Peer-to-Peer Networking and Applications (2022) 15:751–767
https://doi.org/10.1007/s12083-021-01280-6

An intelligent traffic prediction framework for 5G network using SDN and fusion learning

K. Tamil Selvi[1] · **R. Thamilselvan**[2]

Received: 4 July 2021 / Accepted: 17 December 2021 / Published online: 11 January 2022

Abstract
Traffic prediction and analysis is an important part of traffic engineering in Software Defined Networking (SDN). Effective forecasting of network traffic needs the composition of capturing the dependency features. SDN is embedded in 5G networks which decouples control plane from the data plane for network programmability. Flow based SDN architecture provides precise prediction of network traffic with fine granularity of data plane network statistics. Deep learning models like Gated Recurrent Unit (GRU) enables the time series forecasting with high performance. The major problem with usage of deep learning models is its communication overhead with the convergence of the model. To provide a communication efficient and intelligent traffic prediction framework, fusion learning is used between the data plane and control plane of the SDN environment. Fusion learning provides the prediction model with the exchange of model parameters of the SDN client models and its data distribution with single communication. The global topology manager of the SDN controller with the deep neural prediction model enhances the forecasting of the network traffic. Time series GRU captures only the temporal dependency. To handle dynamics of the network traffic and efficient capture of traffic pattern, spatial dependency must be captured. The diffusion convolution operation embedded in the GRU capture both spatial and temporal dependency of the features in encoder-decoder architecture. The stochastic gradient based scheduled sampling improves the performance of the prediction model with the optimal decay rate. The proposed framework is tested with the simulated data on Abilene network topology with RYU SDN controller. The experimental results exhibit improved accuracy in both local and global model of 87%—94%. Further, hyperparameter tuning is done for precise forecasting of network traffic with the minimal prediction error of 7.98% for one hour horizon. The proposed prediction model observes improvement over the base line prediction models.

Keywords Traffic prediction · Fusion learning · Diffusion convolution · Gated recurrent unit · Software defined networking

1 Introduction

The Fifth Generation (5G) network provides high capacity with more volume of users. One of the salient feature of the network is zero latency to achieve high reliability [1]. The complex 5G network requires simplified solution to accomplish its characteristics within their network control. This can be provided by machine learning and deep learning algorithm to discover the hidden pattern thus providing clear picture of cause and effect in the data enabled network. The performance of the network can be enriched with the identification of the demand–supply scenarios and deployment of resources to avoid the congestion and service disruption in the network. In the network with Bigdata, processing capability of the nodes must to enhanced to handle the learning mechanism with data. And also, communication bottleneck should be avoided with the transmission of data and model parameters between the client and server architecture in the learning environment. The learning model can be centralized or distributed client–server architecture. The stand-alone model provides high latency which is not suitable for 5G environment [2].

Software Defined Networking (SDN) and Network Function Virtualization (NFV) are enabling technologies in 5G network. The cloud computing paradigm is leveraged by

✉ K. Tamil Selvi
ktamilselvikec@gmail.com

R. Thamilselvan
r_thamilselvan@kongu.ac.in

[1] Department of Computer Science and Engineering, Kongu Engineering College, Erode, Tamil Nadu 638060, India

[2] Department of Information Technology, Kongu Engineering College, Erode, Tamil Nadu 638060, India

SDN in network management and control. SDN network has centralized controller which optimize the performance of the network by dynamically steer and manage the traffic flow across the nodes and networks and orchestration of network resource allocation. Artificial Intelligence (AI) can be induced in the centralized SDN controller for network optimization and performance improvement. Even the SDN controller can be used in the distributed learning techniques like federated learning [3]. The communication overhead and latency are higher in the centralized architecture as the data and model parameters exchanges take place for the convergence of the model. The parallelization technique like data parallelism and model parallelism used in distributed learning environment provides application in many domains like health care, tele communication, automotive industry and so on. The centralized client–server machine learning or deep learning results in slower convergence of the model and hence not suitable for delay-sensitive applications.

The 5G network provides three classification of services namely enhanced Mobile Broadband (eMBB), Ultra-reliable and Low Latency Communications (URLLC) and massive Machine Type Communication (mMTC). eMBB requires high data rate with wider coverage, stringent requirement of latency and reliability is imposed by URLLC and support for wide variety of devices in mMTC. All these usecases need to satisfy the required Quality of Service (QoS) and need for environment study and its impact can be done through machine learning or deep learning techniques. Even it is wiser to use the learning techniques in the 5G environment, learning by the model, parameter sharing and converge of the network model may results in communication overhead and latency. Hence communication efficient and low latency model is the need of the hour in the 5G uses cases. Distributed machine learning mechanism [4] can support reliability but results in delay and communication overhead. To reduce this, federated learning can be used to reduce the communication overhead with less amount of data transmission between distributed client and the centralized SDN controller.

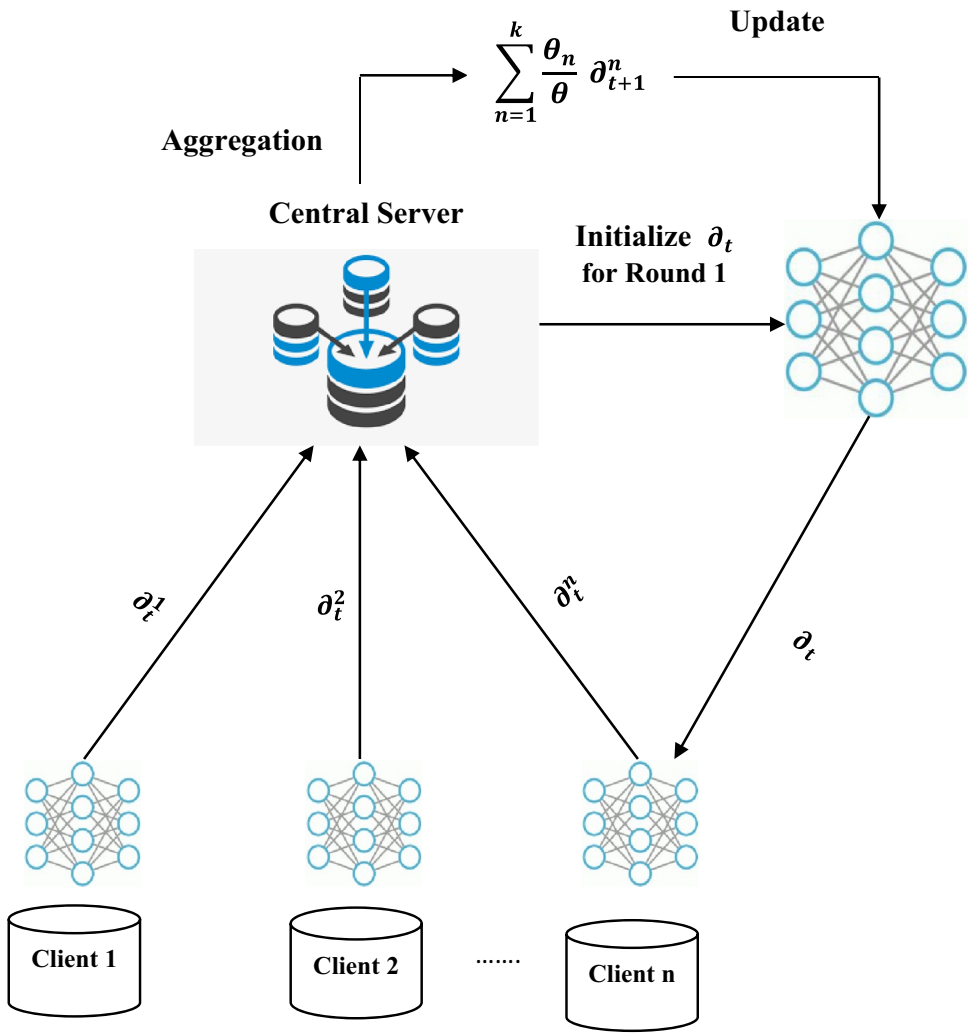
Federated learning [5] finds its application in many domains next word prediction, emoji prediction, prediction models [6] in health care to ensure the privacy of the patient data and many more. In federated learning, the centralized model is trained on data that is distributed among the multiple clients as shown in Fig. 1. The central model coordinating the training of the model does not receive the data from the client but only the model parameters. At each iteration, the central server model selects a subset of clients which receive the current model parameters. The clients will run stochastic gradient descent in its nodes and return the updated model parameters to the central server model. These model parameters aggregation is performed and the training process is repeated until the convergence of the model. The several problems arise in the federated learning environment are communication bottleneck, need for network optimization and so on. With the larger model, the number parameters exchanged between the clients and the central server increases exponentially. This communication overhead results in delay in convergence of the model and does not suit well for 5G requirements. To overcome the pitfalls in federated learning, Fusion learning [7] is proposed in SDN environment for traffic prediction for efficient performance of the network.

One-shot learning based on ensemble technique is used to reduce the communication bottleneck with improved performance. Instead of receiving the model parameters and updates from all the associated clients, selection based on certain constraints can be imposed. The ensemble method is used for selection of optimal selection of clients. Based on cross validation, only those models of clients that satisfy the baseline performance will send the model updates to the server. The ensembling is performed by the server on the best performing clients and hence the performance of the overall model will be improved. Another ensembling technique is based on data selection in which only those clients that have undergone the given baseline amount of training on local data is allowed for sending parameter updates to the server. Random selection of clients can also be applied in the ensemble methods. All these mechanisms improve the quality of the global model and hence the performance metrics enhancement. In all the above methods, only one round of communication takes place between client and server with the selection of best local models.

Distilled one-short learning [8] is the enhanced version of one-shot federated learning. Instead of sending bulk number of gradients and weights of deep networks, the clients distill their data and send only the synthetic data images to the server. Soft reset and random masking can be used to handle the poor distribution of client data. Further to minimize the number of contact rounds of transmission between client and centralized server, fusion learning is used. Instead of working only with model parameters for each round of updates, the client nodes send model parameters for its data and its data distribution parameters. Based the number of features in the dataset, the data distribution parameters have to be computed only once and transmitted to the server. Based on the received distribution parameters, the server will generate the synthetic data and the cumulative dataset is constructed. The global model is constructed and the updated model parameters are distributed among the clients. The learning takes in both client and server with their own local data only and distribution of data provides the insight on the model performance. The proposed framework limits the communication overhead with improved performance.

The proposed framework consists of the centralized SDN controller which acts as aggregator of local models at server

Fig. 1 Federated Learning



in the federated learning environment. The SDN controller has the global topology view of the network of clients and its communication path. The clients have their own local data consists of features for the model to explore the hidden patterns or data insights. Instead of sending the updated gradients and weights to the SDN controller, the distribution of features of the local data is transmitted in one communication round to the server, i.e. SDN controller. The overall aggregated model is cumulated at the SDN controller to get the global model. This fusion learning finds its place in the dynamic internet environment and improved efficiency. Further the model performance is improved with the selection of the best clients using ensembling techniques based on baseline performance metrics. The traffic prediction model is proposed using Gated Recurrent Unit (GRU).

Network traffic prediction provides the future status of the network requirements and the statistics of the environment. Network traffic prediction along with network modeling are the key performance indicators in the wireless networks. An accurate prediction network traffic in the dynamic network environment provides key step to improve the performance of the network. The spatial and temporal variations in the network traffic also explores the hidden pattern and hence accurate prediction. Spatial features are extracted using the attention mechanism and temporal features are obtained using sequence modeling. The flows [9] in the SDN are characterized by 12-tuples and ensemble based fusion learning provides traffic prediction with minimal communication overhead.

The remainder of the article is organized as follows. Section 2 discusses the relevant works in the area of traffic prediction, federated learning and traffic characteristics. Methodologies for implementation of traffic prediction model is elaborated in Sect. 3. The model is tested with simulated data and evaluated its performance in Sect. 4. Finally, the article concludes in Sect. 5 with discussion on the scope for future enhancement and application.

2 Related works

The continuous exponential increase of network traffic needs accurate prediction of traffic flow for better customer experience. The network traffic prediction and monitoring are essential for efficient network management with improved efficiency. Time series data analysis is mostly used for network traffic prediction. Linear analysis models like Auto Regressive Moving Average model (ARMA), Auto Regressive Integrated Moving Average model (ARIMA), Fractional Auto Regressive Integrated Moving Average model (FARIMA), etc., are used. These models do not capture the non-linear trends in traffic like uncertain fluctuations. To address these issues, models like Support Vector Machine (SVM), Least Square Support Vector Machine (LSSVM) were preferred to address the non-linear characteristics of network traffic to improve the prediction accuracy. However, autocorrelation factor of network traffic is not accounted. The autocorrelation captures both the trend and periodicity of the network traffic [10]. In SDN environment, the flow based traffic prediction is done using equal interval sampling algorithm and ARMA model for time-series estimation of parameters [11]. The non-stationary SDN traffic is smoothened to transform into stationary sequence to apply the linear model. The SDN flow sequence are obtained using equally spaced sampling algorithm with the appropriate time slice.

Most of the studies reveals that the network traffic is self-similar nature with high burstiness and follows heavy tail distribution. The self-similarity and heavy tail distribution of the network traffic can be captured with alpha stable model. But this model does not hold for infinite variance and results in high least square errors. To address the above challenge, covariation orthogonal [12] criteria is applied to extract the relationship between past and future traffic process. This model is effective in capturing burstiness among the self-similar network traffic. The new neural network framework is based on echo state network [13] which provides stronger non-linear data processing with short term memory. The generative adversarial network is used with echo state network as generative model and feed forward neural network as discriminative model for the prediction of network traffic. The various characteristics of network traffic are mutability, chaos, self-similar and so on.

Deep learning algorithms capture the inherent features of the network traffic and provide good performance measures. The spatial and temporal correlations of the features provide the exploration of the inherent flow of network traffic and its prediction. The stacked denoising autoencoder provides scope for learning the generic features of network traffic and the model is trained using greedy approach layer-wise [14]. Dynamic allocation of resource with minimal energy consumption is proposed in [15]. Short-term prediction of network traffic provides a trend pattern and the predictor estimation is of import concern. This can be implemented by wavelet transforms with double exponential smoothing. Non-linear characteristics and complex nature of network traffic provides the exploration of spatial and temporal correlations. Attention mechanisms are embedded with the prediction model [16] for incorporation of intermediate results for improved accuracy. LA-ResNet based architecture with three components namely residual network, the recurrent neural network and attention module.

Most of the existing literature reveals that spatial and temporal feature and its correlations provides accurate insight on traffic prediction. Statistical modeling and deep learning models find suitable for capturing spatial and temporal features of the network traffic data. Convolution Neural Network (CNN) and LSTM is used to capture the spatio-temporal characteristics conventionally. These models lag in capturing temporal pattern for long-term or short-term dependencies. To address these issues, Spatio-Temporal U-Net (ST-UNet) is proposed [17] with graph modeling of time series network data. The dynamic spatio-temporal graph is used for the given network and graph convolution operation is performed to capture the spatio-temporal correlations. 3D Temporal Graph Convolutional Network (3D-TGCN) [18] is employed for traffic prediction that constructs the temporal adjacency matrix to learn the temporal patterns of the data-driven network traffic. The static and dynamic information about the network traffic is captured by graph convolution operations in non-Euclidean structure. Diffusion Convolutional Recurrent Neural Network (DCRNN) uses bidirectional random walk on the convolution graph to capture spatial dependency and encoder-decoder with scheduled sampling to capture temporal dependency [19].

The parameters learning models can be reduced with Chebyshev polynomial approximation with minimal training time [20] in the graph convolution mechanism. The temporal features are extracted using the gated recurrent units. For accurate forecasting of network traffic, learning models employed requires large volume of data. The minimal data with the employment of network traffic prediction results in minimal accuracy. And also, privacy of data also plays a crucial role in many domains like inter-cellular traffic prediction. To overcome these issues, federated learning approach is used. The centralized global model aggregates the local models learning parameters and provides the updated values without sharing of data. Federated gated recurrent unit [21] is used to predict the traffic flow in the network with security mechanisms. There is a limitation of the model due

to high communication overhead for the convergence of the model. The network traffic prediction model cannot tolerate the delay and communication overhead, though there is accurate prediction and privacy preservation. This is due to the fact that the 5G network needs minimal latency with reliable data communication. Hence alternative methods with minimal communication delay are explored. Blockchain based federated model is used for preventing security attacks in the traffic flow prediction environment [6]. Since the central global model is updated based on local models learning parameters, poisoning of local models results in degradation of model in prediction of network traffic. In the distributed traffic prediction area, local differential privacy preserving policy can be used for effective protection of data and the prediction models.

The main contribution of this work is summarized as follow:

- The traffic prediction model with fusion learning between the data plane and control plane of the SDN environment is proposed with reduced communication overhead.
- The efficient forecasting model is built with diffusion convolution operations in GRU to capture the spatial dependencies and temporal dependencies of the features of the network traffic data.
- The experimental analysis is done with the simulated data for various time horizon and compared with the baseline models

3 Methodology

The centralized learning results in single point of failure and may incur delay which is not suitable for 5G usecases like URLLC. So distributed learning can be adopted with minimal communication overheads, the technique being used is fusion learning which is an enhanced version of federated learning. The traffic prediction framework consists of fusion learning for deep learning predictive model with the exchange of distribution parameters in single communication overhead between the global model and the local models. Two ways of achieving minimal communication overhead and latency in 5G usecases are distribution parameter exchange and Generative Adversarial Network (GAN) generators. The statistical mechanism suits well in SDN context since the controller may be communicated during the OpenFlow message interaction and the distribution parameter can be embedded with its without additional communication overheads. GAN is a combination of generator and discriminator with the generator being shared with the central model. Sharing of generator frequently for the improved accuracy of the training model results in degradation of the performance of the traffic prediction model. The accurate mapping of the data distribution parameters on the client local data is the contributing context in the fusion learning environment. More accurate estimation of distribution parameter for the client data results in improved aggregated data in the global model.

3.1 Fusion Learning

In fusion learning, the number of communications between the central server and the clients is reduced to one. It is commonly called as one-shot federated learning. To reduce the multiple iterations on training, the generative information is sent from the client to the server. With this generative information, the central server can generate sample data for model aggregation and updation. The effectiveness of the generative information provides the nature of the original data and the similar model. There are different approaches to derive the generative information. They are listed below.

a Distribution Parameters

The statistical techniques are used to estimate the distribution of data of local models. This parameter distribution is sent to the server along with the model parameters. The central server picks up the random point from the distribution to train its model. The various distribution available in the statistical model for the approximation of the client data is listed in Table 1. To find the appropriate and the most suitable distribution, Kolmogorov–Smirnov (KS) test is performed which assign score to the individual distribution for the client data. The distribution that scores the maximum will be assigned for the client data.

b GAN Generators

Generative Adversarial Networks is used to generate new and synthetic data rather than passing real data. It consists of generators and discriminators. The underlying pattern of the data is learnt by the generator and the sample with maximal accuracy is produced. The data validation is verified by the discriminator. The convergence between the generator and discriminator end with successful fool of the discriminator by the generator. Now the generator is sent to the central server for the training of the model.

Table 1 Distribution types for verification of client data

norm	exponweib	weibull max	weibull min	hi2
pareto	lognorm	beta	chi	pearson3
genextreme	expon	cauchy	cosine	powerlaw
gamma	logistic	lomax	maxwell	rdist
uniform	vonmises	wald	wrapcauchy	erlang

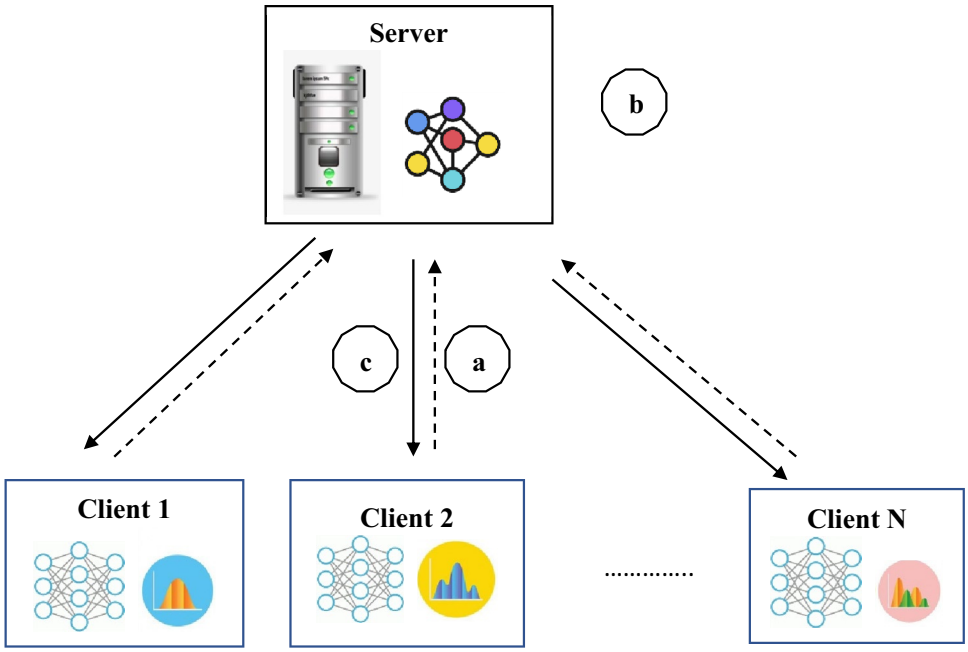


Fig. 2 Fusion Learning Architecture

The fusion learning architecture consists of the clients with the local model and the central server with the global model as shown in Fig. 2. For each feature in the client dataset, the data distribution is estimated and using KS test optimal distribution parameter will be chosen. The local models in the clients enter into the training phase and the model parameters like weights, learning rate and other optimization parameters are updated based on the gradient flow. The training process is repeated for the given number of epochs for the convergence of the model. After the local models are trained for the given epochs, the distribution parameters and model parameters are sent to SDN controller (Server) to construct the global model. In the server side, with the received distribution parameter for each client, the data points are generated and aggregated to obtain the global dataset. The global model is built on the aggregated data with training for the given number of epochs. After the convergence of model for the given epochs based on performance metrics, The global model transmits the global model parameters (weights, learning rate, loss) to the local models in the clients.

The summarized workflow consists of the following steps:

a. The model parameters and distributer parameter are sent by the client to the server
b. The server generates the data from the distribution parameter and new global model is generated with model parameters of clients
c. The clients are updated with the global model parameters

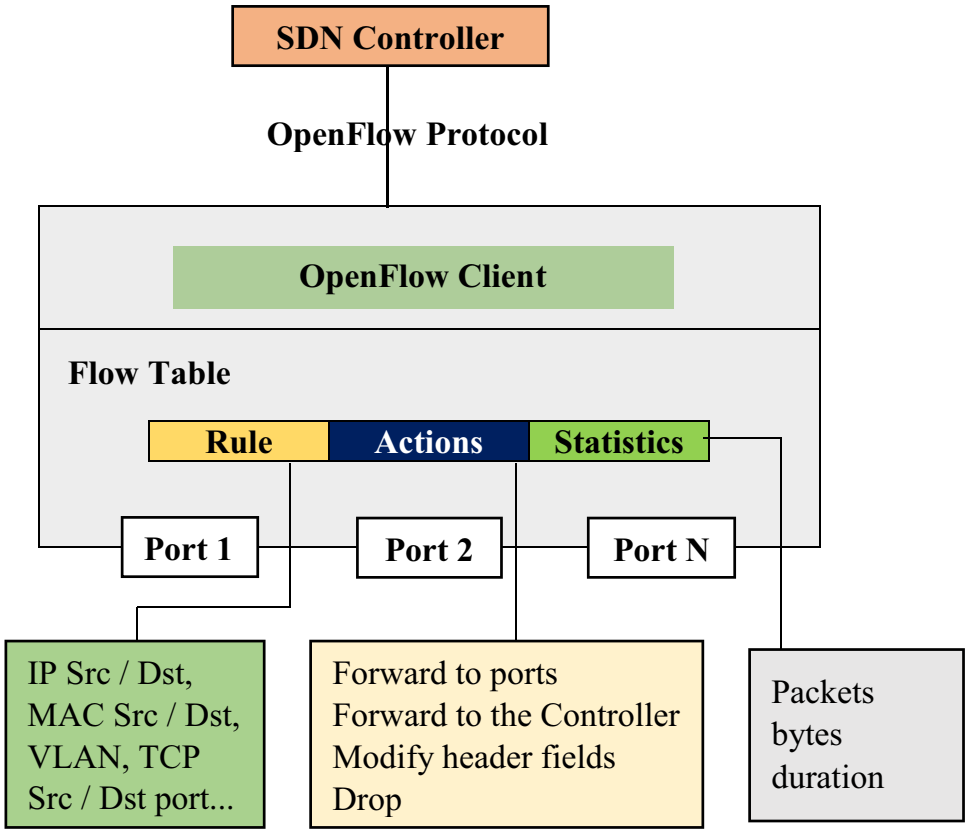
3.2 SDN controller as central server for traffic prediction with fusion Learning

Traffic prediction in SDN is crucial for network management and planning. With traffic prediction, the future traffic can be forecasted so that network performance can be improved efficiently. SDN network traffic exhibits correlation and self-similarity features and traffic prediction is a complex task. The OpenFlow protocol enables the communication between the SDN controller and the switch. The controller sends the OpenFlow messages to query the information about flow statistics. The 12-tuples of OpenFlow packet are shown in the Table 2. When a network flow arrives and no matching in the flow table, the OpenFlow switch (OF switch) forward the packet to the SDN controller through PACKET_IN message. The controller will send PACKET_MOD message with the flow survival duration for the received PACKET_IN message and send the packet by PACKET_OUT message. The overall structure of the OpenFlow switch is depicted in the Fig. 3.

Table 2 12-tuple OpenFlow Packet

In Port	Src MAC	Dst MAC	Eth Type	VLAN ID	IP ToS	IP Proto	IP Src	IP Dst	TCP Src Port	TCP Dst Port	MPLS Label

Fig. 3 OpenFlow Switch



With SDN based networking, short-term and long-term traffic prediction can be discussed. The fusion learning framework is established between the SDN controller and OpenFlow switch. When the flow table does not contain the destination for the incoming packet, it is forwarded to the controller. The local model in the switch provides the distribution of the traffic flow and provides scope for short-term traffic prediction. The fusion learning explores the insight on long-term traffic prediction. The burstiness of the flow in the network can be captured with the striking feature of SDN. The switch communicates with the controller for the missing entry of the flow table. During this communication, the updated flow distribution can be intimated to the SDN controller without additional communication overhead.

The intelligent traffic prediction framework employs the OpenFlow switch with the SDN controller in the fusion learning environment as shown in Fig. 4. The edge devices are connected with the OF switch and the switches are aggregated to the SDN controller. The SDN controller has the global topology view and graph neural network can be used for traffic prediction which exploits the spatial and temporal dependencies. Recurring entities like peak hour burst traffic conditions and unexpected events results in non-stationary time series data results in complex traffic prediction problem. Graph Convolutional Network (GCN) relates the spectral graph theory with the deep neural network. The popular GCN is ChebNet [22] with fast local convolution operates in directed and undirected graph network. In ChebNet, the features or attributes are transformed into signals, spectral domain by Eigen decomposition on the specified kernel of the convolution operation. Moving from spectral domain to vertex domain is achieved by replacement of convolution operation by diffusion process across the nodes of the graph.

3.3 Traffic Prediction

The traditional network traffic can be modeled as linear systems of polynomial function, whereas the current network traffic follows non-linear pattern. Hene, there is a need for automatic pattern extraction for the Bigdata of network traffic from diverse application in the 5G environment. The accurate prediction of network traffic is essential in 5G use-cases for NFV, network slicing and resource management and provision. In the fusion learning based traffic prediction framework, the traffic forecasting process is modeled based on extraction of spatial and temporal dependencies using graph convolutional neural network. The diffusion operation is used in the deep neural network to capture the dependencies and scheduled sampling is applied for the optimal convergence of the learning model.

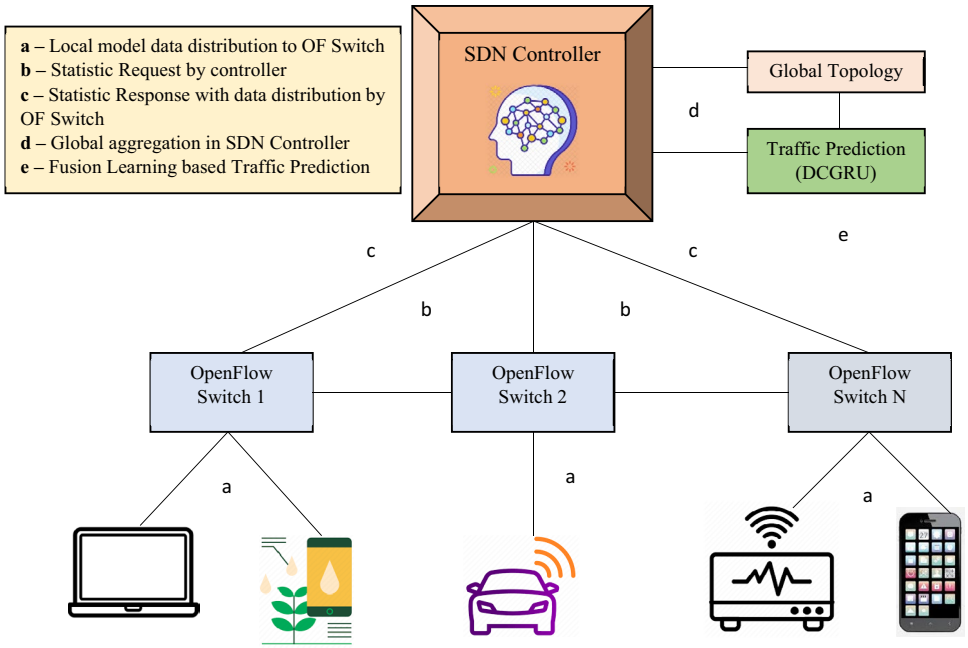


Fig. 4 Fusion Learning based Traffic Prediction Framework

3.3.1 Problem Formulation

The network traffic prediction problem can be formulated as the forecasting of the future network traffic from the previously observed time series network flow from the OF switches (N) in the SDN environment. The traffic network can be represented as directed graph, $G = (V, E, W)$. The set of open flow switches is denoted by $|V| = N$ and the link between them is given by E with the associated bandwidth and other attributes of the flow by $W \in \mathbb{R}^{N \times N}$. The traffic flow of the switch is given by graph signal as $I \in \mathbb{R}^{N \times F}$, where F denotes the feature set of the flow in the network traffic. The flow graph signal at time t is denoted as $I^{(t)}$ and the network flow prediction aims to learn the correlation function $\delta(x)$ that maps the historic time series graph T' to the future traffic signal T for the given network graph, G. It is represented as shown Eq. (2).

$$\left[I^{\left(t-T'+1\right)}, \ldots, I^{(t)}; G\right] \overset{\delta(x)}{\rightarrow} \left[I^{(t+1)}, \ldots, I^{(t+T)}\right] \tag{1}$$

3.3.2 Modeling spatial dependency

The spatial dependency of the graph network is related with diffusion process to capture the stochastic dynamics of the traffic flow in the network. The random walk in the graph G with the $\alpha \in (0, 1)$ as the restart probability. The distribution of stationary process with diffusion operation can be represented as the weighted combination of random walks in the graph G. It denoted by Eq. (3). $D = diag(W1)$ is a diagonal matric with out-degree and one vector. After many iterations, the process converges to the distribution parameter, P with the diffusion likelihood towards the bandwidth of the switch. The bidirectional diffusion can be used to capture the dependencies of upstream and downstream traffic flow in the network.

$$P = \sum_{k=0}^{\infty} \alpha(1-\alpha)^k (DW)^k \tag{3}$$

The graph signal, $I \in \mathbb{R}^{N \times F}$ with filter k_θ is diffused convoluted and the resultant operation is depicted as shown in Eq. (4). Here, θ defines the parameters of the filters of the kernel k.

$$I_p \star_d k_\theta = \sum_{i=0}^{K-1} \left(\theta_{i,1}(DW)^i + \theta_{i,2}\left(DW^T\right)^i\right) \quad I_p for p \in \{1, \ldots, P\} \tag{4}$$

The diffusion convolutional layer can be built that map U – dimension features to V – dimension outputs. It is represented as shown in Eq. (5) with tensor parameters $\Theta \in \mathbb{R}^{U \times V \times P \times 2} = [\theta]_{u,v}$. The activation function is denoted by a and it can be ReLU, Sigmoid and so on. The diffusion convolutional layer learns the structured data of the graph and trained based on stochastic gradient method.

$$O_v = a\left(\sum_{p=1}^{P} I_p \star_d k \Theta_{u,v}\right) \tag{5}$$

The overall activities of the fusion learning based traffic prediction framework is briefed with the following steps:

a. The local model embedded in the devices or the network entities find the distribution parameter for each feature of the network data. The network node and the SDN controller is interfaced by the OpenFlow protocol. The OpenFlow switch is the aggregator for the network devices to pass the distribution parameters and model parameters of the local models.
b. The SDN controller can query the OpenFlow switch using FEATURES_REQUEST and STATS_REQUEST messages.
c. The OpenFlow switch provides response back to the SDN controller using FEATURES_REPLY and STATS_REPY with the model parameters and distribution parameters along with the other network parameters (Flow ID, Number of packets, Idle timeout, priority, duration, etc.).
d. The aggregated information obtained by the SDN controller is used to construct the global model for traffic prediction. After training the model for the given number of epochs, the SDN controller broadcast the global model parameters for the local model through OpenFlow switch.
e. The Fusion learning model in the SDN controller with its global topology manager can forecast the network traffic for the given time horizon using DCGRU.

3.3.3 Modeling temporal dependency

The modeling of sequence problems with feed forward neural network has limitations of long-term dependencies that results in vanishing gradients. It is overcome by the recurrent neural network. Gated Recurrent Unit (GRU) is a variant of recurrent network trains faster than LSTM. It works only with hidden state rather than the cell state. The GRU cell consists of two gates namely Update gate and reset gate. The main work of the reset gate is to decide which part of the previous hidden information has to be combined with the current input to propose the new hidden state. The update gate determines how much of the previous hidden information has to be retained and what part of the new hidden state of reset gate is to be added with the final hidden state. The mathematical representation of GRU is shown in Eqs. (6)–(7). The r_t,u_t,c_t and h_t are reset gate, update gate, cell state and hidden state respectively. The activation functions are sigmoid (σ) and tanh. The two distinguishing features of GRU are.

- Short-term dependencies are captured by reset gates
- Long-term dependencies are captured by update gates

$$r_t = \sigma\left(W_r.x_t + U_r.h_{t-1} + b_r\right) \tag{6}$$

$$u_t = \sigma\left(W_u.x_t + U_u.h_{t-1} + b_u\right) \tag{7}$$

$$c_t = \tanh\left(W.x_t + U.\left(r_t.h_{t-1}\right) + b_c\right) \tag{8}$$

$$h_t = u_t.h_{t-1} + \left(1 - u_t\right).c_t \tag{9}$$

In Diffusion Convolutional Gated Recurrent Unit (DCGRU), the matrix multiplication is replaced by diffusion convolutional operation as shown in the Algorithm 2.1. For traffic prediction in SDN environment encoder-decoder based sequence to sequence architecture [23] with DCGRU is used for improved accuracy. During the model training, the encoder is fed with the input time series data and final state of encoder is used to initialize the decoder. The decoder provides the prediction with the ground truth value also. But during testing, the ground truth value is replaced by the model prediction value itself. There is a degradation of performance due to imbalance between the distribution of data for the training and testing of the model. To overcome this issue, random sampling mechanism is used. The scheduled sampling provides the model at the $i - th$ iteration with ground truth prediction value with the probability of ϵ_i or the model prediction value with the probability of $1 - \epsilon_i$ as shown in Fig. 5. The value of decay ϵ_i reduces gradually to 0 like stochastic gradient descent method during the training process which allows the model to learn from test data distribution. The decay can be expressed as below in Eqs. (10)–(12).

Algorithm 1.1 Fusion Learning - Local Model

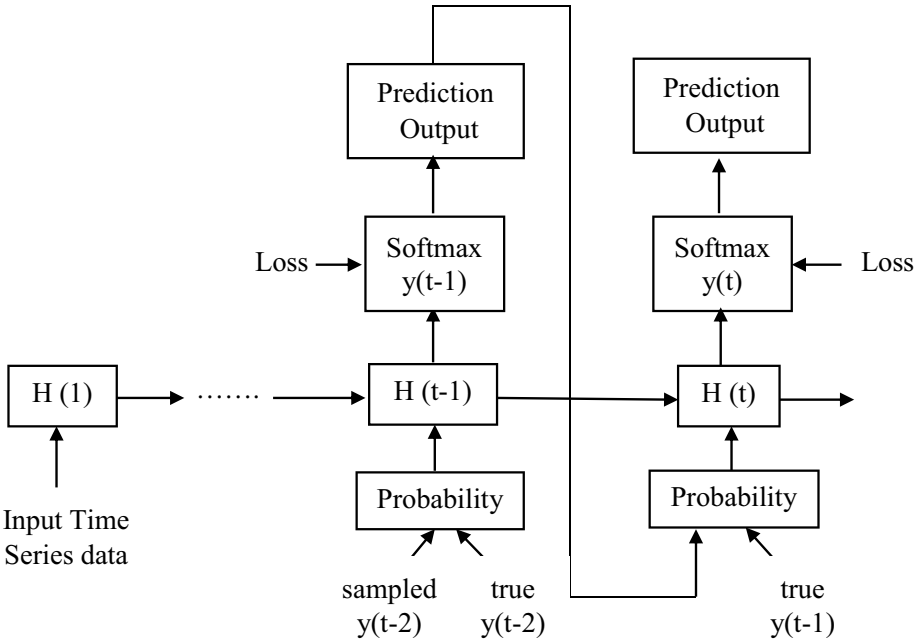
for $k \in \{set\ of\ attributes\}$ *do*
    i. Calculate the d value for each distribution using K-S test
    ii. Find the maximum value among the distributions for the attribute
    iii. Set the distribution parameter (p) for the attribute
end for
for $i \in \{1 \ldots N\}$ *epochs do*
    for $x \in \{Inputs\}$ *do*
        Update weights with $\theta = weights, \eta = learning\ rate\ and\ L_k = Loss$
        $\theta^k = \theta^k - \eta\delta L_k(\theta^k, b)$
    end for
end for
Store the updated weights (ω)
return (ω, p)

In linear decay model, based on the expected speed of convergence of the model, minimum amount of true predicted value is given to the model along with the decay slope (c) and offset (k).

$$\epsilon_i = \max(\epsilon, k - ci) \quad 0 \le \epsilon < 1 \tag{10}$$

In exponent decay model, the convergence speed of the model is dependent on the constant $(k < 1)$.

Fig. 5 Scheduled sampling



$$\epsilon_i = k^i \tag{11}$$

With $(k \geq 1)$, the inverse sigmoid decay can be expressed as

$$\epsilon_i = k/(k + exp(i/k)) \tag{12}$$

In scheduled sampling, when the model samples the previous prediction value $output_{t-1}$ during training, the gradient losses can be back propagated at time $t \rightarrow T$ through that decision. The entire DCGRU network is trained by backpropagation of maximum likelihood generation of future network traffic with exploration of spatiotemporal dependencies.

Algorithm 1.2 Fusion Learning - Global Model

for $c \in \{set\ of\ clients\ N\}\ do$

i. Generate data points from the distribution points of features of clients
ii. Find the prediction value for data points using parameters of model

end for
Merge data points to generate data set, $D = \bigcup_{i=1}^{N} D_i$
Build the traffic prediction model using the dataset, D
Return the global model parameters to the local models, client

Algorithm 2.1 Traffic Prediction

Perform Initial Fusion Learning setup – Algorithm 1.1 and Algorithm 1.2
For each OF Switch,
SDN controller query OF_FLOW_STATS for individual flow and OF_GROUP_STATS using multipart message

The OF switch returns (packet_count, byte_count, flow_count) in aggregated response with updated distribution parameter

Algorithm 1.2 – Global model
Diffusion Convolutional Gated Recurrent Unit (DCGRU)
I^t is the Input and O^t is the Output at time t
$r^{(t)}$ is the reset gate
$u^{(t)}$ is the update gate
$C^{(t)}$ is the cell gate
$\star_d$ is the diffusion convolutional operation
Parameters: $\Theta_u, \Theta_r, \Theta_C$

$$r^{(t)} = \sigma\left(\Theta_r \star_d \left[I^{(t)}, O^{(t-1)}\right] + b_r\right)$$

$$u^{(t)} = \sigma\left(\Theta_u \star_d \left[I^{(t)}, O^{(t-1)}\right] + b_u\right)$$

$$c^{(t)} = \tanh\left(\Theta_C \star_d \left[I^{(t)}, \left(r^{(t)} \odot O^{(t-1)}\right)\right] + b_c\right)$$

$$O^t = u^{(t)} \odot O^{(t-1)} + \left(1 - u^{(t)}\right) \odot c^{(t)}$$

The encoder-decoder model is used for multi-step network traffic forecasting with the recurrent units. The encoder model reads the time series data and compresses into internal representation of fixed length. The decoder interprets the internal representation of data and provides output traffic prediction value. The overall structure with diffusion convolutional operation is shown in Fig. 6.

4 Experiments and results

The experimental setup consists of the SDN environment with the diffusion convolution network with traffic prediction model. The scheduled sampling provides the improved model convergence both in local and global contexts. The Abilene topology is used in the Mininet environment and SDN flows are captured by the network monitoring tool,

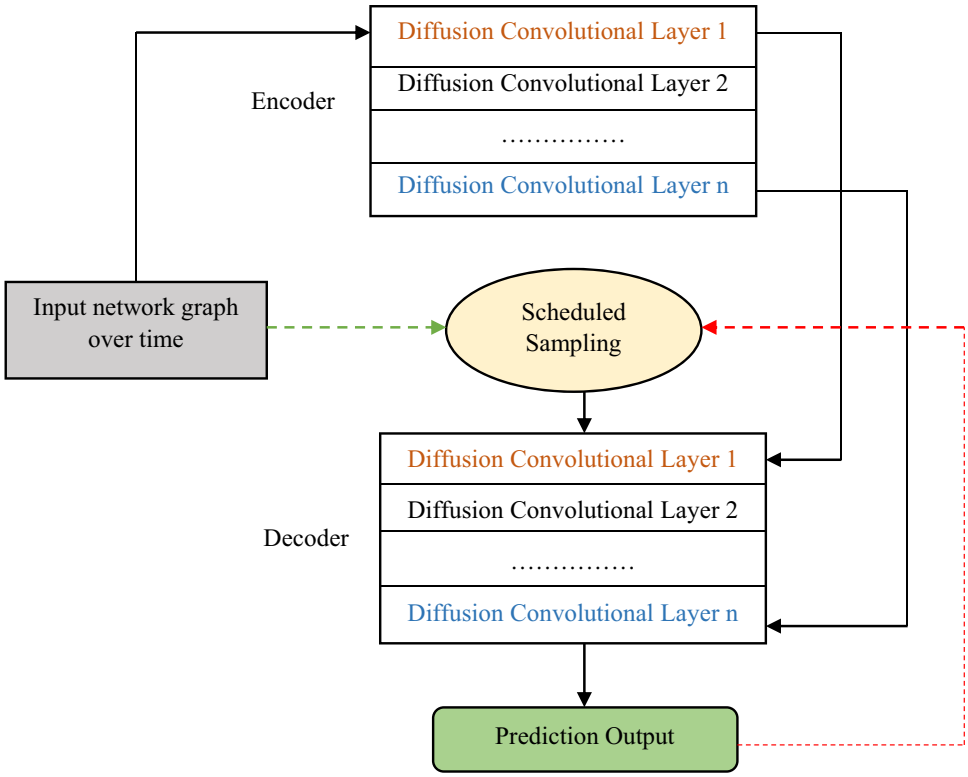


Fig. 6 Diffusion Convolutional Encoder-Decoder model

Wireshark. The captured data is preprocessed for the learning process and exploratory data analysis is performed to identify the distribution pattern of the network traffic data.

4.1 Simulation environment

The data obtained for simulation is obtained from RYU SDN controller with Abilene backbone network topology as shown in Fig. 7. Mininet is widely used SDN simulation environment and for this experiment with RYU controller and OpenvSwitch 1.4.3. The controller provides topology discovery, network visualization and network statistics measurements. With OF_FLOW_STATS message the following statistics measurements from the SDN switch. They are summarized as follow:

(i) PACKET_IN time – time of control message from the SDN switch to the controller
(ii) PACKET_OUT time – time of control message from the controller to the SDN switch

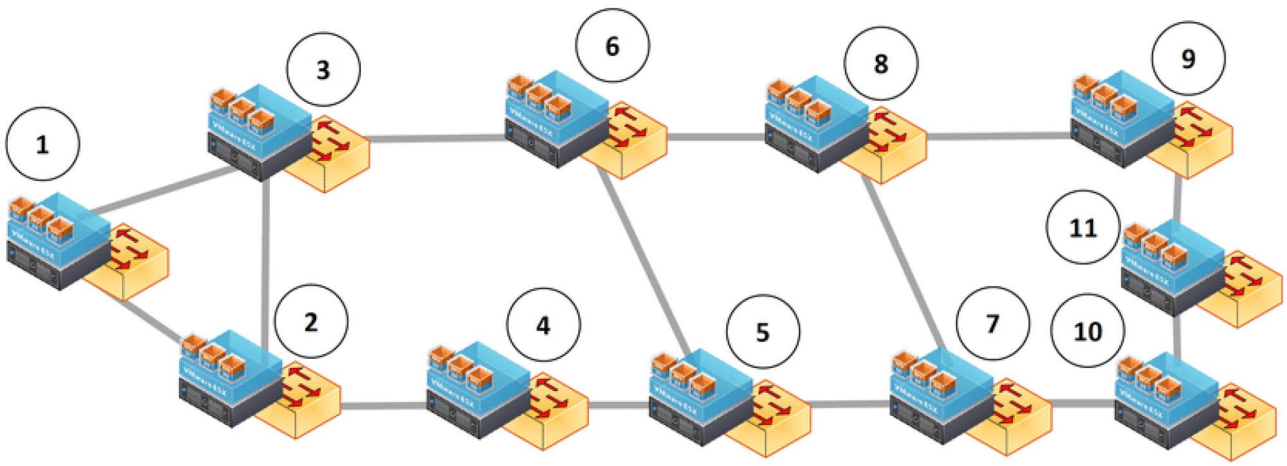


Fig. 7 Abilene Network Topology [24]

(iii) Time to process PACKET_OUT message by switch
(iv) Time taken by the packet to reach destination
(v) Number of packets per flow
(vi) Received bytes
(vii) Utilized bandwidth with meter table
(viii) Link utilization from utilized bandwidth

The dataset for this experiment is collected for every 2 min for a period of 24 h. The applications used are HTTP, DNS and FTP traffic. The SDN flow is characterized by 12-tuples and the optimal features are selected as described above. The induced burstiness in traffic is maintained in the traffic flow randomly for efficient prediction of network traffic flow. The prediction model needs random distribution of data to identify the underlying pattern in the data. The collected data is split into training data and test data. Each switch maintains its own local data and the global model resides in the controller contains aggregated data of all the switches in the topology. Each OpenvSwitch has its own features distribution although all switches share the same feature set.

4.2 Data distribution of each feature of the clients

KS test is performed on the features of the dataset to identify the data distribution of each feature for all the clients. The features and its data distribution are given in Table 3. PACKET_IN and PACKET_OUT time exhibits lognorm distribution since it will take only the positive values. The packet processing time by the controller provides normal distribution with the values routed around the mean value of the processing time. Exponential Weibull distribution provides reliability measure based on shape parameter of other distribution for transmission time of the packets. The symmetric beta distribution is rdist shown by per flow packet count. The genextreme distribution is followed by received packet for the cumulative analysis of the extreme cumulative distribution function. The dependent and independent variable analysis and data distribution is provided by bandwidth and link utilization.

Table 3 Feature data distribution

Features	Distribution
PACKET_IN time	lognorm
PACKET_OUT time	lognorm
PACKET_OUT process time	norm
Transmission time	exponweib
Packets per flow	rdist
Received bytes	genextreme
Bandwidth utilization	logistic
Link utilization	logistic

4.3 Training and test accuracy of the fusion learning model

The training accuracy of the model is same as test accuracy as the model is not trained with the original data. Based on the data distribution of each feature, the model is trained and tested for its performance evaluation and efficiency. The centralized learning provides an accuracy of 96.35%, federated learning with 97.12% and fusion learning of 97.10%. With fusion learning, the communication overheads are reduced with data distribution sharing and achieves an accuracy in.

nearer to federated learning. The accurate prediction of data distribution is the most important aspect for improved accuracy of the model. It can be observed from Fig. 8 that the accuracy of the fusion and federated learning models starts with considerable value due to the availability of the local data.

The test accuracy of all 11 clients is obtained with respect to local model and global model. The global model outperforms the local model with the model parameter and data distribution. Figure 9 showcases the test accuracy of local and global model with local data of the client and global model with model parameter and data distribution. The main advantage of the fusion learning is the improvement in communication efficiency. The federated model takes N epochs for the convergence of the global model. So, in federated learning, the total communication cost is $2 * N$ with the exchange of model parameters. But in case of fusion learning, there are two network calls, one for sending model parameters and feature distribution parameters and the global model in SDN controller returns the global model parameters to the clients.

4.4 Evaluation of traffic prediction model

Traffic prediction is based on DCGRU prediction model. It is a time series prediction model to capture spatial and temporal dependencies. The different forecasting horizons are 15, 30 min and 1 h ahead to model short-term forecasting and per day forecasting for long-term modeling. Table 4 presents the performance comparison of traffic prediction based on fully connected neural network (FNN), LSTM and DCGRU. The model is compared with the LSTM model which provides the time series prediction. The LSTM model consist of 256 LSTM units with batch size of 64. It is trained against the loss function Mean Absolute Error (MAE) with the learning rate of 0.0005. In DCGRU, the recurrent layer consists of 128 neurons, learning rate of 0.05. The above values for the parameters are chosen based on hyperparameter tuning as shown in Table 4. The evaluation of the model is based on the following metrics. The linear value is MAE that provides an equally weighted individual differences of predicted and ground truth values. It is denoted by Eq. (13). The true

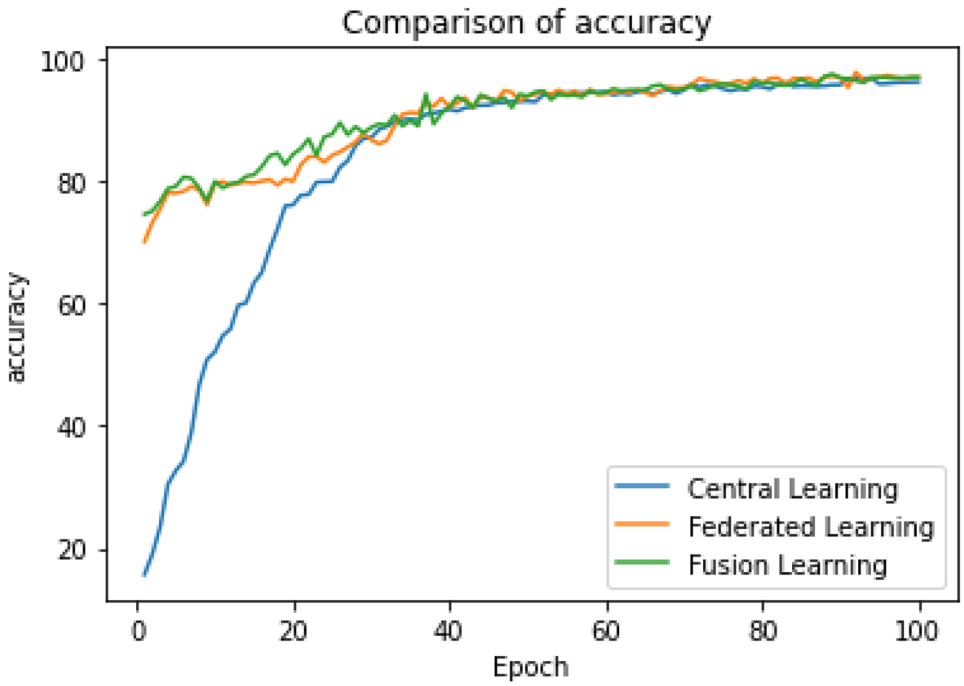


Fig. 8 Comparison of training accuracy

value is given by x and predicted value is denoted by $\hat{x}$ for n items. The lesser value of MAE denotes better model performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{13}$$

Root Mean Square Error (RMSE) is a model performance evaluation metric that provides higher weightage to the larger error components. The model performance is observed using the Eq. (14). RMSE is useful in dealing with larger error to model its impact on the learning model performance.

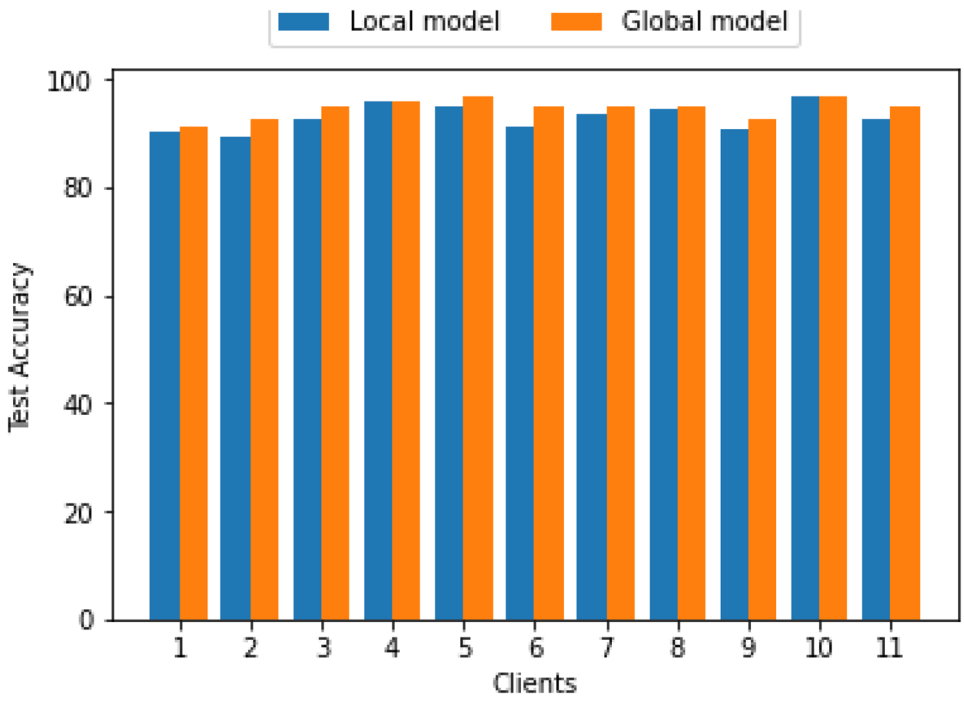


Fig. 9 Comparison of local and global model test accuracy

Table 4 Hyperparameter tuning for traffic prediction model

Hyperparameters	Values
Hidden units	64, 128, 256, 512, 1024
Batch size	0,16,32,64,128,256
Dropout	0.1,0.2,0.3,0.4,0.5
Optimizer	Adam, SGD, Adadelta, RMSProp, Nadelta, Nadam

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2} \tag{14}$$

Mean Absolute Percentage Error (MAPE) provides the measure on forecast errors as percentage and works best for the series data without extreme values. MAPE is presented in Eq. (15).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \tag{15}$$

The deep learning models are dependent on hyperparameters like number of hidden units, batch size, dropout, optimizers, learning rate and others. The prediction errors can be further minimized by hyperparameter tuning. The optimal selection of these parameters prevents the model from over-fitting and provides generalized model. The sensitivity analysis of the hyperparameters is essential for the optimal performance of the model. The results of the parameter are shown in Fig. 10.

Table 5 Performance comparison of traffic prediction models

Time	Metric	FNN	LSTM	DCGRU
15 min	MAE	4.12	3.78	**3.02**
30 min	RMSE	7.98	5.43	**5.12**
1 h	MAPE %	9.23	9.07	**7.98**

From Table 5, it is observed that DCGRU outperforms other forecasting models for network traffic prediction. Recurrent neural network can capture only temporal dependency but not spatial dependency. DCGRU provides improved performance metrics due its effectiveness in modeling spatial and temporal dependency. The increase in forecasting horizon exhibits non-linear temporal dependency which is effectively captured by deep neural models rather than linear models like historical average, etc.

To better understand the performance of the model, prediction of the traffic by the models are visualized as shown in Fig. 11. In the Figs. 11 and 12, the X-axis denotes the timestamp. For instance, 09–23 09 denotes, the date of 23–09 with time at 09 h. The following observations are made. Figure 11 visualizes the performance of the DCGRU compared with the baseline model LSTM. DCGRU is robust model aligned with prediction of variation of traffic in the network. The diffusion model provides smooth forecasting of the network traffic by both local and global model. The spatial dependency features are captured by DCGRU to predict the abrupt changes in the

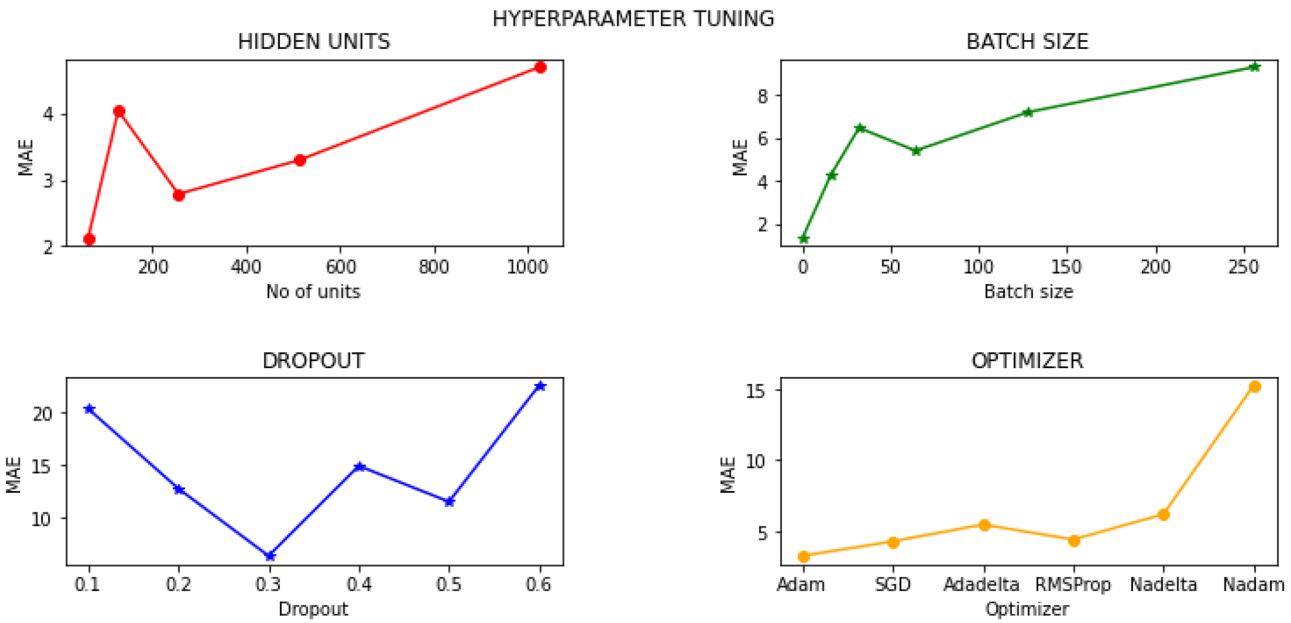


Fig. 10 Sensitivity analysis of hyperparameters

Fig. 11 Traffic prediction for 24 h

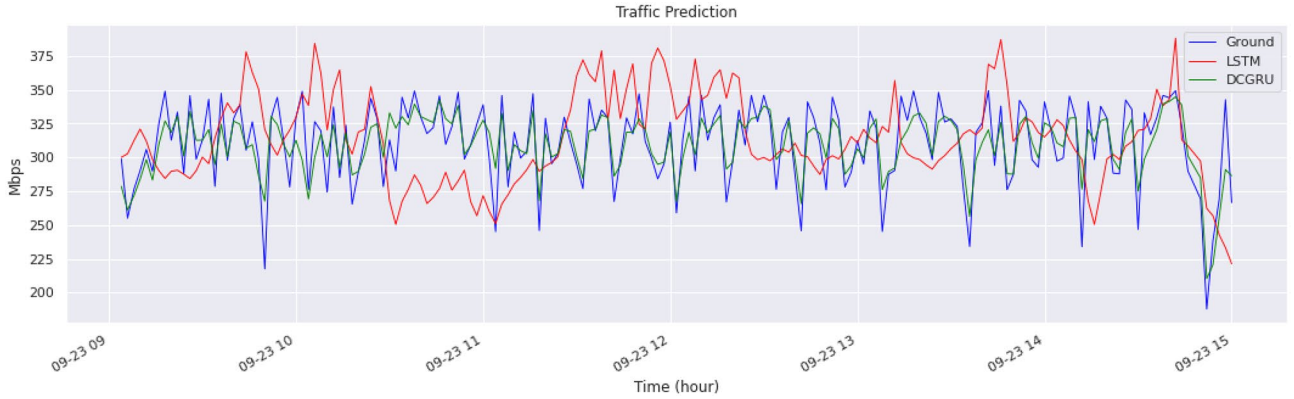


Fig. 12 DCGRU smooth prediction of traffic burstiness

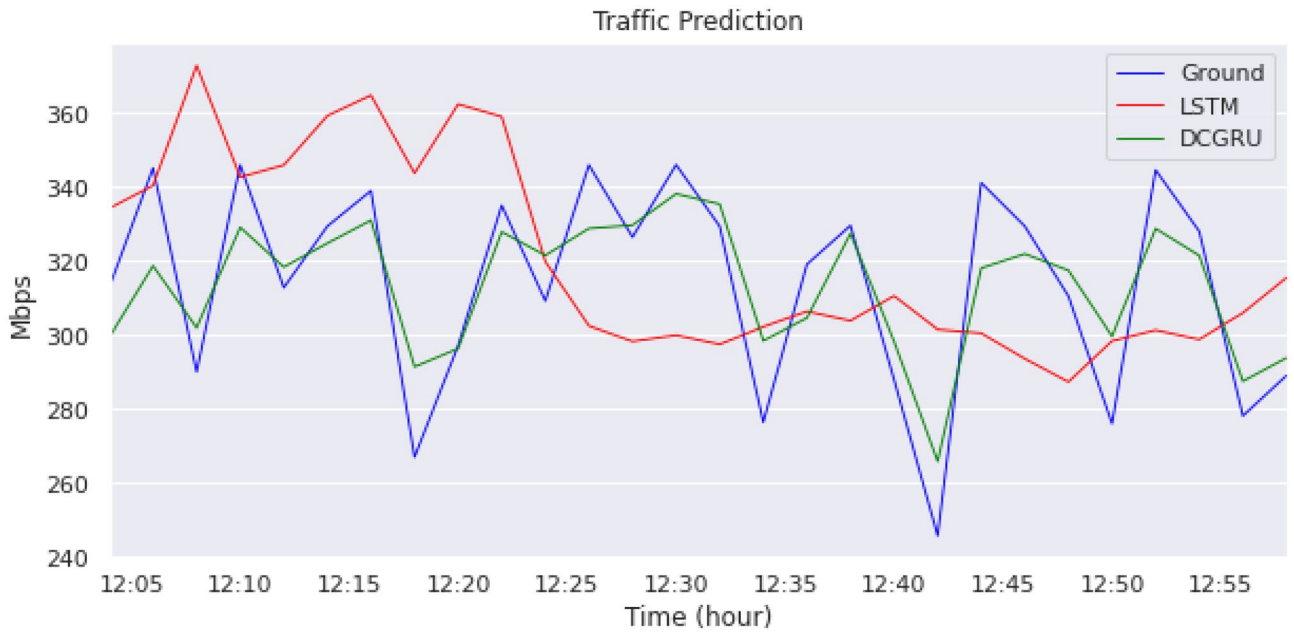


Fig. 13 Behavior of DCGRU in 1 h with more traffic variation pattern

network traffic as depicted in Fig. 12. The reason behind the smooth prediction is the usage of the change in the network traffic pattern captured by diffusion process. The SDN controller with the prediction model enables the traffic forecasting for dynamic resource provisioning, automatic network slicing and so on. Figure 13 picturizes the traffic forecasting for one hour. The DCGRU model is aligned with the ground value because the diffusion process localizes the weights and weight diffusion along the participating clients.

5 Conclusion

Traffic prediction plays a crucial role in network performance analysis and network planning. SDN enables the prediction of network traffic through global network topology manager and traffic prediction algorithm in the controller. Fusion learning provides distributed learning with minimal communication overhead between the SDN clients and the controller. The spatial dependencies of the features are captured by diffusion convolutional operations and temporal dependencies are explored by gated recurrent units. Scheduled sampling allows the passing of the data to next process of learning with the probability for improved accuracy of the model. The framework is tested with the simulation data for the traffic prediction in baseline network topology. The model exhibits slower prediction error compared to the baseline models like LSTM. It is capable of capturing the sudden burstiness in the traffic pattern with diffusion process.

The model can be further integrated with dynamic resourcing provisioning in the edge nodes of the wireless network. Network function virtualization with SDN provides the agile and dynamic environment in 5G network for resource provisioning and differentiated customer-oriented services on the fly as the service.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Popovski P, Trillingsgaard KF, Simeone O, Durisi G (2018) 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. IEEE Access 6:55765–55779
2. Santos GL, Endo PT, Sadok D, Kelner J (2020) When 5G meets deep learning: a systematic review. Algorithms 13(9):208
3. Niknam S, Dhillon HS, Reed JH (2020) Federated learning for wireless communications: Motivation, opportunities, and challenges. IEEE Commun Mag 58(6):46–51
4. Huang Y, Qian L, Feng A, Yu N, Wu Y (2019) Short-term traffic prediction by two-level data driven model in 5G-enabled edge computing networks. IEEE Access 7:123981–123991
5. Subramanya T, Riggio R (2021) Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond. IEEE Trans Netw Serv Manage 18(1):63–78
6. Qi Y, Hossain MS, Nie J, Li X (2021) Privacy-preserving Blockchain-based federated learning for traffic flow prediction. Futur Gener Comput Syst 117:328–337
7. Kasturi A, Ellore AR, Hota C (2020) Fusion learning: A one shot federated learning. In: International Conference on Computational Science, Springer. pp. 424– 436
8. Zhou Y, Pu G, Ma X, Li X, Wu D (2020) Distilled one-shot federated learning. arXiv preprint arXiv: 2009.07999
9. Zhang J, Zheng Y, Sun J, Qi D (2019) Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Trans Knowl Data Eng 32(3):468–478
10. Wang S, Zhuo Q, Yan H, Li Q, Qi Y (2019) A network traffic prediction method based on LSTM. ZTE Communications 17(2):19–25
11. Wang Y, Jiang D, Huo L, Zhao Y (2021) A new traffic prediction algorithm to software defined networking. Mobile Networks and Applications 26(2):716–725
12. Xiang L, Ge X-H, Liu C, Shu L, Wang C-X (2010) A new hybrid network traffic prediction method. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pp. 1–5
13. Zhou J, Wang H, Xiao F, Yan X, Sun L (2020) Network traffic prediction method based on echo state network with adaptive reservoir. Software: Practice and Experience
14. Wang W, Bai Y, Yu C, Gu Y, Feng P, Wang X, Wang R (2018) A network traffic flow prediction with deep learning approach for large-scale metropolitan area network. In: NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9
15. Iqbal MF, Zahid M, Habib D, John LK (2019) Efficient prediction of network traffic for real-time applications. Journal of Computer Networks and Communications
16. Li M, Wang Y, Wang Z, Zheng H (2020) A deep learning method based on an attention mechanism for wireless network traffic prediction. Ad Hoc Networks 107:102258
17. Yu B, Yin H, Zhu Z (2019) St-unet: A spatio-temporal u-network for graph structured time series modeling. arXiv preprint arXiv: 1903.05631
18. Yu B, Li M, Zhang J, Zhu Z (2019) 3d graph convolutional networks with temporal graphs: A spatial information free framework for traffic forecasting. arXiv preprint arXiv: 1903.00919
19. Li Y, Yu R, Shahabi C, Liu Y (2017) Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv: 1707.01926
20. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv: 1709.04875
21. Liu Y, James J, Kang J, Niyato D, Zhang S (2020) Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet Things J 7(8):7751–7763
22. Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. Computational Social Networks 6(1):1–23
23. Bengio S, Vinyals O, Jaitly N, Shazeer N (2015) Scheduled sampling for sequence prediction with recurrent neural networks. arXiv preprint arXiv: 1506.03099
24. Bu T, Duffield N, Presti FL, Towsley D (2002) Network tomography on general topologies. ACM SIGMETRICS Performance Evaluation Review 30(1):21–30

K. Tamil Selvi is an Assistant Professor in the Department of CSE, Kongu Engineering College, Erode, Tamil Nadu. She is pursuing Ph.D in the area of Traffic Engineering in Software Defined Networking (SDN). Her other areas of interest are Machine Learning, Deep Learning and Networking. She has published more than 15 papers in international journals and conferences.

R. Thamilselvan is a Professor in the Department of Information Technology, Kongu Engineering College, Erode, Tamil Nadu, India. He has completed his M.E Computer Science and Engineering in 2005 and PhD in Computer Science and Engineering in 2013 under Anna University Chennai. He has completed 20 years of teaching experience. He has published 33 papers in International Journal, 10 papers in International Conference and 15 papers in National Conference. He has completed one research project sponsored by AICTE, New Delhi under the scheme Research Promotion Scheme (RPS) and organised 2 national level seminar and 1 faculty development programme sponsored by AICTE, New Delhi. His area of interest includes Grid and Cloud Computing, Parallel Processing, Big Data Analytics and Distributed Computing.