Vol. 1 No. 1

Advances in Engineering and Multidisciplinary Knowledge

Load Balancing in Cloud-integrated SDN using Machine Learning

Stojanovi Mirjana[1], *Heumüller Ralf*[2]

[1] University of Belgrade, Serbia
[2] Otto von Guericke University, Germany

Abstract: Software-defined network (SDN) is relatively a new approach for managing computer networks. In SDN, network administrator manages the network easily through software without deeper involvement in the network at its lower layers. For SDN monitoring and performance analysis, it becomes important to forecast the network traffic. In the proposed work, a systematic method to predict the future congestions in the network has been introduced. It applies a Machine Learning technique to fulfill number of relative performance attributes. These attributes are reliable input to Long Short-Term Memory (LSTM) neural network. The objective of this neural network is to predict congestions and activate load balancer in a timely manner. It is not feasible to extract numerical features with absolute values and use them as income with the neural network for machine learning and deep learning algorithms, especially LSTM neural networks. The best way is to provide pre-processed and easy-to-analyze income data for Artificial Intelligence (AI) techniques. The neural network therefore has clearer information no matter how much the network's conditions change so that the network does not care how much the numerical values change as its income become within a specified range. The simulation results depict the ability of LSTM to adapt the behavior of the network performance activating load balancing through attributed features with an accuracy of up to 92% before reaching the throttling state.

Keywords: Software-defined networks (SDN), Load balancing, Neural network, Machine learning (ML), Long Short-Term Memory (LSTM).

1 Introduction

New technologies in networking are emerging nowadays. Cloud computing and big data are the most important and modern technologies which witness great developments. Therefore, managing traditional networks has become a big challenge. Based on the modern and new developments, it is necessary to change the traditional architecture of the network. In response, the concept of Software Defined Network (SDN) has been proposed. This concept makes network management more compatible, convenient, and manageable. SDN is a programmable network model which refers to the ability of software applications to dynamically program individual network devices and thus control the network behavior as a whole. SDN offers a mechanism to improve various considerations of the network management. The basic concept of the SDN is to make a clear separation between the level of control and the level of redirection. SDNs are able to organize and manage up to thousands of network devices through a management center, network monitoring in terms of resources and connectivity, automatically changing network behavior, and improving the utility of the devices such as network bandwidth improvement, load balancing, congestion control and other considerations associated with programmability and scalability [11].
It is a known fact that network usually suffers from the resources limitations. This affects the Quality of Service (QoS) in the network. Load balancing is one of the most important aspects, followed in the networks, to overcome the problem of resources limitations. It aims at distributing the network traffic in order to increase the network efficiency and reliability. In conventional networks, load balancing is done based on the local information of the network. Therefore, load balancing is not so accurate in such networks. On the other hand, SDN controllers have a holistic view of the network and can produce perfectly distributed loads [2]. Technically, load balancing divides the network load in order to avoid

*Corresponding author: stojanovi.m@bg.edu.rs

the overloading of the network resources. Increasing productivity, reducing response time, and regulating congestion traffic are the main load balancing goals. It is imperative to mention that traditional networks do not have a holistic view of the network. Therefore, load balancing methods in traditional networks are not accurate. Load balancing methods in SDN are more accurate with better performance. Due to industry concerns, in any research related to SDN, the problem of load balancing is one of the most important issues [3].

Recently, researchers are focusing on introducing Artificial Intelligence (AI) and Machine Learning (ML) techniques into the network's software structure in order to improve load balancing. The ideal load balance should exhibit smart behavior in customizing and distributing requests. Intelligent resource allocation includes equal load distribution on all servers at every point in time. The emergence and growth of AI has transformed load balancing research from traditional load scheduling algorithms such as min-min and round-robin into smart load balancing models based on ML and deep learning [4].

Machine learning is an AI application that provides the system's ability to learn and improve from experience without being programmed at this level. It uses past data to train and predict the accurate results. Using ML algorithms, it is possible to predict the occurrence of many of the events in future based on the experience. Thus, for the considered problem in SDN, it will be possible to predict the load and the load balancing algorithm will be automatically activated. It will replace the human intervention to the bare minimum. In this study the Long Short-Term Memory (LSTM) neural network is used as a type of iterative Recurrent Neural Networks (RNN). In RNN, the output of the last step is fed as an input to the current step. LSTM can, by default, retain information for a long period of time. It is used for processing, forecasting and classification based on time series data.

The outline of this paper is as follows. After the introduction in section 1, section 2 delves into the recent related work. Section 3 defines the problem, addressed in this work. The proposed method, that applies ML for the load balancing problem in SDN network, has been presented in section 4. In section 5, the performance study of the proposed model has been carried out which has been done through simulation experiments. Finally, in section 6, the conclusion of the work has been drawn.

2 Related Work

Chen-xiao and Ya-bin (2016) proposed a strategy to maintain network load balancing in SDN. The strategy is adaptive to grid congestion movements and schedules streams based on the route load calculated using the synthetic neural network with rear diffusion. The proposed load balancing strategy aims to improve the track load and choose the least-carrying path for the new data flow. The methodology relied on the use of four features expressing load in each expected path. These features are bandwidth utilization ratio, package loss rate, transmission time, and transmission jumps. Using these four load traits, the neural network model is trained to predict the choice of path with the lowest load as a data transfer path. The results show that the load-balancing strategy proposed in this work can determine a more efficient transmission route for the data flow and achieve a 19.3% reduction in network transition time [5].

Azzouni and Pujolle (2018) introduced NeuTM, a software platform for predicting traffic matrix (TM) network congestion by adopting long-short term memory repetitive neural networks (LSTM RNNs). A congestion matrix (TM) is a mathematical matrix that gives congestion sizes for a certain period of time between all possible couples from start to destinations in a particular IP address field. Network-based methodology (LSTM RNNs) has demonstrated the ability to successfully predict TM matrix in large networks [6].

*Corresponding author: stojanovi.m@bg.edu.rs

Yeo et .al in [1] introduced the switch migration relay method based on Reinforcement Learning (RL). Dynamic relay of allowances provides a balanced load distribution among SDN controllers. Current enhanced learning methods (RL), to migrate dynamic switches such as MARVEL modeling load balance for each controller has linear improvement. Although it is widely used to model network flow, this type of linear improvement is not suitable for real-world workloads of SDN controllers. It is because linkages between resource types change unexpectedly and continuously. Therefore, a controller selection scheme and an enhanced learning-based switch have been proposed to migrate switches between controllers, balancing the user switch-aware reinforcement learning load balancing (SAR-LB). SAR-LB uses the utilization ratio of different resource types in both controllers and switches as input to the neural network. They also consider keys as RL factors to reduce the workspace for learning, while taking into account all migration cases. Results show that SAR-LB achieved a better distribution (near to equal distribution) of the load between SDN controllers due to precise decision-making to migrate the switch as it achieved a better standard deviation between distributed SDN controllers compared to current systems of up to 34% [1].

Babayigit and Ulu in [7] used deep learning technology (DL) to balance the load in SDNs. The deep learning model, similar to the traditional neural network, was trained for variable load between connections where the neural network income represents the cost between the specific pathways. DL's load-balancing response time was compared with artificial neural network (ANN), support vector machine (SVM) and logistic regression (LR). Experimental results revealed that ANN and DL response time results are lower than those of SVM and LR algorithms, and DL resolution is higher than ANN.

Chen et .al in [8] proposed an automated load-balancing software structure in SDN based on enhanced learning (ALBRL). In this, the load balancing optimization model is designed in high-load congestion motion scenarios and adjusts the Deep Deterministic Policy Gradient (DDPG) enhanced deep gradient algorithm to find a near-perfect path among the network users. The proposed ALBRL method intends to update the set of parameters (bandwidth for each link and average bandwidth used) in DDPG. Results show that the proposed ALBRL has faster training speed than traditional augmented learning algorithms and significantly improves the network productivity.

Begam et .al in [9] used a multiple regression-based search algorithm (MRBS) to choose the optimal server and routing track in SDN to improve the performance even under large load conditions and asphyxiations such as high message rate, different message frequency and unexpected congestion patterns. The proposed algorithm reduced the delay and time to more than 45% and shows a better usage of the server by 83% when compared to traditional algorithms.

Through the research reviewed in this section, the widespread interest in improving the efficiency of load balancing in SDN has been observed. It intends to reduce the congestions by integrating load balancing techniques with artificial intelligence and machine learning. Most of the works confine to the network status data to classify the state of congestion and do not intend to predict enough time to perform any action for preventing the congestion. However, this research work focus on the use of AI techniques in predicting the congestions so that rapid intervention can be provided to network specialists to solve the problem in a timely manner. In the present work, focus will be on extracting features that are useful in prediction to be used as income for the LSTM neural network [10-19].

3 The discussed Problem

TSDN networks suffer from congestion problems. This research aims at avoiding the congestion problems in SDN based on load balancing technique. That is, the Service's inability to process requests received, thereby reducing the efficiency and reliability of the network. In the SDN network, to achieve the required load balancing, the console must complete a series of Real time Less-loaded Server selections (RLS). To identify the target server for a new stream, RLS is applied and is also used to calculate the path to the target server while the new stream enters the domain for the first time. Based on the real-time network status, RLS makes the decision to redirect for each new stream. However, the use of RLS as a central controller in SDN poses some problems such as congestions of a single controller, poor responsiveness, reliability and poor scalability.

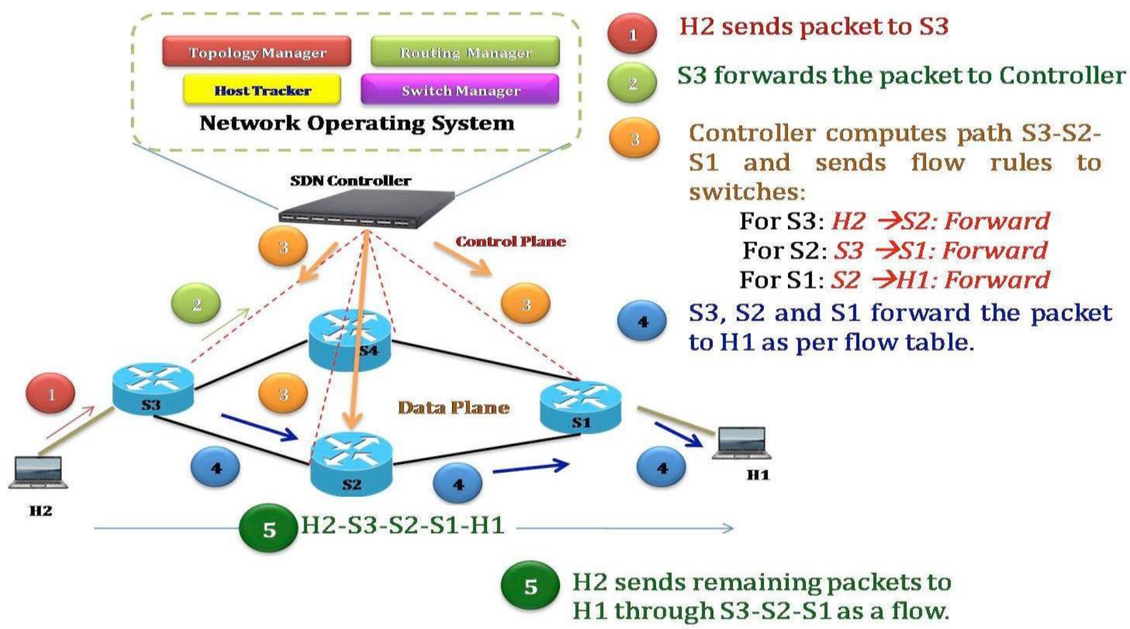

Figure 1: SDN network working mechanism

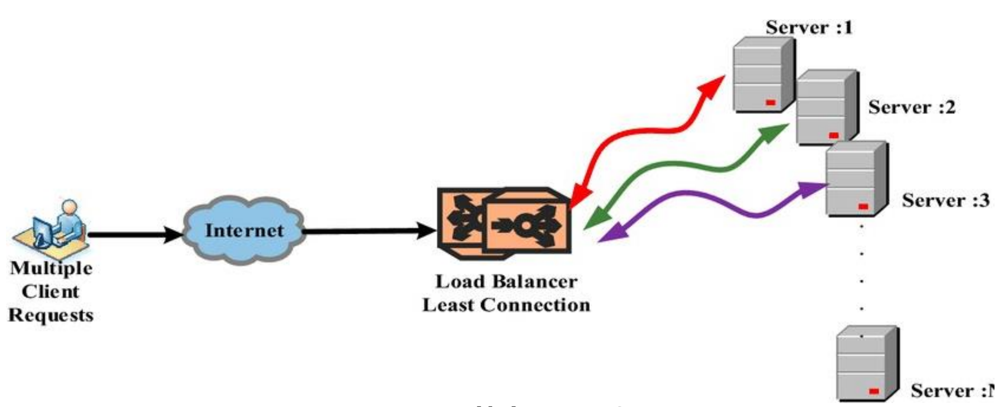

Figure 2: Load balancing in SDN

3.1 *LSTM Neural Networks*

Long Short-Term Memory Networks is a deep learning, sequential neural network that allows information to persist. It is a special type of Recurrent Neural Network which is capable of handling the vanishing gradient problem faced by RNN. LSTM was designed by Hochreiter and Schmidhuber that resolves the problem caused by traditional RNNs and machine learning algorithms. LSTM can be

*Corresponding author: stojanovi.m@bg.edu.rs

implemented in Python using the Keras library. For example, while watching a video, the user can remember the previous scene, or while reading a book, the reader knows what happened in the earlier chapter. RNNs work similarly; they remember the previous information and use it for processing the current input. The shortcoming of RNN is they cannot remember long short-term dependencies due to vanishing gradient. LSTMs are explicitly designed to avoid long-term dependency problems.

4 The Proposed Method

The proposed work utilizes the abilities of LSTM in order to extract appropriate standard features. Based on the extracted features, and different working scenarios, congestions can be predicted in the proposed network (figure 4). Load balancing algorithm, within the SDN network microcontroller, is activated taking the advantage of the advanced prediction using LSTM. The research proposes a particular network structure for conducting the required experiments.

4.1 Load balancing in SDN

Both, congestion in SDN networks and load balancing, result in the problem of congestions in computer networks. The main objective of this research is to explore the attributes and data to be used with AI tools for predicting the congestions in SDN and then to solve them. In this study, the main objective is to study the possibility of providing features and data that can later be used with AI tools in order to predict congestions or work to resolve them at some point. This study has worked to extract features that have the greatest impact on congestions behavior so that single-dimensional signals are usable by different AI technologies. To study, SDN network has been simulated using Mininet simulator. The simulated SDN contains number of users and switches. For ease of the experimental study, the simulated SDN has limited number of elements (users and switches).

As shown in Figure 3, the proposed load balancing process has the following steps.

1. Creating and configuring SDN: A network is created which contains 4 servers, 4 users (hosts) and 10 switches interfaces. The maximum available bandwidth for the connection is 100 Mbps.
2. Load balancing: The Weighted Round Robin method is used as a load balancing method.
3. Extracting results: Work on number of measures to reflect the network status and to show clear changes in the transition to asphyxiation.
4. Finally, these features are used for prediction using neural network Long Short Term Memory (LSTM).

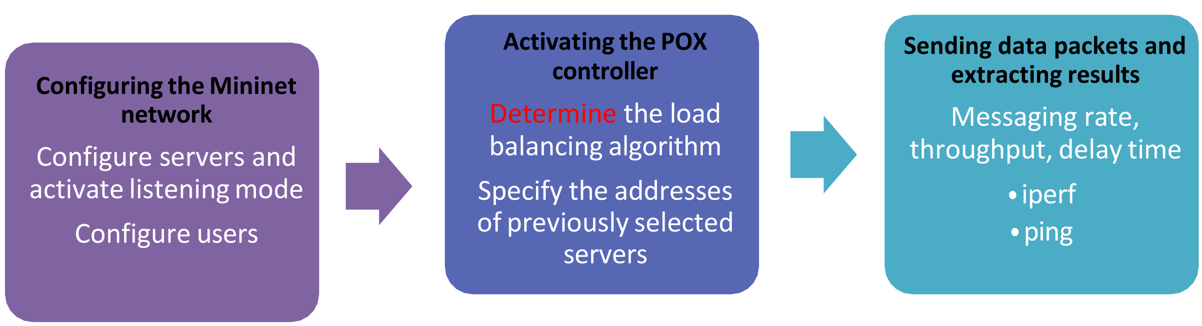

Figure 3: The proposed load balancing process in SDN

The SDN architecture designed using Mininet simulator is shown in Figure 4. Devices from h1 to h4 depict servers while h5 to h8 are users.

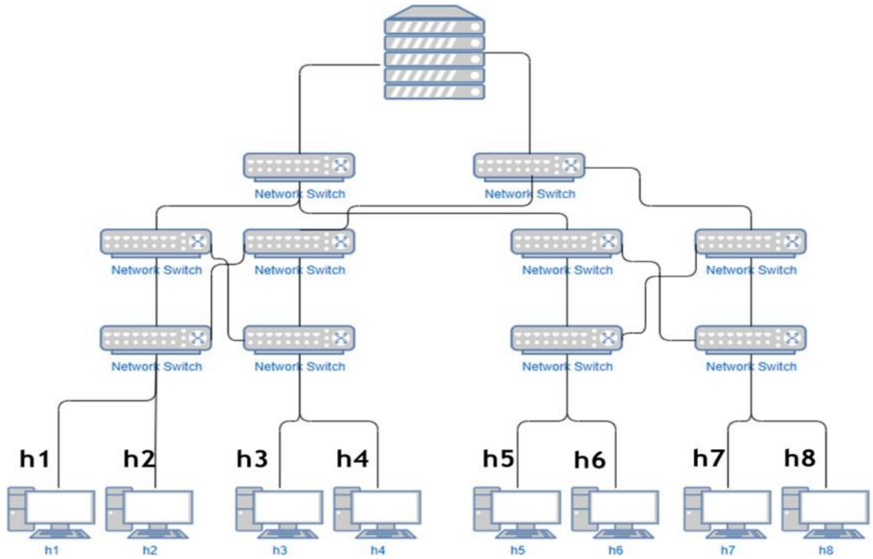

Figure 4: SDN architecture using Mininet simulator

It is important to mention that the chosen structure of this SDN network has been taken so that it is similar to some existing structures in the real world. The chosen structure is suitable for studying the case of congestion and applying the proposed solution. It has been built based on different analysis criteria so that is serves the purpose of this research.

4.2 Extracting SDN Performance Parameters

Three parameters have been considered to measure the performance. These are transmission rate, received data, and bandwidth. Productivity of the current network has been measured and evaluated based on three parameters mentioned above. Critical change in their values indicates congestion. Experimentally, it is better to work with relative values of these parameters in order to avoid both the available physical resources and the normal state of data traffic. The concept of relative values expresses a sequence of values that are all attributed to a fixed value. Therefore, by estimating the change attributed to network performance, a relative value is extracted which indicate the extent to which the output differs over the time compared to the initial value. For example, in case of increasing transmission speed, the value of 1 indicates a relative increase in transmission speed and thus better network performance, and vice versa. In this research, time scale signals have been divided into epochs periods so that each period represents 3 seconds of each signal. The relative signal has been calculated by the value of the first moment of the current period.

Extracting methodology for the attributed metrics is shown in Figure 5. It involves the following steps.

- Create messaging processes between users and predefined servers
- Follow-up the performance of messaging operations through Iperf.
- Automatic recording of values within the csv file for each epoch is 3 seconds.
- Assign the recorded values of each scale to the initial value (t=0).

*Corresponding author: stojanovi.m@bg.edu.rs

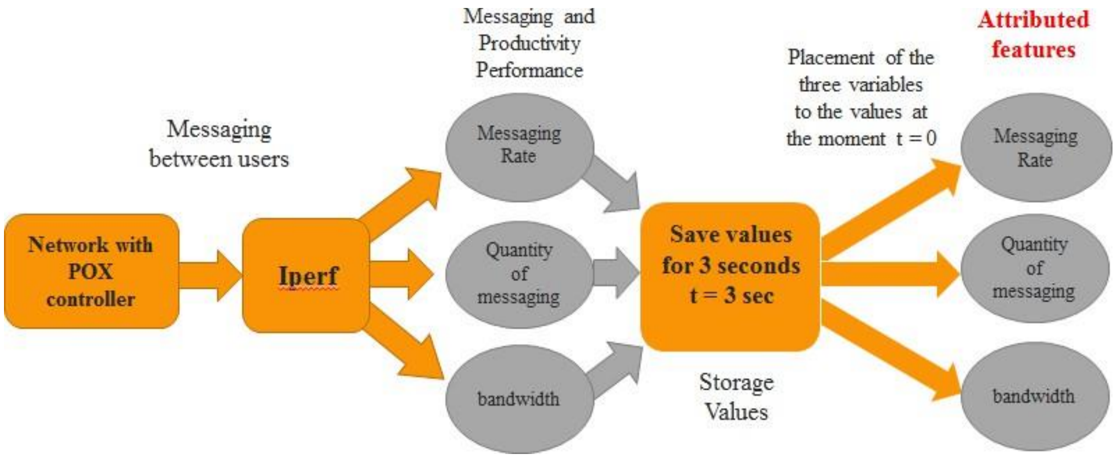

Figure 5: The Proposed Methodology for Feature Extraction in Evaluating Network Performance

4.3 Bottleneck prediction using LSTM

The basic application of LSTM Neural Network is to predict and forecast the temporal signals in advance. In this work, LSTM neural network has been employed with time signals or features attributed to SDN network performance metrics. Thus, it is possible to predict the occurrence of congestion in advance. Using an alert mechanism, network engineers are intervened in time and congestion can be avoided. Network engineers in this case are able to employ, activate or optimize load balancing algorithms so that they can avoid congestion and deterioration of the network performance. In this work, LSTM neural networks have been implemented using the library of deep learning tools in the MATLAB program (MathWorks, Natick, MA, United States). All computations have been performed on a laptop that has an Intel (R) Core I i5- 4300U CPU @ 1.90 GHz and 2494 MHz with two cores and 4 logical processors. Figure 6 shows training LSTM on the measured features and predicting their behavior.

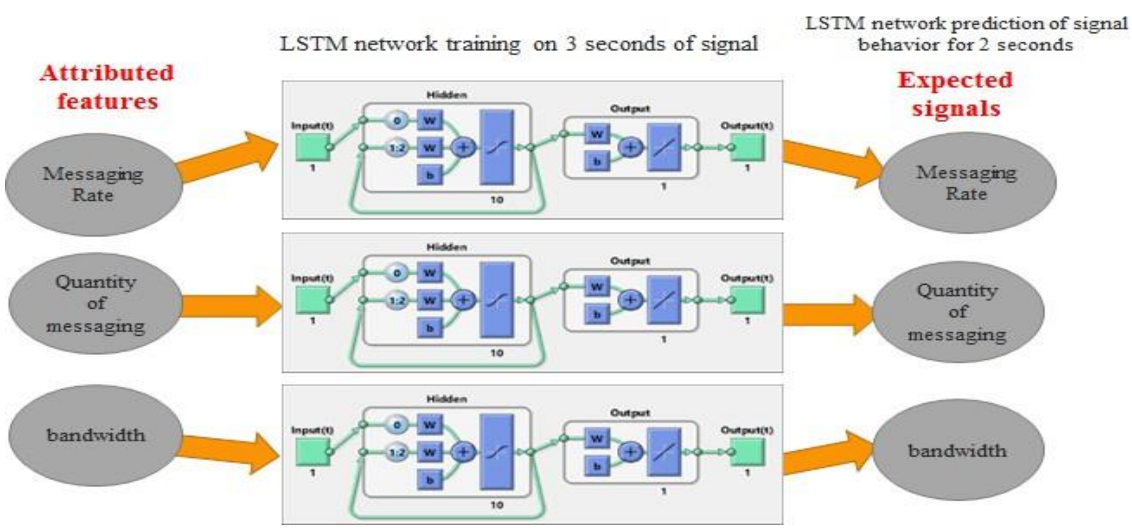

Figure 6: Training LSTM networks on the measured features and predicting their behavior

5 Performance Evaluation

Performance is measured using prediction accuracy (AC) and prediction sensitivity (SE). These two values are calculated using equations (1) and (2) respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

Where:

- True Positive (TP): number of correctly ranked attempts.
- False Positive (FP): Number of attempts wrongly classified as a chokehold which is normal.
- True Negative (TN): number of attempts that are classified as normal and they actually are.
- False Negative (FN): number of attempts classified as normal is a case of congestion.

5.1 Predicting Attributed Performance Features using LSTM

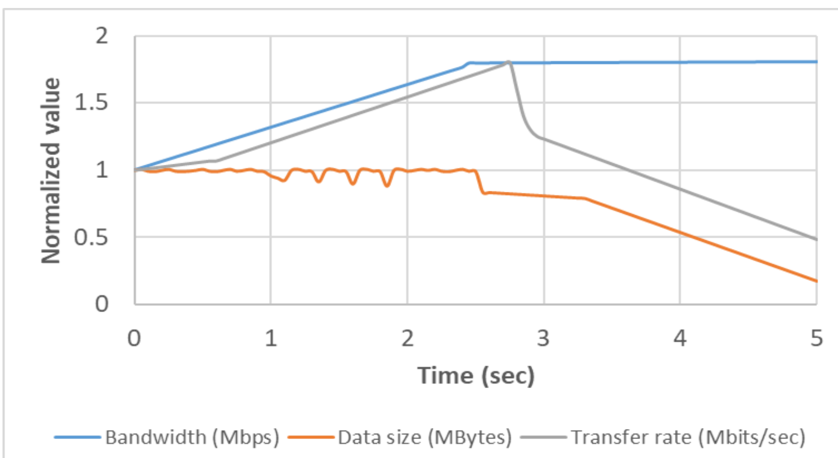

Figure 7: Attributed network performance during messaging operations and congestions

As clear from Figure 7, one-dimensional signals of each feature have been used with LSTM Neural Network to predict future signal behavior and thus the behavior of the network. By changing the signals and training the network on the rate of their change, congestions can be predicted. Congestions lead to a decrease in the amount of messaging, messaging rate, and the use of the entire bandwidth.

*Corresponding author: stojanovi.m@bg.edu.rs

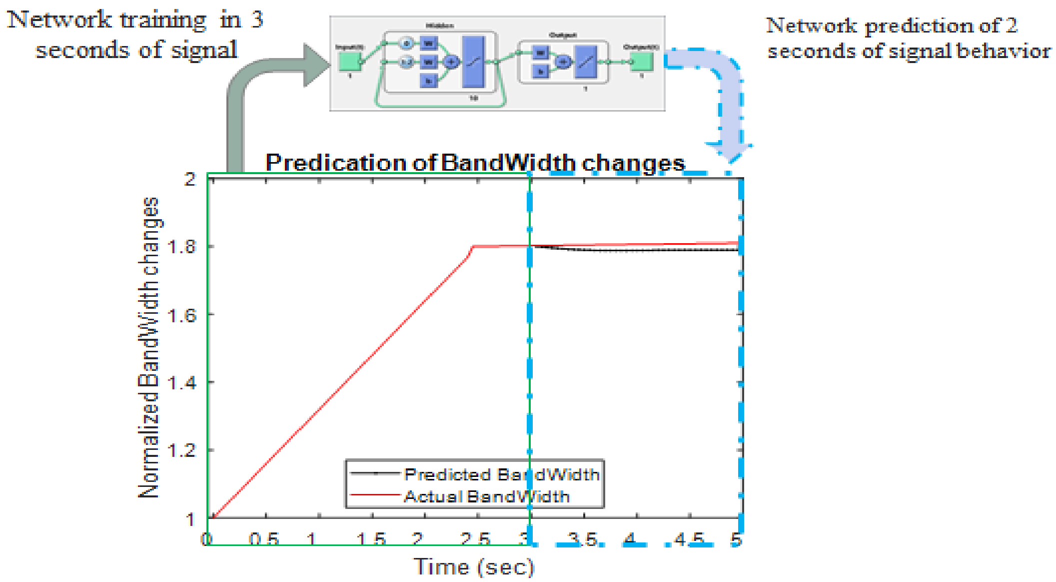

Figure 8: Predicting network performance during messaging operations and congestions using LSTM

Figure 8 shows the expected (black) and actual (red) values of attributed bandwidth behavior. The process revolves around using LSTM time signal or the attributed performance signal for 3 seconds in order to train and modify LSTM network parameters. LSTM is then able to predict the future signal behavior for two seconds as clear from Figure 8.

Training rate of LSTM Neural Network, during one of the experiments is illustrated in Figure 9 during 250 repetitions of training.

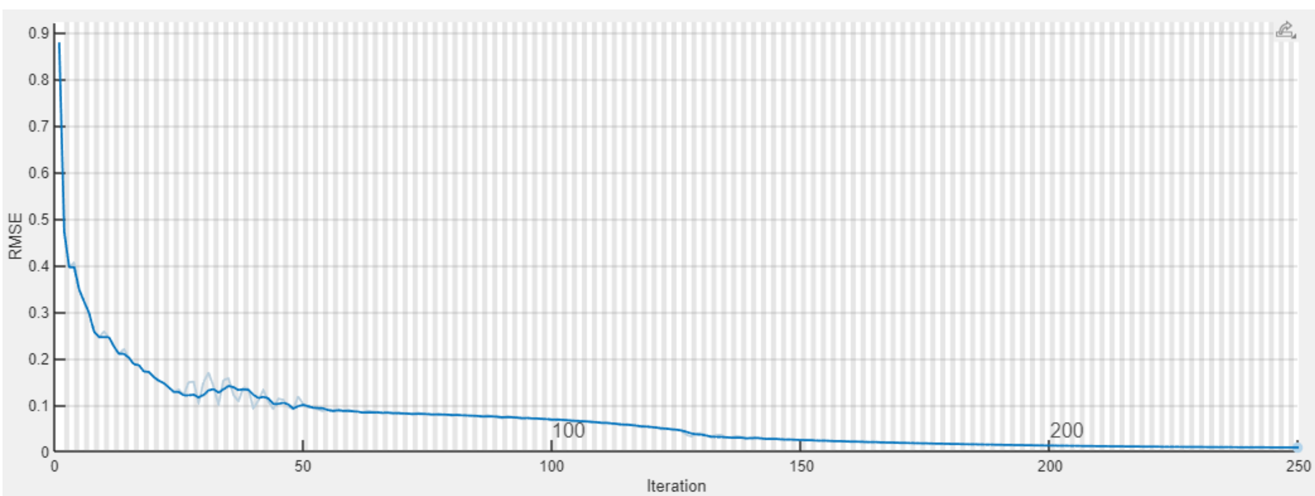

Figure 9: Training rate of the LSTM network through root mean squared error (RMSE)

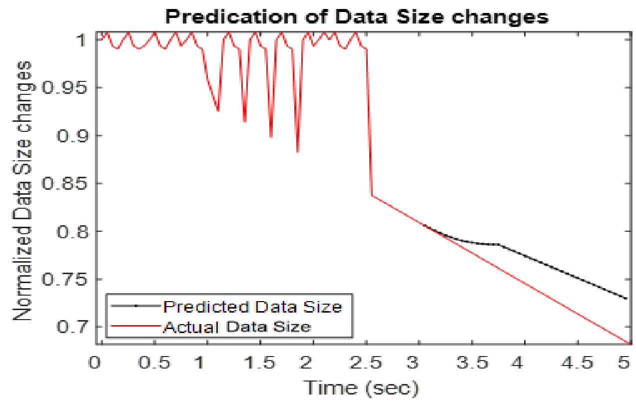

Figure 10 (a): Predicting change in data size behavior in two future seconds

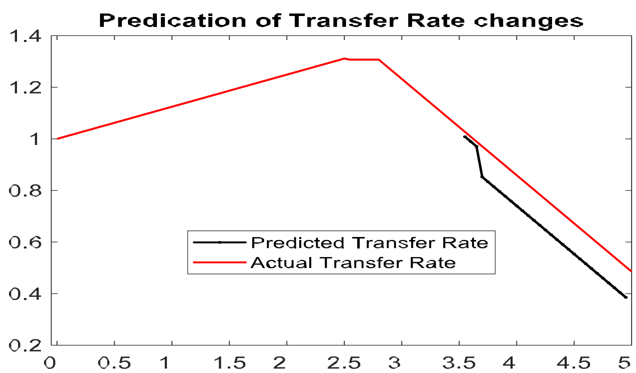

Figure 10 (b): Predicting change in transfer rate behavior in two future seconds

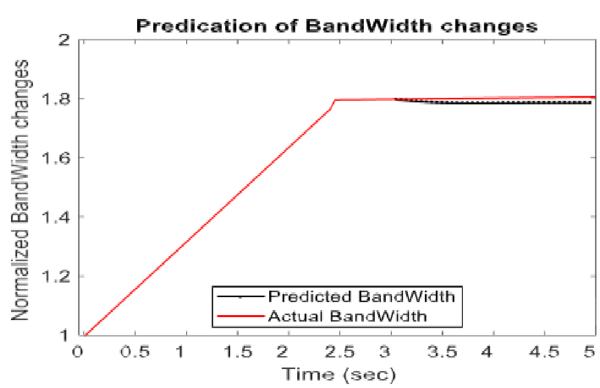

Figure 10 (c): Predicting change in bandwidth behavior in two future seconds

Figures 10(a), (b), and (c) show the actual blueprint of the attribute utilization mechanism (data size, transfer rate, and bandwidth in order) as a 3-second signal plus the storage of two additional seconds in order to verify the future forecast and introduce it to LSTM to predict its behavior. It shows the ability of LSTM to give an approximate behavior of the network performance change from the deviation rate and distance from the initial situation. LSTM is also able to give an approximate numerical prediction

*Corresponding author: stojanovi.m@bg.edu.rs

of the transmission rate, throughput and bandwidth used.

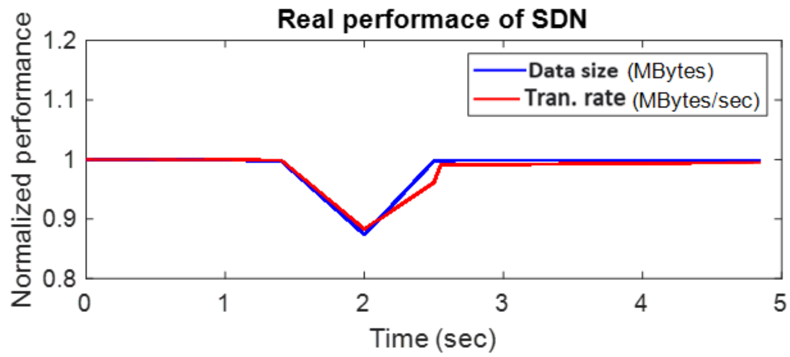

Figure 11 (a): Case of servers' congestion and LSTM prediction

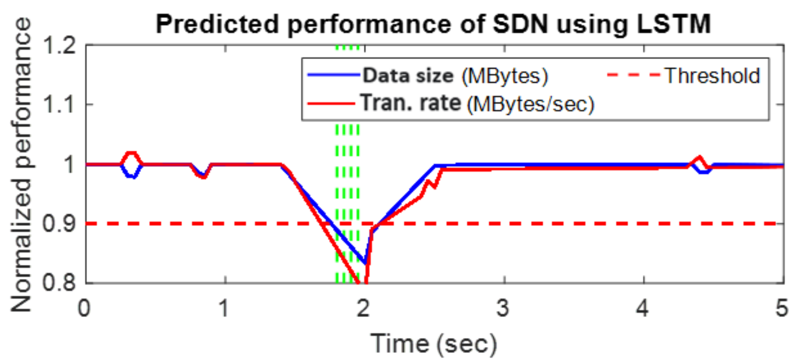

Figure 11 (b): Case of servers' congestion and LSTM prediction on applying load balancing

Figure 11(a) and (b) show the effect of applying the proposed load balancing mechanism. Figure 11(a) shows both the messaging rate and amount of messaging in case of server congestion. When network performance degrades due to load imbalance, LSTM predicts the state and load balancing mechanism is activated in a timely manner. Figure 11(b) shows the case when LSTM network predicts a breakdown in network performance and activates load balancing. The studied SDN returns to its normal state in 2 seconds after applying load balancing mechanism. A threshold of 0.9 has been set for the predicted values from the LSTM network as an activation command within the SDN microcontroller to activate load balancing using weighted round robin (WRR).

Figure 12 shows server's performances during the gradual congestion and after the intervention of the load balancer based on LSTM Neural Network prediction.

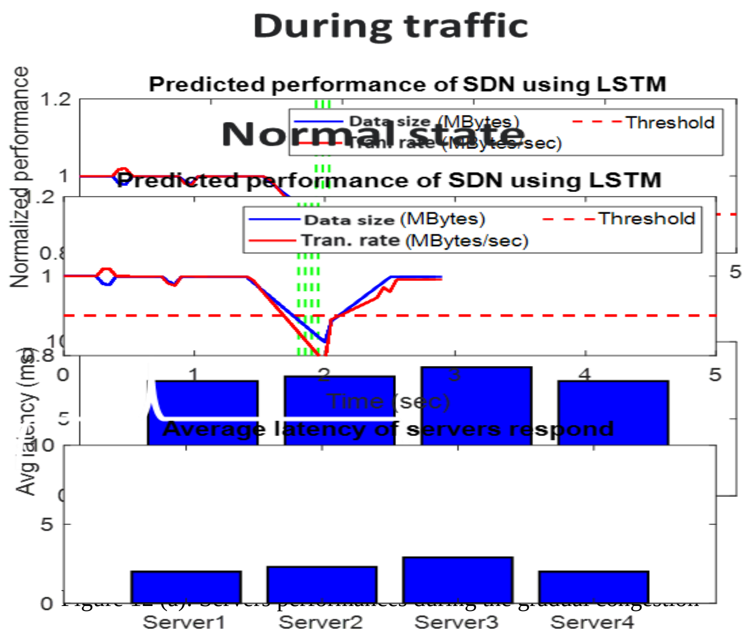

Figure 12 (b): Servers performance during congestion and load balancing activation

During initialization, the server was able to process all requests. During the start of congestion and the arrival of predictions to the pre-set threshold value, the average response time of the service was increased as in the figure 12(a) until the load balancing algorithm has been activated proactively. The load balancing algorithm has been able to redistribute the requests to the service according to each server's ability to process requests and return response to its normal values 12(b).

5.2 A Comparative Analysis

The proposed work has been compared with the state of the art on various performance parameters. The results have been depicted in Table 1.

*Corresponding author: stojanovi.m@bg.edu.rs

Table 1: A Comparative Analysis

Ref	Objectives	ML tool	Input	Results
(Chen-Xiao, 2016)	predict the integrated load for different path and to choose one with least load	BP Artificial Neural Network	bandwidth utilization ratio, packet loss rate, transmission latency and transmission hops.	19.3% network latency decrease at most.
(Azzouni, 2019)	Traffic Matrix prediction	Long Short-Term Memory (LSTM)	traffic vector (TV)	-
(Babayigit, 2021)	load-balancing of SDN-based data center networks (DCNs)	deep learning model	cost between determined paths	DL-based load balancing is better than the other ML-based load balancing strategies.
(Chen, 2022)	find a near-optimal path between network hosts.	Deep Deterministic Policy Gradient (DDPG) algorithm	bandwidth of each link and the average used bandwidth	has a faster training speed than existing reinforcement-learning algorithms and significantly improves network throughput.
The proposed method	Performance metrics prediction	LSTM	Normalized features of transfer rate and delivered data size	88% of correct prediction and LB activation

6 Conclusions and Future Work

In this research, a virtual SDN network has been built with a specific number of users and servers. Messaging rate, bandwidth used, and time delay with fixed bandwidth of 100 Mbps were studied. The designed network has been tested by changing transmission rate, delay time, and bandwidth. It has been noticed that the network reaches a throttling state by increasing time delay and decreasing the transmission rate. Values of the extracted features have been used as attributed values to optimal reference value. In addition, these attributed values have been used as inputs to LSTM Neural Network. The proposed method is to track the values of data transmission rate, bandwidth, and time delay and converting them to relative values. This method provides a new methodology that applies AI techniques in predicting the network congestion. As a result, the performance of load balancing algorithms has been evaluated. The obtained results show the ability of LSTM to predict an approximate behavior of the network performance change. This prediction is based on deviation rate of the values and their distance from the initial state. It gives an approximate numerical prediction of the transmission rate and bandwidth. This research presents a new concept for predicting network congestion situations. This provides an early warning for network experts in order to intervene in time to solve congestion problems. The principle of predicting network behavior using LSTM can be used not only to activate the load balancer instantaneously, but also to improve the parameters of some algorithms proposed in the literature to avoid congestion. From this work, it can be concluded that integrating LSTM into SDN microcontroller architecture provides a faster computational mechanism. A time delay of 3 seconds, in data processing due to the software integration between Python and Matlab, has been noticed. As this work has been done by implementing prediction and load balancing at the server level, in future work can be done at the link level, by predicting the cost of links and choosing the best one.

In future, the work can be further developed by proposing different machine learning approaches based on different AI techniques. Comparisons between the different approaches can also be made.

Conflicts of Interest Statement

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] Begam, G. S., Sangeetha, M., & Shanker, N. R. (2022). Load Balancing in DCN Servers through SDN Machine Learning Algorithm. Arabian Journal for Science and Engineering, 47(2), 1423-1434.

[2] Lin W, Zhang L. The load balancing research of SDN based on ant colony algorithm with job classification. In2016 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA-16) 2016 May (pp. 472-476). Atlantis Press.

[3] Chen-Xiao, C., & Ya-Bin, X. (2016). Research on load balance method in SDN. International Journal of Grid and Distributed Computing, 9(1), 25-36.

[4] Muchori, J. G., & Mwangi, P. M. (2022). Machine Learning Load Balancing Techniques in Cloud Computing: A Review. 11(06), 179-186.

[5] Azzouni, A., & Pujolle, G. (2018, April). NeuTM: A neural network-based framework for traffic matrix prediction in SDN. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium (pp. 1-5). IEEE.

[6] Yeo, S., Naing, Y., Kim, T., & Oh, S. (2021). Achieving Balanced Load Distribution with Reinforcement Learning-Based Switch Migration in Distributed SDN Controllers. Electronics 2021, 10, 162.

[7] Babayigit, B., & Ulu, B. (2021). Deep learning for load balancing of SDN-based data center networks. International Journal of Communication Systems, 34(7), e4760.

[8] Chen, J., Wang, Y., Ou, J., Fan, C., Lu, X., Liao, C., ... & Zhang, H. (2022). ALBRL: Automatic Load-Balancing Architecture Based on Reinforcement Learning in Software-Defined Networking. Wireless Communications and Mobile Computing, 2022.

[9] M. Anbar, M. Nassr, M. Abdallah, E. Vostorgina, M. Kolistratov and H. A. A. Alasadi, "Sidelobe Canceller Performance Evaluation using Sample Matrix Inversion algorithm," 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), 1-6 (2022)

[10] AL-Asadi, H. A. A., Hasan, R., Nassr, M., & Anbar, M. Power Consumption in Wireless Sensor Network: A Machine Learning Approach. Computing, Performance and Communication Systems, V6. N1 (2022)

[11] Ranim Youns, Fadi Ghosna, Mohammad. Anbar, Mohammad. Nassr "Predicting BER value in OFDM-FSO systems using Machine Learning techique", OPTICA PURA Y APLICADA, V55. N4 2022

[12] Ranim Youns, Mohammad. Nassr, Mohammad. Anbar, Fadi Ghosna, Hamid Ali Abed Alasadi, Daria K Voronkova " Machine Learning Approach for predicting suitable wavelength in OFDM-FSO system", 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), V5, 1-5 (2023)

[13] Neghabi AA, Navimipour NJ, Hosseinzadeh M, Rezaee A. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. IEEE Access. 2018 Mar 5;6:14159-78.

[14] Boucadair M, Jacquenet C. Software-defined networking: A perspective from within a service provider environment.

*Corresponding author: stojanovi.m@bg.edu.rs

InRFC 7149 2014 Mar.

[15] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM computer communication review. 2008 Mar 31;38(2):69-74.

[16] Mikami, A. (2016). Long short-term memory recurrent neural network architectures for generating music and japanese lyrics. Comput. Sci. Dep, 1-27.

[17] Polat, H., Türkoğlu, M., Polat, O., & Şengür, A. (2022). A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks. Expert Systems with Applications, 197, 116748.

[18] Chen-Xiao, C., & Ya-Bin, X. (2016). Research on load balance method in SDN. International Journal of Grid and Distributed Computing, 9(1), 25-36.

[19] Azzouni, A., & Pujolle, G. (2018, April). NeuTM: A neural network-based framework for traffic matrix prediction in SDN. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium (pp. 1-5). IEEE.