2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-1490-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCWC51732.2021.9376123

SDPredictNet-A Topology based SDN Neural Routing Framework with Traffic Prediction Analysis

Sowmya Sanagavarapu
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University, 600025
Chennai, India
sowmya.ssanagavarapu@gmail.com

Sashank Sridhar
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University, 600025
Chennai, India
sashank.ssridhar@gmail.com

Abstract— **Software Defined Networking is an intelligent network management approach for monitoring and improving performance such as in cloud computing. These networks follow separation between the Forwarding layer and the Control layer in the system to enable efficient programmability to perform network configurations. The SDN Controller present in the network has complete information about the network schema and its components to update the routing information of the switches present. The dynamic nature of traffic in these networks with multi-layer switches hinder the efficiency of SDN's performance in multi-cloud environments. This paper proposes SDPredictNet, a Recurrent Neural Network framework deployed on the SDN Controller that can predict the traffic in the network and update flow tables of the higher layer switches to perform routing based on the perceived bottlenecks in the network. SDPredictNet uses a Sequence-to-Sequence model that trains on the network data congestion to forecast the traffic in the SDN which is then modelled by an Artificial Neural Network to predict the path of the packets. SDPredictNet has achieved a RMSE score of 0.07 and an accuracy of 99.88% for traffic estimation and subsequent path determination.**

Keywords—Software Defined Networks, Traffic Prediction, Multi-layer switches, Deep Neural Routing, LSTM, Sequence to Sequence Modelling, Encoder-Decoder

I. Introduction

In today's age, multi-functional networks with high level programmability and network function virtualization are used for global operations [1]. Software Defined Networks (SDNs) are used for their centralised architecture [2] due to the decoupling of the Control Layer with the Forwarding Layer. These networks have multifunctional layers that ensure high security and the ability to control and shape the traffic present in the network. SDNs achieve deployment of services at accelerating speeds across the network from a single point of control in Wide Area Networks (WAN) [3] whilst optimising network costs. The routing in the SDNs is performed by the SDN Controller present in the network that has information about the network topology [4], called the SDN Domain and the measure of traffic flowing through the cross-links.

The network traffic in the SDN [5] varies with the number of hosts communicating in the network actively and the network bandwidth of the links connecting the hosts in the network. The traffic parameters collected from the network determines the degree of congestion in the network [6]. High traffic rates present in the network may adversely affect the efficiency functioning of the network due to performance bottlenecks. The prediction of traffic in the network [7] can be used to route the communication flows across the network between the hosts which could increase the robustness of the system to make it secure.

Traffic-based routing in the network proposes to select the most optimal path for the communication between the active hosts in the network [8]. In this way, the least cost path will be predicted for the multilayer switches in the network topology. The deep neural routing model can be trained to predict the activation of switches in the path of the network for automation, delay reduction and better performance. The time taken for updating the flow table of the higher layer switches in the network can be considerably reduced.

Sequence to Sequence (Seq2Seq) modelling [9] tries to map the dependencies between an input and output sequence of data by encoding a variable length to a fixed length vector which can then be decoded to form the output sequence. The encoder-decoder framework [10] in this paper makes use of Long-Short Term Memory (LSTM).

LSTM networks have the ability to take advantage of spatial features [11], which are static in nature, and temporal features, which are dynamic in nature, from the sequential data for modelling and prediction of non-linear time-series problems [12]. Their application in real-time traffic traces for intelligent path computation [13] is used in this paper to build a deep neural routing model, SDPredictNet.

SDPredictNet uses structure, path and traffic parameters to determine the route that a given packet should take. Instead of only taking static parameters into account, SDPredictNet forecasts the traffic parameters at the given instant based on preceding values. The dynamically determined traffic parameters are used along with the static network parameters to predict the path of packets. The dynamic path prediction [14] from SDPredictNet can be fed to the SDN Controller to update the flow table of the switches in the network using REST APIs in the Application Layer of SDN.

The rest of the paper is organised as follows. Section II gives the summary of some of the best works done in SDN and deep neural network routing. The system design of SDPredictNet is given in Section III, followed by the implementation details in Section IV. The analysis of the traffic forecasting and path prediction of SDPredictNet are given in Section V. Section VI presents the conclusion of the paper with Section VII summarising the proposed future works for the paper.

II. Related Works

A. SDN traffic prediction using deep neural network architecture

The measurement and management of traffic in software-based networking was discussed by Shu et al [15] and the subsequent proposal of traffic management to achieve load balancing and energy-saving scheduling in the network. The measure of traffic was proposed to be formed in the application layer and network layer of the SDN to obtain the traffic matrix to reflect the current state of traffic in the network. This matrix could then enable the designing of traffic scheduling algorithms to dynamically plan data forwarding paths.

Traffic based analysis was performed in SDPredictNet using LSTM Seq2Seq modelling after collecting the dataset using iPerf analysis between hosts present in the system. The prediction of traffic on the collected data was performed for constructing a dynamic deep neural routing model to predict the path in the higher layer switches in the network. Neural modelling of traffic data helps to dynamically gauge the network conditions in the present based on the network conditions of the past, which helps the routing model to avoid congestion and data loss by choosing the path with least traffic.

Real-time urban analysis of traffic in SDNs was performed by Bhatia et al [16] to predict the congestion-sensitive spots in the network containing a SDN controller in a cloud architecture. This was done by the use of a clustering algorithm to predict the future traffic densities at each spot by using a short-term prediction model for predicting nonlinear traffic flow in the network. After tuning the hyperparameters of the model, it was evaluated using Absolute Percentage Error and Mean Square Error for each batch of data in the prediction of traffic.

In our paper, the SDN network was constructed using multi-layer switches to collect the structure, network and traffic parameters from the hosts communicating in the network. The traffic parameters were processed in SDPredictNet by using a LSTM Seq2Seq sub-model to analyse the non-linear flow of the traffic parameters across the higher-layer switches in the network and to forecast these values for the activation of switches in the network for deep neural routing. Seq2Seq modelling uses a window of previous parameters to best determine the current parameters, which helps capture the variance of network parameters.

Yang and Sun [17] did research in their paper on the optimisation of traffic in SDN using Maximum Entropy- Hidden Markov Model (ME- HMM). They planned to resolve the imbalance of load in the network by monitoring the data generated in the SDN using OpenFlow protocol. The denoised data was analysed to predict the routing for early deployment of Controller to each subdomain for reducing delays using maximum entropy algorithm. The evaluation for traffic prediction was carried on this data for deployment in the network.

The data collected from the traffic flows in our network were captured using a packet-tracer tool by filtering the OpenFlow protocol-based packers. The data was standardised for lowering the noise before feeding into SDPredictNet for future traffic prediction in the network at the higher-layer switches and subsequent network routing. SDPredictNet was evaluated with Mean Absolute Error and Root Mean Square Error for analysis.

B. Dynamic path prediction in SDNs using deep neural routing

Dynamic routing optimisation was discussed by Shreya et al [18] proposed a dynamic routing model in SDN that uses a bio inspired optimisation mechanism- Ant Colony Optimisation (ACO). The ACO works by random movement across the network until the most optimised route is established between the chosen destination from the host. Analysis of variance (ANOVA) statistical model was used for evaluation to compare the delay and throughput of the system with existing Dijkstra's routing algorithm.

In SDPredictNet, a ANN-based neural routing sub-model is proposed to analyse the non-linear data and to capture the stochastic information in the network for deep neural routing in SDN. The data collected from the SDN included the variation of data flows in the system with higher layer switches. Structure, path and traffic parameters were collected to feed into SDPredictNet for the successful path prediction by activation of switches in the network. The ANN tries to determine the combination of switches that will be activated for the given input network parameters.

Traffic Engineering was handled by using multi-path routing by Abugabah et al [19] for transportation systems using the modified wave algorithm. Routing using Artificial Intelligence with congestion avoidance was modelled by Wu et al [20]. The system aimed to alleviate the impact of monitoring periods for congestion along with learning ability for the system to make superior routing decisions in the network. The routing model was deployed in the Control Plane of the SDN with Ryu Controller for topology discovery and monitoring. The data for training the neural model was collected and set a congestion flag for each flow during data transfer. The trained model then predicted the probability of congestion in the network based on the active data flows in the SDN.

Dynamic routing was performed in SDPredictNet using a ANN sub-model for the dynamic routing of the data packets in the network by predicting the probability of the upper layer switches to be activated during the communication between hosts. The dataset collected from the network had structure and path parameters and forecasted traffic parameters such as bandwidth and congestion in the network to determine the path of the packets during a transfer for choosing the most optimised path in the network.

In summary, traffic parameters were collected from the clos-topology based system using a packet tracer tool hosted on the Application Layer in SDN. The LSTM Seq2Seq sub-model of SDPredictNet studied the non-linear features from the traffic parameters present in the dataset and is used to determine the traffic parameters at a particular instant by using the past parameters. The deep neural routing ANN sub-model of SDPredictNet is fed the predicted dynamic traffic data along with static parameters for route prediction in the system via activation of the higher layer switches in the network. By taking into account both static and dynamic network conditions, the network tries to determine the most optimal path for a packet in the given topology.

III. System Design

This section gives a brief description of the design of the SDN system along with the LSTM based Seq2Seq parameter prediction sub-model and ANN routing sub-model of SDPredictNet.

Figure 1 shows the overall architecture proposed in SDPredictNet for SDN routing. The proposed system uses traffic, path and structure parameters to determine the route of activated switches for the given data. The path and structure parameters are static and are given as inputs to the ANN route determination sub-model whereas the traffic parameters are dynamic in nature and hence are predicted using a LSTM based Seq2Seq sub-model. The predicted traffic parameters are fed along with the static parameters and the path taken by the packets are predicted dynamically.

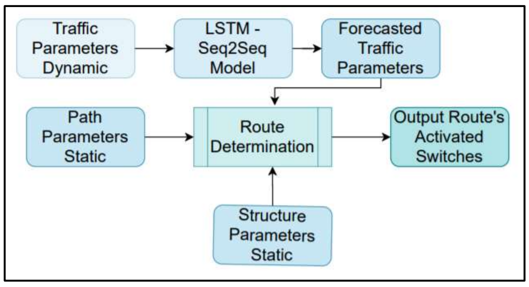
Traffic Parameters Dynamic
LSTM - Seq2Seq Model
Forecasted Traffic Parameters
Path Parameters Static
Route Determination
Output Route's Activated Switches
Structure Parameters Static

Fig. 1. Overall Architecture of SDPredictNet

A. SDN Topology

Topology [21] is the organization of the different network elements in the SDN. A multi-stage circuit switching topology called the Clos topology [22], is widely used in the Cloud-based networks. The first-layer switches in the clos topology refer to the switches connected directly to the hosts present in the system. The n-higher layer switches refer to a network of $n - 1$ higher layer switches.

B. Dataset Description

TABLE I. Types of Parameters

S. No.	Type of Parameters	Description
1	Traffic Parameters	These parameters help to measure the traffic present in the system during the active data flow, given in Table II.
2	Structure Parameters	They identify the hosts actively communicating in the network.
3	Path Parameters	These parameters specify the n-layer switches that are used for the hosts to communicate during a single transmission in the network. These switches are identified using a packet tracer tool such as sflow-rt [25]

The dataset that is used in this paper is collected from the data flows of communication between each host present in the topology and it comprises three types of parameters [23] as seen in Table I. Each parameter can be categorized as static or dynamic features [24]. Static features indicate the network parameters at a particular instant of time as there is no dependency on variation with time. Dynamic or sequential features are time variants and are dependent on the past network parameters. In this paper, structure and path parameters are static and traffic parameters are dynamic in nature. Traffic parameters can be forecasted to take into account their effect on the prediction of data routes in the network. The forecasted traffic parameters along with structure and path parameters are then used to train the neural route prediction sub-model of SDPredictNet.

C. Dataset Preprocessing

The dataset containing the network parameters undergo normalization [26] to scale the value between 0 and 1 to reduce the range of the input values. Min-Max normalization processes all inputs to in the range (0,1) and is calculated using (1). This ensures that speed of learning improves by reducing the time taken to converge on the local minima.

$$\hat{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

TABLE II. Traffic Parameters Collected from the Network

No.	Name	Description
1	Interval	Time Taken between successive packets
2	Ct	TCP connect time
3	Transfer	Amount of data transferred
4	Bandwidth	Bandwisth of the link
5	Write	Total number of socket writes
6	Err	Total number of non-fatal socket write errors
7	Rtry	Number of TCP Retries
8	Cwnd	TCP Congestion Window
9	RTT	Round Trip Time
10	NetPwr	Network Power as Throughput/RTT

D. Sequence to Sequence Modelling using LSTM for traffic prediction

The prediction of traffic in the SDN can help to predict the path in the network for higher layer switches. By writing rules into the flow table of the switches using REST APIs, the delay in the network can be reduced to enhance the performance of the system.

LSTMs are used in prediction as the dependencies on past values are mapped and retained. LSTMs contain three gates: input (i_t), output (o_t) and forget (f_t) gates. These gates are sigmoid in nature and they return 0 if they are blocking information or they return 1 if they allow information to pass through. The gates are represented using equations (2), (3) and (4) [27].

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{3}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{4}$$

Where, $w_x, b_x, x \in i, o, f$, represents the weights and biases for the three gates, h_{t-1} represents output of previous LSTM block and x_t represents the current input.

For each LSTM cell, a candidate cell state is calculated using equation (5).

$$\hat{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c) \quad (5)$$

Using this candidate cell state, the memory vector for the current instant is calculated using equation (6).

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t \quad (6)$$

The input and forget gates determine which values should be retained and which should be dropped. The cell state is then filtered using the output gate and passed through an activation function to get the predicted value as seen in equation (7).

$$h_t = o_t * \tanh(c_t) \quad (7)$$

Sequence to Sequence modelling (Seq2Seq) [28] involves mapping an input sequence of values to an output sequence using an Encoder/Decoder framework. The encoder converts a variable length input sequence to a fixed vector which is then decoded into the output sequence.

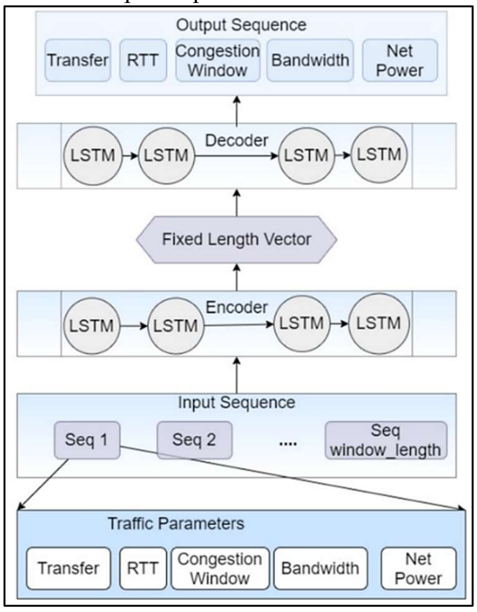

Fig. 2. Encoder-Decoder Architecture of SDPredictNet's Traffic Parameter Prediction Sub-Model

Figure 2 depicts the encoder-decoder framework used for forecasting of parameters. The encoder in the model consists of LSTM units that try to model the dependencies between past and present sequential data. Traffic parameters such as Transfer, Bandwidth, Congestion Window, RTT and NetPower are taken as features for a particular packet. The input sequence is a sequence of traffic parameters for a sliding window of $n-1$ packets. The output sequence is the nth packet that immediately follows the sliding window.

For predicting the traffic parameters for a particular packet, the parameters of a chosen sliding window of previous packets are given as input. The window length is chosen such that the loss of prediction is minimum. The encoded fixed length vector is decoded using a sequence of LSTM units as well.

E. Building a ANN for Deep Neural Routing

Neural routing takes into account each network parameter presents at that instant and determines which is the best possible path for the given packet. The path is mapped by activation of corresponding switches between the host and destination. The structure and path parameters and forecasted traffic parameters are fed to an Artificial Neural Network (ANN) as seen in Figure 3. For a given set of inputs $x = (x_1, x_2, \dots x_n)$, the output at each output node l is given using equation (8).

$$z_l = \sum_{j=1}^{m} v_j \sigma\left(\sum_{i=1}^{n} w_{ji} x_i + b_j\right) \quad (8)$$

Where, m is number of hidden units, v_j is the weight from hidden unit to output unit, n is number of inputs, w_{ji} is the weight from input unit to hidden unit and b_j is the bias associated.

The ANN tries to map the parameters to the output switches. Each output switch is a binary classification model that is independent of the other switches. The number of output nodes in the ANN corresponds to the number of switches present in the network topology. The activation of switches is to be predicted solely based on the network parameters without taking into consideration the activation of the other switches.

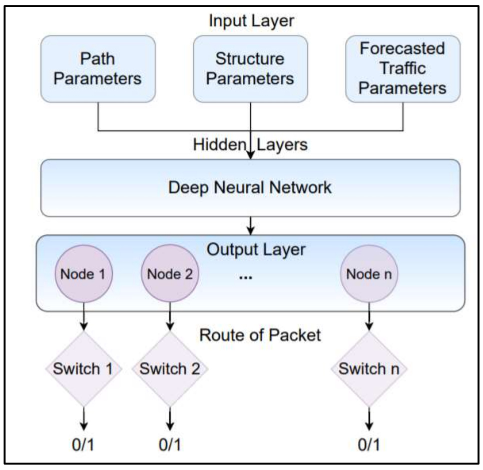

Fig. 3. SDPredictNet's Neural Routing Model using Artificial Neural Networks

F. Configuring REST APIs in the SDN

The communication between the applications and services hosted on the SDNs and the SDN Controller is enabled using Representational State Transfer or REST APIs [29]. To retrieve the switch flow logs and update the flow table, OFCTL_REST API [30] is used. Each data packet has a timeout defined, which is the maximum time a flow entry is maintained in the switch tables. Each switch is updated with the IP Address and the Port number of the packet to check if there is a matching filed in the

Flow Table for the data packet. The flow is logged only if a match is found or the packet will be dropped if the match isn't found in the switch Flow Table.

IV. Implementation

This section gives the implementation details of the Deep Neural network along with its integration with the SDN. The project was set up in a Windows 10 with Intel i7 7th Gen Processor with NVidia GeForce 940MX with Oracle VM Virtual Box with 4GB RAM allocation to host the Mininet version 2.2.1 with OpenFlow 1.3 on Ubuntu 18.04. RYU controller 4.34 is established in the SDN as the central coordinator of the network making up the Control Plane with OpenVSwitch 2.14.

A. SDN Setup

In this paper, the SDN network is built using clos topology [31] with double-layered switches as seen in Figure 4. The SDN network controlled by Ryu Controller [30] has 8 hosts connected to 7 switches. Clos topology is implemented in this paper as it is one of the most common topologies and it helps establish a multi-layer switching network. Horizontal scaling and lowered latency rates helps efficient access of data within the network. The network is also fault tolerant and helps to route packets through the alive switches in case of switch failure.

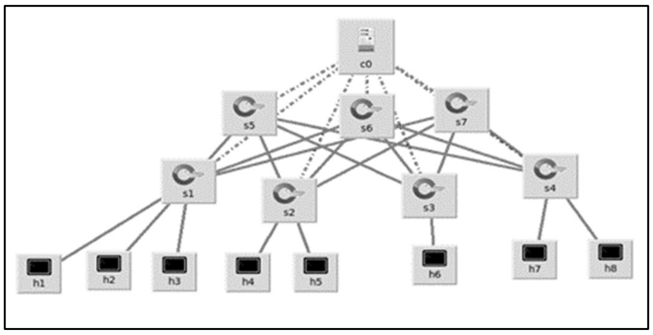

Fig. 4. SDN with Clos topology constructed on the Mininet interface with Controller c0, connected to switches s1-s7 with hosts h1-h8.

B. Dataset collection using the iPerf command

1) Traffic parameters

Using the iPerf command in the Xterm terminal of hosts in the SDN, traffic parameters for the network are extracted. The iPerf tool [32] sets up a client server link between a source host and a destination host and uses this to measure network link quality and bandwidth. The parameters that the iPerf command uses is given in Table III.

2) Structure Parameters

Simple ping commands help establish communication between source and destination hosts. The Xterm terminal is used to transfer data from the source host.

3) Path Parameters

Sflow-rt helps trace the path followed by the data packets within the network. The switches that transfer the data packets during the current transfer are recorded in the dataset. The first layer switches connected to the host are always active during data transfer. The traffic state in the system determines which second layer switches are activated. In Figure 5, it is observed that the data transfer between the hosts is actively handled by the second-layer switch S5.

TABLE III. iPerf Commands on the Server and the Client Side

Server side	
Command	**Description**
-s -server	Starts iPerf in server model and waits for an iPerf client to contact it
Client side	
Command	**Description**
-c -client <Host>	Starts iPerf in client mode and connects with the iPerf server <Host> (IP address or DNS name)
-t time <Time> (default: 10 seconds)	Sets the duration of the connection in seconds
-I -interval	Seconds between periodic bandwidth reports
-b -bandwidth <BW>	Sets the bandwidth for data transfer
-e -enhanced output	Display enhanced output reports
-w -window size <Size>	TCP window size
-f -format	Format to report: Kbits, Mbits, Kbytes, Mbytes
-h -help	Outputs the help text
-v -version	Outputs the version

C. Preprocessing the Dataset

Using the tools mentioned in the previous section, a dataset comprising 1,048,575 instances is collected. The dataset was then split into a training and testing set with a ratio of 80:20. Normalization was performed on the clean dataset using Keras' MinMaxScaler [33] for each input field which subtracts the minimum value from each instance and divides the difference by the maximum value for that instance.

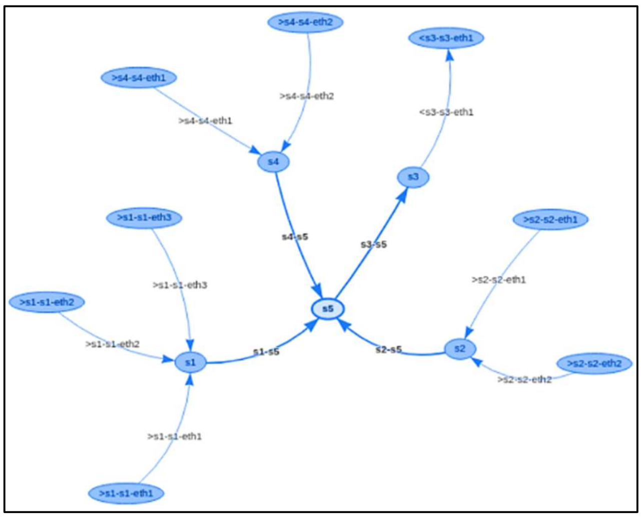

Fig. 5. Tracing of the OpenFlow packets using sflow-rt to identify the active Layer-2 switch(S5)

D. Sliding Window for Traffic Parameter Prediction

Sliding windows help create a sequence of inputs that can be fed into the LSTM based Seq2Seq model. For each packet $i + 1$, a sliding window of i preceding packets are taken as input. This was repeated for each packet to generate a sliding window of inputs and outputs as seen in Figure 6. Mean Absolute Error (MAE) is measured with variation of sliding window size as seen in Figure 7. A sliding window size of 100 packets is taken as the MAE with a value of 0.0094 is least for that window size.

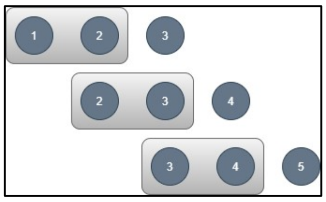

Fig. 6. Sliding window protocol

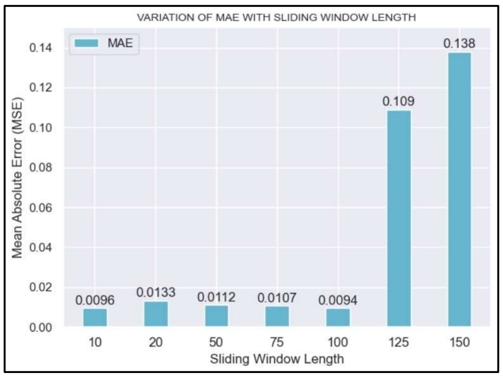

Fig. 7. Variation of MAE with Sliding Window Size

E. LSTM based Seq2Seq Model for Traffic Parameter Prediction

The LSTM based Seq2Seq sub-model makes use of an encoder-decoder architecture. The encoder consists of a LSTM layer with 32 nodes. The inputs when passed through the encoder are converted to fixed length hidden state and context vectors [34]. The fixed length vectors are decoded by using a LSTM layer with 32 * 2 nodes. The output of the decoder is fed to 5 nodes which are activated by ReLu activation function [35] each of which corresponds to a specific traffic parameter. ReLu is used as the predictions lie within a continuous range of values between 0 and 1. The prediction model is trained using the parameters given in Table IV.

TABLE IV. TRAINING PARAMETERS OF LSTM BASED SEQ2SEQ SUB-MODEL

Parameter	Value
Epochs	5
Optimizer	RMS Prop
Loss	Mean Absolute Error
Batch Size	256
Learning Rate	0.001

F. Artificial Neural Network for Path Prediction

The ANN sub-model used has 23 nodes corresponding to the structure, path and forecasted traffic parameters. The sub-model has one hidden layer with 16 nodes activated using ReLu activation function. ReLu [35] is used to remove any negative values during gradient descent and thus helps increase speed of convergence during learning. The outputs from the hidden layer are then fed to an output layer consisting of 7 nodes, each corresponding to a switch in the topology, and are activated using Sigmoid activation function [35]. The sigmoid activation function is used as each node represents the activation state of the switch which is either 0 or 1. The path is determined by taking the sequence of activated switches. The path prediction model is trained using the parameters specified in Table V.

TABLE V. TRAINING PARAMETERS OF ANN BASED PATH PREDICTION SUB-MODEL

Parameter	Value
Epochs	12
Optimizer	Adam
Loss	Binary Cross Entropy
Batch Size	128
Learning Rate	0.001

V. RESULTS AND ANALYSIS

In this section, the performance of the traffic forecasting sub-model and the neural routing sub-model are studied. The impact that the traffic parameters have on the path taken by packets is analyzed by comparing the performance of the neural routing model with the predicted values.

A. Training plot of the LSTM model for traffic prediction in the network

The training of the neural sub-models is done on Google Colab with Intel(R) Xeon(R) CPU @ 2.20GHz Processor, 13 GB RAM and 12GB NVIDIA Tesla K80 graphics processor.

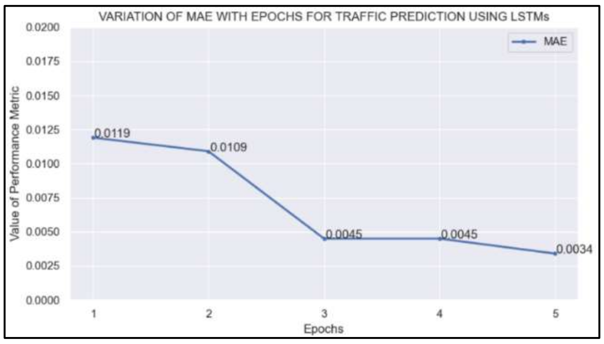

Fig. 8. Training plot of the LSTM sub-model for predicting traffic in SDN

The LSTM deep neural sub-model is used for the prediction of traffic in the network from the traffic data collected from the existing multi-layer topology. The sub-model is trained with sequential data collected from the packet tracer tool for 5 epochs along with the Mean Absolute Error value and the results are plotted in Figure 8.

Mean Absolute Error (MAE) [36] is defined as the measure of errors between paired observations expressing the same phenomenon and is calculated using equation (9).

$$MAE = \sum_{i=1}^{n} |y_i - x_i| / n \qquad (9)$$

Where, y_i is the prediction, x_i is the actual value and n is the number of data points.

The value of MAE reduces to 0.0034 at the fifth epoch and the training is stopped to avoid overfitting the data.

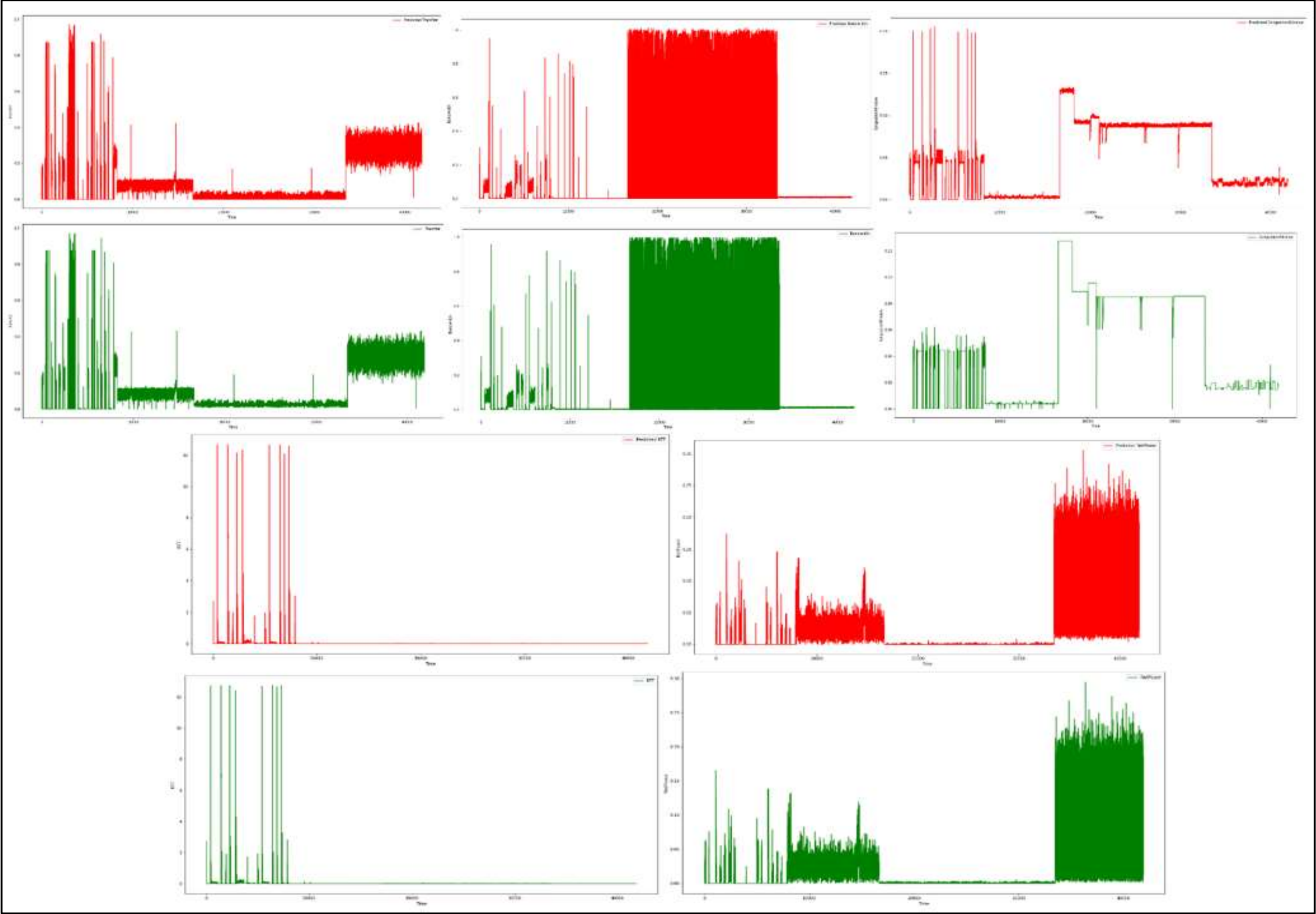

Fig. 9. The comparison of the actual values (in red) w.r.t. time(s) of a) Transfer, b) Bandwidth, c) CongestionWindow, g) RTT h) NetPower with the predicted values (in green) by the LSTM model for d) Transfer, e) Bandwidth, f) CongestionWindow, i) RTT and j) NetPower.

B. *Training plot of Artificial Neural Network for Routing in the Network*

The training plot of the deep neural sub-model for routing in the SDN was performed by an ANN whose Accuracy and Loss values during training are plotted in Figure 10. The training accuracy reached a peak of 99.89% with the loss value at 3.43.

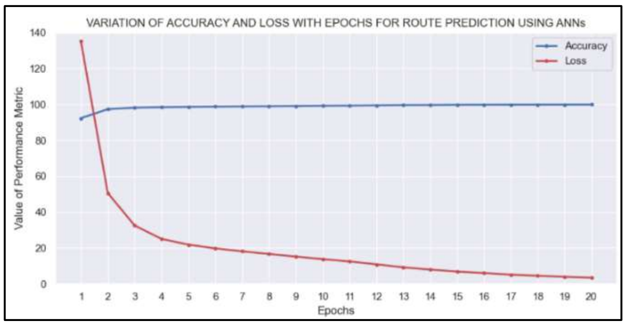

Fig. 10. Plotting the Accuracy and Loss of the ANN model for network routing

C. *Evaluation of the LSTM-based traffic prediction model using MSE and RMSE metrics*

The model for traffic prediction is evaluated using MSE and RMSE values as given in Table VI. The Mean Squared Error (MSE) [36] is the average of the squared differences between the actual and the predicted value by the model as is given by equation (10). The Root Mean Squared Value (RMSE) [36] is the square root of the MSE value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (10)$$

TABLE VI. MSE AND RMSE VALUES FOR THE PREDICTION MODEL

Metric	MSE	RMSE
Transfer	9.3649374e-05	9.677261e-03
Bandwidth	2.1541771e-05	4.6413112e-03
Congestion Window	2.3106697e-05	4806943 e-03
RTT	14.813986e-05	1.2171272e-02
NetPower	1.0653032e-05	32638984e-03

D. *Variation of predicted values of Traffic Parameters with the actual values*

The variation of the predicted values (in green) of Transfer, Bandwidth, Congestion Window, Round Trip Time and Network Power with the actual dataset (in red) collected from the network are analysed in Figure 9. It can be seen that the actual and predicted values are similar to each other indicating reduced loss values.

E. *Performance of Neural Routing with Actual Traffic*

The deep neural network routing performance is evaluated against actual traffic collected from the SDN in Figure 11. The value of accuracy reaches 98.87% with a low performance loss at 0.49.

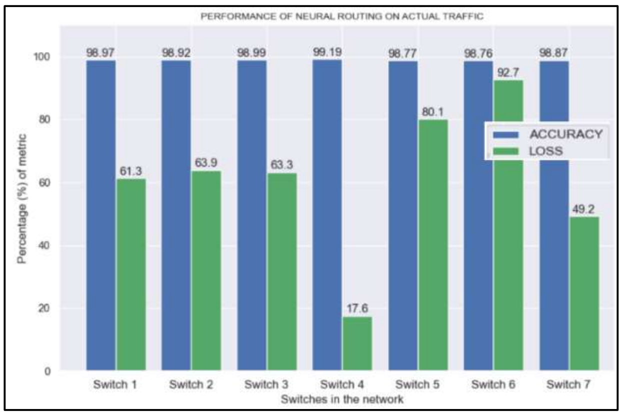

Fig. 11. Performance of the deep neural routing model on actual traffic

F. *Performance of Neural Routing Model with Predicted Traffic*

The deep neural network routing performance is evaluated with the traffic predicted by the LSTM model as seen in Figure 12. The value of accuracy reaches 99.99% with a low performance loss at 0.31. It can be seen that the route prediction with forecasted data is as accurate as route prediction with actual data. This is because traffic in the network is essential in identifying the most optimal route to be taken. Prediction of traffic ensures that the model takes into account the effect from previously sent packets and tries to determine the path with least cost.

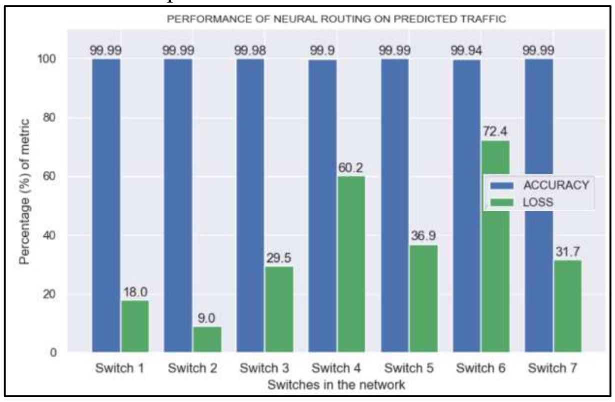

Fig. 12. Performance of the deep neural routing model on predicted traffic

G. *Performance metrics for the comparison of neural routing in the predicted traffic vs actual traffic*

The performance of the ANN sub-model for neural routing with the actual data is compared against neural routing with the predicted data for the clos network topology emulated in the system as seen in Figure 13. Predicted traffic helps the neural model decide which switch activations are most optimal and hence increase the accuracy of prediction of path of packets. Considering traffic parameters as dynamic value, helps the model to forecast regions of heavy traffic and the packets can be forwarded in alternate routes.

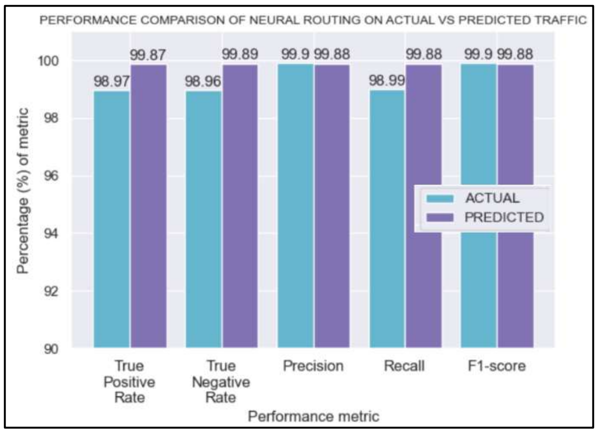

Fig. 13. Performance comparison of the deep neural routing model on actual and predicted traffic in the network

VI. Conclusion

The implementation of multi-layered clos topology-based system SDN was done in Mininet with a Ryu Controller. Sflow-rt, was used for the analysis of the flows in the network to collect the network, structure and traffic parameters from the dynamic network. In this paper we propose SDPredictNet, a deep neural model for route prediction of packets in SDNs. SDPredictNet comprises of a LSTM based Seq2Seq traffic prediction sub-model and an ANN based route prediction sub-model. The values of Bandwidth, Network Performance, RTT, Congestion Window and Transfer were fed as sequential data to the LSTM sub-model for traffic prediction. The traffic prediction will help to update the flow table of the switches present in the network to increase efficiency of performance by reducing delays in the network. The deep neural routing sub-model predicts routes with an accuracy of 98.87% for actual data and 99.88% for forecasted data.

VII. Future Works

The implementation of a reinforcement learning model hosted on the SDN Application Layer can help to predict dynamic flows in the network for inter-network communication between the hosts. The model can be extended to create a dynamically adaptable traffic prediction system in the network to identify and report congestion points in the network for analysis and their prevention in the SDN. Use of Generative Adversarial Networks for the prediction and simulation of traffic in the network can help optimize the performance of the SDN with the multi-layered switch overlay. These networks also learn the internal representation of the collected data along with its density representation. The reduction in the time for the updating the flow table of switches opens such frameworks to a number of opportunities for their employment in time-critical applications.

REFERENCES

[1] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, "Network Functions Virtualization: The Long Road to Commercial Deployments," IEEE Access, vol. 7, pp. 60439–60464, 2019.

[2] A. Kumari, J. Chandra, and A. S. Sairam, "Predictive Flow Modeling in Software Defined Network," in TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), pp. 1494–1498, 2019.

[3] J. Xie et al., "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 393–430, 2019.

[4] C. Metter, S. Gebert, S. Lange, T. Zinner, P. Tran-Gia, and M. Jarschel, "Investigating the impact of network topology on the processing times of SDN controllers," in 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1214–1219, 2015.

[5] A. Lazaris and V. K. Prasanna, "Deep Learning Models For Aggregated Network Traffic Prediction," in 2019 15th International Conference on Network and Service Management (CNSM), pp. 1–5, 2019.

[6] Y.-F. Liu, K. C.-J. Lin, and C.-C. Tseng, "Dynamic Cluster-based Flow Management for Software Defined Networks," IEEE Transactions on Services Computing, pp. 1–11, 2019.

[7] Y. Wang, D. Jiang, L. Huo, and Y. Zhao, "A New Traffic Prediction Algorithm to Software Defined Networking," Mobile Networks and Applications, pp. 1–10, 2019.

[8] A. Chaturvedi, D. Somwanshi, M. Bundele, and C. Dubey, "Monitoring and Traffic Optimization Using Vertical Controller in Multi-domain SDN," Algorithms for Intelligent Systems, pp. 349–357, 2020.

[9] G. Gong, X. An, N. K. Mahato, S. Sun, S. Chen, and Y. Wen, "Research on Short-Term Load Prediction Based on Seq2seq Model," Energies, vol. 12, no. 16, p. 3199, 2019.

[10] Z. Wang, X. Su, and Z. Ding, "Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture," IEEE Transactions on Intelligent Transportation Systems, pp. 1–11, 2020.

[11] S. P. Sone, J. J. Lehtomaki, and Z. Khan, "Wireless Traffic Usage Forecasting Using Real Enterprise Network Data: Analysis and Methods," IEEE Open Journal of the Communications Society, vol. 1, pp. 777–797, 2020.

[12] V. A. Le, P. L. Nguyen, and Y. Ji, "Deep Convolutional LSTM Network-based Traffic Matrix Prediction with Partial Information," in 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 261–269, 2019.

[13] B. Mao, F. Tang, Z. Md. Fadlullah, and N. Kato, "An Intelligent Route Computation Approach Based on Real-Time Deep Learning Strategy for Software Defined Communication Systems," IEEE Transactions on Emerging Topics in Computing, pp. 1–12, 2019.

[14] Y. Zuo, Y. Wu, G. Min, and L. Cui, "Learning-based network path planning for traffic engineering," Future Generation Computer Systems, vol. 92, pp. 59–67, 2019.

[15] Z. Shu et al., "Traffic engineering in software-defined networking: Measurement and management," IEEE Access, vol. 4, pp. 3246–3256, 2016.

[16] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "SDN-based real-time urban traffic analysis in VANET environment," Computer Communications, vol. 149, pp. 162–175, 2020.

[17] Y. Yang and H. Sun, "Research on Traffic Optimization Scheme of SDN Network Based on ME-HMM," Journal of Physics: Conference Series, vol. 1624, pp. 1-6, 2020.

[18] T. Shreya, M. M. Mulla, S. Shinde, and D. G. Narayan, "Ant Colony Optimization-based Dynamic Routing in Software Defined Networks," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7, 2020.

[19] A. Abugabah, A. A. Alzubi, O. Alfarraj, M. Al-Maitah, and W. S. Alnumay, "Intelligent Traffic Engineering in Software-Defined Vehicular Networking Based on Multi-Path Routing," IEEE Access, vol. 8, pp. 62334–62342, 2020.

[20] Y.-J. Wu, P.-C. Hwang, W.-S. Hwang, and M.-H. Cheng, "Artificial Intelligence Enabled Routing in Software Defined Networking," Applied Sciences, vol. 10, no. 18, pp. 6564-6580, 2020.

[21] L. Mamushiane, J. Mwangama, and A. A. Lysko, "Given a SDN Topology, How Many Controllers are Needed and Where Should They Go?," in 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–6, 2018.

[22] A. Singh et al., "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 183–197, 2015.

[23] D. J. Hamad, K. G. Yalda, and I. T. Okumus, "Getting traffic statistics from network devices in an SDN environment using OpenFlow," in Information Technology and Systems 2015, pp. 951–956, 2015.

[24] A. Leontjeva and I. Kuzovkin, "Combining Static and Dynamic Features for Multivariate Sequence Classification," in 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 21–30, 2016.

[25] "sFlow-RT," sflow-rt.com. [Online] https://sflow-rt.com/ (accessed Dec. 28, 2020).

[26] S. Bhanja and A. Das, "Impact of Data Normalization on Deep Neural Network for Time Series Forecasting," arXiv:1812.05519 [cs, stat], 2019, Accessed: Dec. 28, 2020. [Online]. Available: https://arxiv.org/abs/1812.05519.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," Transportation Research Part C: Emerging Technologies, vol. 107, pp. 287–300, 2019.

[29] W. Zhou, L. Li, M. Luo, and W. Chou, "REST API Design Patterns for SDN Northbound API," in 2014 28th International Conference on Advanced Information Networking and Applications Workshops, pp. 358–365, 2014.

[30] Y. Li, X. Guo, X. Pang, B. Peng, X. Li, and P. Zhang, "Performance Analysis of Floodlight and Ryu SDN Controllers under Mininet Simulator," in 2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops), pp. 85–90, 2020.

[31] R. R. Krishnan and N. Figueira, "Analysis of data center SDN controller architectures: Technology and business impacts," in 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 104–109, 2015.

[32] M. Hasan, H. Dahshan, E. Abdelwanees, and A. Elmoghazy, "SDN Mininet Emulator Benchmarking and Result Analysis," in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 355–360, 2020.

[33] "sklearn.preprocessing.MinMaxScaler," Scikit-learn.org, 2019. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (accessed Dec. 29, 2020).

[34] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5060–5064, 2016.

[35] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in 2018 Chinese Control And Decision Conference (CCDC), pp. 1836–1841, 2018.

[36] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model," IOP Conference Series: Materials Science and Engineering, vol. 324, 2018.