTM, XGBoost, and SARIMA models using an averaging approach. This consensus prediction provides a more reliable forecast for each network traffic metric. After this, we predict values across consecutive time slots to identify temporal patterns and conditions within the network. Based on these observed patterns, network policies can be made to optimize performance and resource utilization. Leveraging the SDN controller's programmability and centralized control, dynamic policies can be determined based on the forecasted temporal patterns and network conditions.

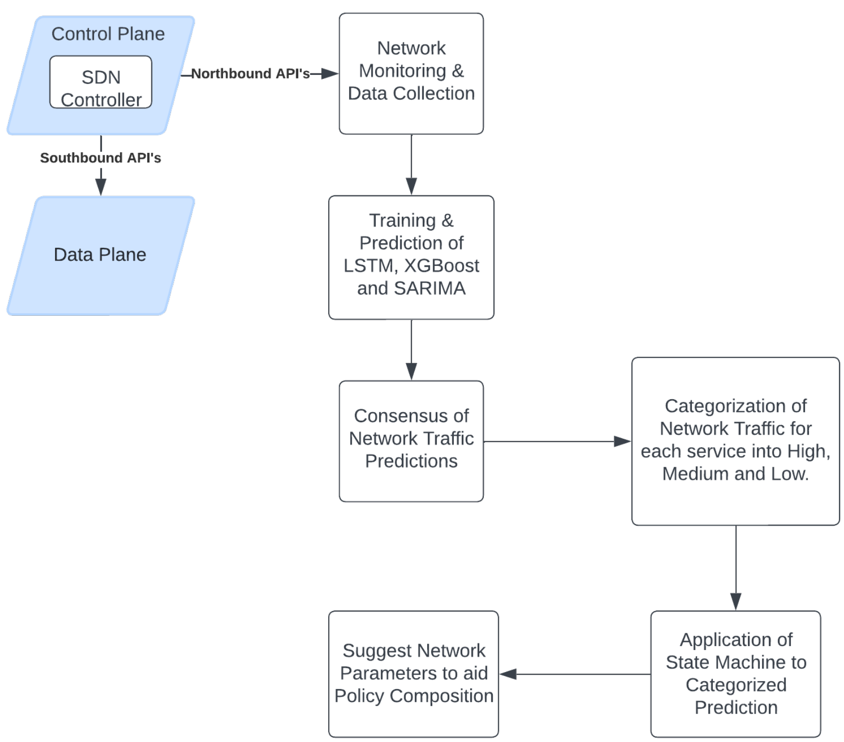

Fig 2. System Architecture from a Policy Makers View

Continuous monitoring and feedback mechanisms are implemented to assess the effectiveness of applied policies and refine them over time. The system's openness is a key feature, offering APIs at different stages. This facilitates accessibility to raw data and predictions. These APIs empower external applications or systems to seamlessly interact with and utilize the data, fostering integration with other tools or platforms. The user interface of the system is a dynamic dashboard, designed for intuitive visualization. Stakeholders, from decision-makers to analysts, can interpret complex data effortlessly. The dashboard visually represents predictions and relevant metrics, enabling quick and informed decision-making. Each component of the dashboard can be given as a service. If the policy maker wants to get access to aggregated data, it can be provided independently as a service. If the policy maker requires aggregated data alone, they can subscribe separately. If they require the complete prediction as well, we can offer it as a separate service. The proposed system, through its interconnected components, not only facilitates real-time decision-making but also lays the foundation for adaptive policies. It transforms raw data into actionable insights, enhancing the efficiency and efficacy of operational and strategic decisions.

2.1 Implementation of proposed system

A. Network Monitoring & Data Collection

A sample topology of 16 hosts connected to an ONOS SDN controller has been created using Mininet. Using a NodeJS server, the data related to the packets sent, packets received, packets dropped while transmitting, packets dropped while receiving, bytes sent, and bytes received for different services are collected. We have used ping and iperf utilities across the hosts through controller to generate small volume and large volume of traffic respectively. The data is collected for 4800 timeslots, where each timeslot is of 5 seconds. The network was monitored continuously using the ONOS SDN controller, and the data was aggregated by harnessing the use of the northbound and southbound APIs, and the data collected was then stored as csv files.

B. Training

The collected data is then passed to various models suitable for predicting time series data like LSTM, XGBoost and Sarima. After training, the models are stored as joblib files, which can be used as a starting point for the next training round. For performance evaluation we have used Mean Squared Error (MSE), Mean Absolute Error (MAE) and Coefficient of Determination (R2) for each of the targets and models. The LSTM model has two layers, stacked sequentially. Both the layers have 50 units and use the ReLU activation function. The model is compiled using the Adam optimizer and Mean Squared Error (MSE) loss function. XGBoost's ability to model nonlinear interactions can lead to more accurate predictions. The input data is scaled using a Min-Max Scaler. The model is then initialized using Multi-Output Regressor, with XGB Regressor as the base estimator. The XGB Regressor is configured with objective as reg:squared_error and n estimators as 100.SARIMA's autoregressive (AR) and moving average (MA) components are well-suited for capturing seasonal dependencies, allowing for accurate predictions based on historical data. An ARIMA (1,0,1) model was used with a seasonal order as 24, which implies one AR term, no differencing, and one MA term, with a period of 24 time slots. A glimpse of a sample dataset for three services (h1, h2 and h3) after handling the missing values and applying normalization process can be seen in Figure 3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	date	h1Packets	h2Packets	h3Packets	h1Packets	h2Packets	h3Packets	h1BytesSe	h2BytesSe	h3BytesSe	h1BytesRe	h2BytesRe	h3BytesRe	h1Packets	h2Packets	h3Packets	h1Packets	h2Packets	h3PacketsT
2	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
3	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
4	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
5	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
6	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
7	2024-02-0	3819	3951	3821	110	236	105	522542	535030	522682	8020	19920	7530	0	0	0	0	0	0
8	2024-02-0	3823	3960	3825	110	241	105	523098	536076	523238	8020	20410	7530	0	0	0	0	0	0
9	2024-02-0	3823	3960	3825	110	241	105	523098	536076	523238	8020	20410	7530	0	0	0	0	0	0
10	2024-02-0	3823	3960	3825	110	241	105	523098	536076	523238	8020	20410	7530	0	0	0	0	0	0
11	2024-02-0	3823	3960	3825	110	241	105	523098	536076	523238	8020	20410	7530	0	0	0	0	0	0
12	2024-02-0	3823	3960	3825	110	241	105	523098	536076	523238	8020	20410	7530	0	0	0	0	0	0
13	2024-02-0	3827	3967	3829	110	244	105	523654	536926	523794	8020	20704	7530	0	0	0	0	0	0
14	2024-02-0	3827	3967	3829	110	244	105	523654	536926	523794	8020	20704	7530	0	0	0	0	0	0
15	2024-02-0	3827	3967	3829	110	244	105	523654	536926	523794	8020	20704	7530	0	0	0	0	0	0
16	2024-02-0	3827	3967	3829	110	244	105	523654	536926	523794	8020	20704	7530	0	0	0	0	0	0
17	2024-02-0	3827	3967	3829	110	244	105	523654	536926	523794	8020	20704	7530	0	0	0	0	0	0
18	2024-02-0	3830	3971	3831	111	246	105	524030	537400	524072	8118	20900	7530	0	0	0	0	0	0
19	2024-02-0	3830	3971	3831	111	246	105	524030	537400	524072	8118	20900	7530	0	0	0	0	0	0
20	2024-02-0	3830	3971	3831	111	246	105	524030	537400	524072	8118	20900	7530	0	0	0	0	0	0
21	2024-02-0	3830	3971	3831	111	246	105	524030	537400	524072	8118	20900	7530	0	0	0	0	0	0
22	2024-02-0	3830	3971	3831	111	246	105	524030	537400	524072	8118	20900	7530	0	0	0	0	0	0
23	2024-02-0	3841	3975	3835	118	246	105	525160	537956	524628	8692	20900	7530	0	0	0	0	0	0
24	2024-02-0	3841	3975	3835	118	246	105	525160	537956	524628	8692	20900	7530	0	0	0	0	0	0
25	2024-02-0	3841	3975	3835	118	246	105	525160	537956	524628	8692	20900	7530	0	0	0	0	0	0

Fig 3. Sample Timeseries data collected by monitoring the network

C. Prediction & Consensus

The benchmark models i.e., SARIMA, XGBoost, and LSTM, were utilized to predict the traffic data for the subsequent 100 timeslots. One model may not be able to make predictions with low error always, so consensus was taken based on the prediction of these 3 models. The implementation results are elucidated in the next section.

3 Results and discussions

It could be seen that MSE, MAE and R-squared error percentages are widely measured in the literature to express the effectiveness of the accuracy[1,2,4,6–8,10,16,17]. Hence, we track the MSE, MAE and R-squared values for the existing algorithms and the proposed technique. We have generated traffic for low and high rates. The duration of the timeslot is assigned as 5 seconds. We have aggregated the packets for 300 seconds as one sequence. The traffic rate ranges between 1000 packets/sequence to 1000/sequence which is equal to 3 to 35 per second. The network utilities iperf and ping are used to generate packets in the Mininet emulator. It could be seen that consensus technique is able to improve accuracy by minimizing MAE error percentage (Figure 4) and MSE error percentage (Figure 5). The increase in the R2 error percentage in comparison with existing algorithms indicates better fitness of the consensus (Figure 6). In all the cases, XGBoost algorithm could be observed as the closest competitor. It could be seen that the error percentages of LSTM and SARIMA models of our results are in line with the results for those models obtained by existing works. The percentage of R2 is measured as 84 and MAE is measured as 76 for LSTM model in[10] whereas MAE values for LSTM and ARIMA are measured as 76 and 77 in[1]. R2 is measured as 65 in[1]. Our results confirm the similar behavior for the typical machine learning models. It could be witnessed that; the application of consensus model outperforms traditional models and achieves an MAE as low as 10-15 and improves R2 as high as 90.

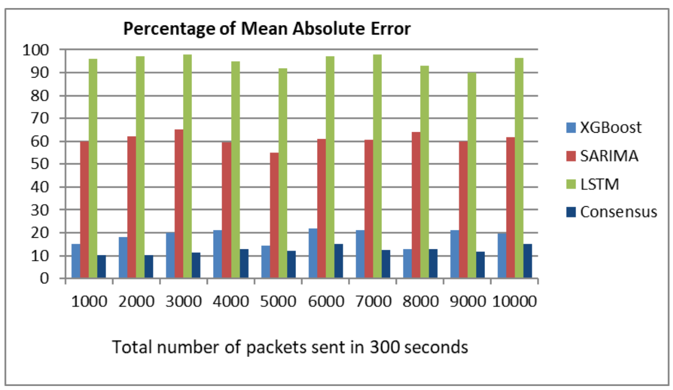

Fig 4. Percentage of Mean Absolute Error compared among existing works and proposed work

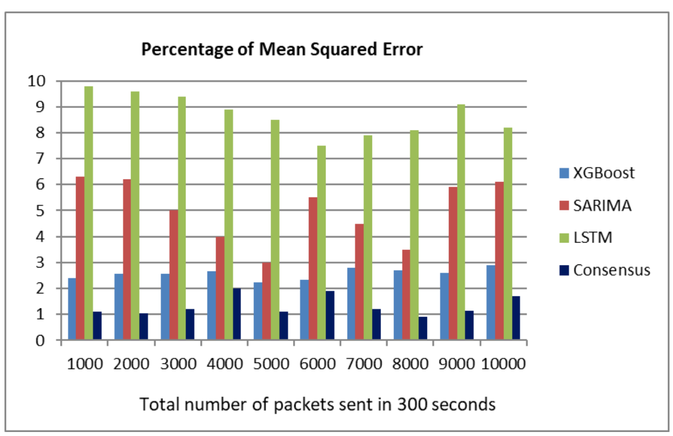

Fig 5. Percentage of Mean Squared Error compared among existing works and proposed work

By leveraging the collective intelligence of multiple predictive models, SDN administrators can identify opportunities for performance optimization and resource allocation based on the consensus prediction. This can lead to more efficient utilization of network resources and improved overall network performance.

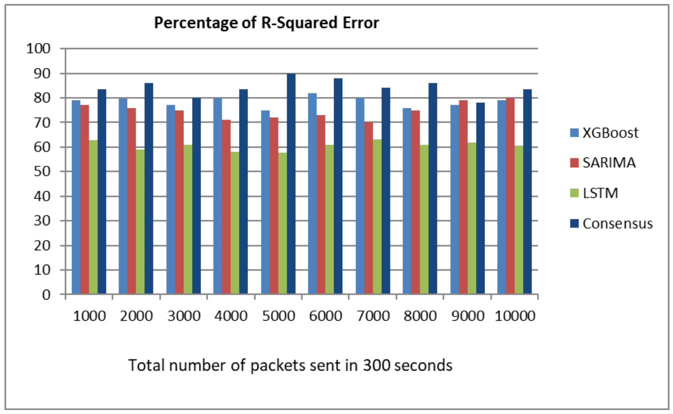

Fig 6. Percentage of R-Squared Error compared among existing works and proposed work

4 Conclusion

This study has proposed a method that mandates performs data collection, and traffic prediction based on consensus in the SDN environment. Improving the accuracy of the traffic prediction and fitness of the model has been taken as the objective. The proposed consensus-based novel technique has been compared against the benchmark models. The results demonstrate accuracy via the reduction in the MAE and MSE error measurements. The goodness of fit could be witnessed through the increase in the R2 value. Our model achieves twice the accuracy and 15% extra fitness than XGBoost which is the closest competitor. The proposed system serves as a comprehensive framework for proactive network management within SDN environments. By harnessing advanced analytics and machine learning techniques, administrators can make data-driven decisions, optimize resource allocation, and enhance overall network performance. Our approach leverages predictive modelling and analytics to drive efficiency and resilience in SDN infrastructures, ultimately leading to enhanced network performance and user experience. In the future, the change of accuracy can be measured with respect to change in the configuration of various statistical parameters involved in traffic predictions.

References

1) Kim Y, Tak HY, Kim S, Yeo H. A hybrid approach of traffic simulation and machine learning techniques for enhancing real-time traffic prediction. *Transportation Research Part C: Emerging Technologies*. 2024;160. Available from: https://doi.org/10.1016/j.trc.2024.104490.
2) Kanakaprabha S, Ganeshkumar G, Sureshkumar A, Udhayakumar T, Nisha M, Poornaprakash P. Advanced Machine Learning Approaches for Improving Traffic Flow Predictions in Smart Transportation Systems. In: Accelerating Discoveries in Data Science and Artificial Intelligence I;vol. 421 of Springer Proceedings in Mathematics & Statistics. Springer. 2024;p. 145–154. Available from: https://doi.org/10.1007/978-3-031-51167-7_14.
3) Canellas F, Mimidis A, Bonjorn N, Soler J. Policy Framework Prototype for ONOS. In: 2019 IEEE Conference on Network Softwarization (NetSoft). IEEE. 2019;p. 218–222. Available from: https://doi.org/10.1109/NETSOFT.2019.8806691.
4) Wan L, Cheng W, Yang J. Optimizing Metro Passenger Flow Prediction: Integrating Machine Learning and Time-Series Analysis with Multimodal Data Fusion. *IET Circuits, Devices & Systems*. 2024;2024:1–13. Available from: https://doi.org/10.1049/2024/5259452.
5) Wassie G, Ding J, Wondie Y. Traffic prediction in SDN for explainable QoS using deep learning approach. *Scientific Reports*. 2023;13(1):1–15. Available from: https://doi.org/10.1038/s41598-023-46471-8.
6) Xie J, Yu FR, Huang T, Xie R, Liu J, Wang C, et al. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Communications Surveys Tutorials*. 2019;21(1):393–430. Available from: https://doi.org/10.1109/COMST.2018.2866942.
7) Zheng S, Yang S. Application of SDN Network Traffic Prediction Based on Speech Recognition in Educational Information Optimization Platform. *Computational Intelligence and Neuroscience*. 2022;2022:1–11. Available from: https://doi.org/10.1155/2022/5716698.
8) Liao K, Zhou W, Wu W. A self-attention dynamic graph convolution network model for traffic flow prediction. *International Journal of Machine Learning and Cybernetics*. 2024. Available from: https://doi.org/10.1007/s13042-024-02210-7.
9) Wang S, Zhuo Q, Yan H, Li Q, Qi Y. A network traffic prediction method based on LSTM. *ZTE Communications*. 2019;17(2):19–25. Available from: https://doi.org/10.12142/ZTECOM.201902004.
10) Yang H, Li X, Qiang W, Zhao Y, Zhang W, Tang C. A network traffic forecasting method based on SA optimized ARIMA–BP neural network. *Computer Networks*. 2021;193. Available from: https://doi.org/10.1016/j.comnet.2021.108102.

11) Yang T, Vural S, Qian P, Rahulan Y, Wang N, Tafazolli R. Achieving robust performance for traffic classification using ensemble learning in SDN networks. In: ICC 2021 - IEEE International Conference on Communications. IEEE. 2021;p. 1–6. Available from: https://doi.org/10.1109/ICC42927.2021.9500571.
12) Nyaramneni S, Potluri T, Somavarapu J. Advanced ensemble machine learning models to predict SDN traffic. *Procedia Computer Science*. 2023;230:417–426. Available from: https://doi.org/10.1016/j.procs.2023.12.097.
13) Latif Z, Umer Q, Lee C, Sharif K, Li F, Biswas S. A machine learning-based anomaly prediction service for software- defined networks. *Sensors*. 2022;22(21):1–17. Available from: https://doi.org/10.3390/s22218434.
14) Dalgitsis M, Cadenelli N, Serrano MA, Bartzoudis N, Alonso L, Antonopoulos A. Nsfaas: Network slice federation as a ser- vice in cloud-native 5g and beyond mobile networks. In: 2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE. 2023;p. 59–64. Available from: https://doi.org/10.1109/NFV-SDN59219.2023.10329748.
15) Giannopoulos D, Papaioannou P, Tranoris C, Denazis S. Monitoring as a service over a 5g network slice. In: 2021 Joint Euro- pean Conference on Networks and Communications 6G Summit (EuCNC/6G Summit). IEEE. 2021;p. 329–334. Available from: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482534.
16) Gao C, Wang J, Yan C. A dual-stage attention based SDN traffic prediction method. In: 2022 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking. IEEE. 2022;p. 315–322. Available from: https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00047.
17) Shaygan M, Meese C, Li W, Zhao XG, Nejad M. Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities. *Transportation Research Part C: Emerging Technologies*. 2022;145. Available from: https://doi.org/10.1016/j.trc.2022.103921.