

## Research Article

# TSN Switch Queue Length Prediction Based on an Improved LSTM Network

Xin Wang,<sup>1,2,3,4</sup> Zhijun Shang<sup>1,2,3</sup>, Changqing Xia,<sup>1,2,3</sup> Shijie Cui,<sup>1,2,3</sup> and Shuai Shao<sup>1,2,3</sup>

<sup>1</sup>Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China

<sup>2</sup>Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

<sup>3</sup>Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China

<sup>4</sup>University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence should be addressed to Zhijun Shang; shangzj@sia.cn

Received 9 October 2021; Revised 9 November 2021; Accepted 11 November 2021; Published 22 December 2021

Academic Editor: Deepak Kumar Jain

Copyright © 2021 Xin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the high-speed development of network technology, time-sensitive networks (TSNs) are experiencing a phase of significant traffic growth. At the same time, they have to ensure that highly critical time-sensitive information can be transmitted in a timely and accurate manner. In the future, TSNs will have to further improve network throughput to meet the increasing traffic demand based on the guaranteed transmission delay. Therefore, an efficient route scheduling scheme is necessary to achieve network load balance and improve network throughput. A time-sensitive software-defined network (TSSDN) can address the highly distributed industrial Internet network infrastructure, which cannot be accomplished by traditional industrial communication technologies, and it can achieve distributed intelligent dynamic route scheduling of the network through global network monitoring. The prerequisite for intelligent dynamic scheduling is that the queue length of future switches can be accurately predicted so that dynamic route planning for flow can be performed based on the prediction results. To address the queue length prediction problem, we propose a TSN switch queue length prediction model based on the TSSDN architecture. The prediction process has three steps: network topology dimension reduction, feature selection, and training prediction. The principal component analysis (PCA) algorithm is used to reduce the dimensionality of the network topology to eliminate unnecessary redundancy and overlap of relevant information. Feature selection requires comprehensive consideration of the influencing factors that affect the switch queue length, such as time and network topology. The training prediction is performed with the help of our enhanced long short-term memory (LSTM) network. The input-output structure of the network is changed based on the extracted features to improve the prediction accuracy, thus predicting the network congestion caused by bursty traffic. Finally, the results of the simulation demonstrate that our proposed TSN switch queue length prediction model based on the improved LSTM network algorithm doubles the prediction accuracy compared to the original model because it considers more influencing factors as features in the neural network for training and learning.

## 1. Introduction

With the explosion of traffic and the continuous emergence of new services on the Internet, existing network technologies have become insufficient for meeting the requirements for real-time performance and reliability in application scenarios such as the industrial Internet, 5G, augmented reality/virtual reality (AR/VR), holographic communication, smart grids, telematics, smart cities, and telemedicine [1]. At the same time, these are also promising application sce-

narios for future wireless time-sensitive networks (TSNs). These applications will drive the rapid development of digital video services and cause a dramatic increase in data flow in TSN packet-switched networks in the future [2]. Due to the burstiness of packet traffic, the traditional routing and scheduling algorithms in TSNs cannot effectively avoid congestion in the network core switches. Therefore, it is necessary to propose a TSN distributed dynamic scheduling algorithm based on the time-sensitive software-defined network (TSSDN) model to achieve load balance in TSNs by

planning the routing of data streams dynamically in real time with a TSSDN controller to optimize the network transmission delay and improve network throughput.

Existing TSN routes are all offline routes and cannot be distributed or scheduled dynamically. Traditional link-based routing algorithms also require TSN switches to inquire about the entire network topology, in contrast to distributed routing algorithms that rely on all neighboring nodes to find the best path to a destination iteratively [3].

Currently, the main solution to the TSN scheduling problem is IEEE 802.1Qbv. Based on Qbv, Zhao proposed a genetic simulated annealing and ant colony system-based TSN scheduling algorithm [4], which guarantees deterministic network delay and jitter by optimizing the gating list inside the switch. However, it is not explained whether the desired results are still available when facing bursty traffic. Therefore, scheduling optimization can also be performed from a routing perspective to avoid the network congestion caused by traffic bursts through load balancing. Ojewale et al. introduced the heuristic LB-DRR and CR-DRR algorithms [5] to achieve multipath selection during network congestion by computing multiple disjoint paths, which in turn enables network load balancing to improve network throughput. However, conventional heuristic algorithms still face the speed of slow convergence when tackling large-scale problems, and the feasible scheduling of one constraint may be unavailable for other constraints due to the diversity of constraints, namely, poor scalability. Based on the above problems, this paper proposes a TSN switch queue prediction model based on an improved long short-term memory neural network to provide distributed intelligent dynamic routing for the network by providing a reasonable prediction of the length of the next time slot switch queue. This routing algorithm based on switch queue prediction is mainly predicted by a neural network prediction model and based on the TSSDN unified control plane for routing decisions to be issued. The key contributions of this article are as follows.

- (1) Topology feature learning: to eliminate the redundancy of the network topology, we propose extracting the topology features by a principal component analysis (PCA) algorithm. This can effectively indicate the connectivity between switches in a TSN. With the PCA algorithm, we can gain a reduced-dimensional node score vector matrix of the network topology
- (2) Improved long short-term memory (LSTM) network prediction: considering the burstiness of traffic, we combine the TSN topology, traffic patterns, and TSN switch queue lengths. We propose an improved LSTM network to predict the queue length of the next time slot TSN switch by incorporating network topology features to design an LSTM network structure with unequal input and output lengths, and the prediction results are used as a measure for intelligent routing decisions. The drawbacks of the slow convergence and poor scalability of heuristic algorithms in dealing with large-scale network problems are eliminated

## 2. Related Works

*2.1. TSN.* Initially, the Ethernet specification ignored the issue of real-time communication. The development of an efficient and economical traffic control scheme is essential to enable real-time communication on a specified network. The IEEE working group developed a communication protocol for time-sensitive networks to support accurate and real-time transmission over the network. This mechanism allows considering the traffic transmission problem from two perspectives: (1) scheduling (i.e., when to transmit traffic) and (2) routing (i.e., on which path to transmit traffic).

*2.2. TSN Scheduling Algorithms.* TSNs mostly use heuristic algorithms for network scheduling to guarantee deterministic delays in transmission. Kentis et al. introduced a heuristic scheduling algorithm for multipath TSNs considering gated congestion [6], which can reduce the gating time by 26%. However, most of the traditional TSN scheduling algorithms appear to use static network route scheduling, which do not achieve good network scheduling and cause network congestion when there are traffic bursts in the network. Currently, Nayak et al. [7] emphasize the importance of routing to reduce the number of transmission operations by avoiding unreasonable routing policies, thus alleviating traffic congestion.

*2.3. Routing Algorithms.* Currently, the most commonly used routing algorithm is the Dijkstra shortest path algorithm, which is a centralized routing algorithm that requires the controller to collect global network topology information and then perform the routing. Distributed routing algorithms, such as Bellman-Ford (BF), rely on the controller to search for the shortest path and therefore do not require the global network topology. However, the dramatic growth of future data traffic poses significant challenges for such algorithms, such as CPU performance of controllers, low latency requirements, and buffer capacity. As a result, many adaptive heuristic routing algorithms have emerged.

Adaptive routing algorithms are aimed at balancing other unbalanced (skewed) traffic patterns [8]. Universal globally aware load balancing (UGAL) determines the congestion level of each packet by estimating the buffer occupancy of neighboring routers. This network-wide aware load balancing idea provides us with an option to solve the problem of large-scale traffic bursts in TSNs. We also need a step-aware prediction of switch queue length in TSNs so that when a switch queue is predicted to be about to be congested, the routing policy can be changed in time to bypass the congestion and reduce the waiting delay. Therefore, it is critical to develop an intelligent real-time routing scheme for TSNs with bursty heavy traffic.

*2.4. TSSDN.* TSN [9] is a family of protocols that achieve deterministic minimum delay in a nondeterministic Ethernet; software-defined networking (SDN) is a new network architecture. It enables the separation of the data plane from the control plane and the programmability of control [10]. It can monitor the global network state and implement management [11]. Although TSN achieves deterministic

transmission, it can only achieve static routing and cannot reasonably perform dynamic route planning for traffic when there is bursty traffic in the network, which causes network congestion. An SDN can achieve global network monitoring, dynamic route planning, and programmable control, but it cannot guarantee deterministic transmission delay. Therefore, it is essential to integrate the advantages of these two networks.

The TSSDN was first proposed by Nayak et al. [12], and the convergence of the SDN and TSN constitutes the TSSDN, which uses the same network abstraction layer and has two main implementations [13], namely, “TSSDN Gateway” and “TSSDN Unified.” “TSSDN Gateway” refers to the interconnection of the TSN and SDN domains through protocol conversion; “TSSDN Unified” means that TSN protocols and SDN functions are implemented directly on the same device.

We choose “TSSDN Unified” for switch queue prediction, which meets the following two requirements: (1) the separation overhead of the data plane and control plane of the SDN does not affect the effectiveness of TSN protocol implementation; (2) the configuration update and distribution ensure real-time performance and consistency. Therefore, the traffic processing flow inside the TSSDN switch contains a series of processes, from incoming port filtering and lookup forwarding to outgoing port queue gating shaping, frame preemption, and physical layer transmission. The process is shown in Figure 1.

To ensure deterministic delay and jitter requirements for time-sensitive flows, the following four protocols are required: (1) network-wide device clock synchronization, (2) end-to-end bandwidth allocation and resource reservation for flows, (3) filtering for incoming port traffic, and (4) gated queue scheduling shaping for outgoing port traffic. These four protocols are the main protocols of TSN.

**2.5. Neural Network Prediction Model.** Neural networks are mathematical models that simulate the structure of neuronal connections in the human brain for information processing, which can continuously adjust the relationships between internal nodes and learn changes in data based on training data and have the ability of self-learning, adaptive, and non-linear approximation. The artificial neural network (ANN) has an active role in improving the decision-making in network operations [14]. Recurrent neural network (RNN) solves the information preservation problem due to its special network model structure, which not only learns the information of the current moment but also relies on the information of the previous sequence, allowing information persistence. However, the output of the next moment of the RNN is only influenced by the output of the previous moment, and therefore, it cannot solve the long-term learning problem. In response, Hochreiter and Schmidhuber [15] proposed the LSTM network, which is an improvement of RNN to achieve learning long-term dependent information by adding cell states to the network. However, the LSTM also only achieves the prediction of temporal data and cannot consider many influencing factors, such as the effect of network topology on the switch queue length. Therefore,

we achieve further optimization of switch queue length prediction by improving the network structure of LSTM. Recently, there have been great breakthroughs in the research of deep learning systems and their applications in fields such as pattern recognition, based on which we introduce an improved LSTM network scheme to implement the prediction of TSN switch queue length as the basis for distributed intelligent routing in TSNs.

Motivated by TSSDNs and neural networks, we propose a TSN switch queue length prediction model based on an improved LSTM network structure under the TSSDN architecture to achieve real-time prediction of switch queues for dynamic route planning.

### 3. TSSDN Model and Topology Feature Extraction

**3.1. Flow Monitoring on the Data Plane.** TSSDN unification [16] is aimed at providing a unified control plane to create a combined network consisting of TSN bridges and SDN switches without any changes to the data plane. Due to the flexible configuration capability, both non-time-sensitive and time-sensitive connections can be set up on demand.

As shown in Figure 1, a TSSDN is mainly composed of three planes: the management plane, the control plane, and the data plane. We rely heavily on the control plane and the data plane to accomplish predictive routing. In the control plane, the TSSDN controller monitors the global network real-time status (i.e., the real-time length of each queue for each switch). When a packet arrives at a TSN switch in the data plane, the TSSDN controller detects the queue lengths and traffic patterns of all the underlying TSN switches and collects this information into the TSSDN controller for processing. The TSSDN controller performs the prediction of the queue length of that switch for the next time slot based on our proposed improved LSTM framework. Based on the prediction result, the switch queue length vector is assigned, the next-hop address is intelligently selected for the packet, and finally, the next-hop address is sent down to the switch in the data plane in the form of a flow table so that dynamic intelligent routing can be achieved. The specific process is shown in Figure 2.

**3.2. Switch Queue Length Prediction for Routing.** We abstract the network topology model by  $G = (V, E)$ , where  $n = |V|$  is the number of fixed points;  $m = |E|$  is the number of edges. Since the shortest path algorithm only considers the path length and ignores the queue length change of the switch during transmission, this will cause the traffic to be stalled in the switch buffer, which in turn causes traffic transmission delay. If the traffic can dynamically choose the transmission path based on the real-time queue length of the switch, then the traffic can be load balanced throughout the network. The queue prediction-based routing scheme considers both the current and predicted switch queue lengths. When the current switch is not the destination, the queue prediction-based routing scheme determines the next hop of the packet in the buffer. When a TSN switch receives a packet, the TSSDN controller intelligently selects the next hop based

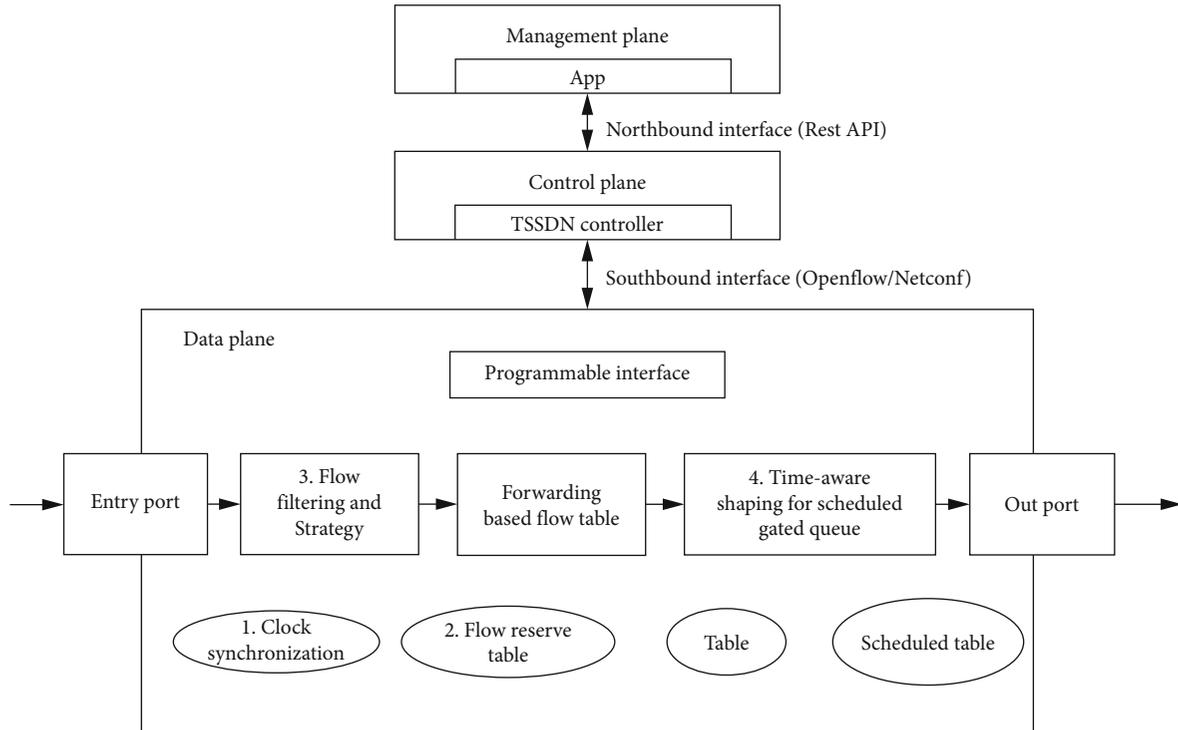


FIGURE 1: TSSDN architecture.

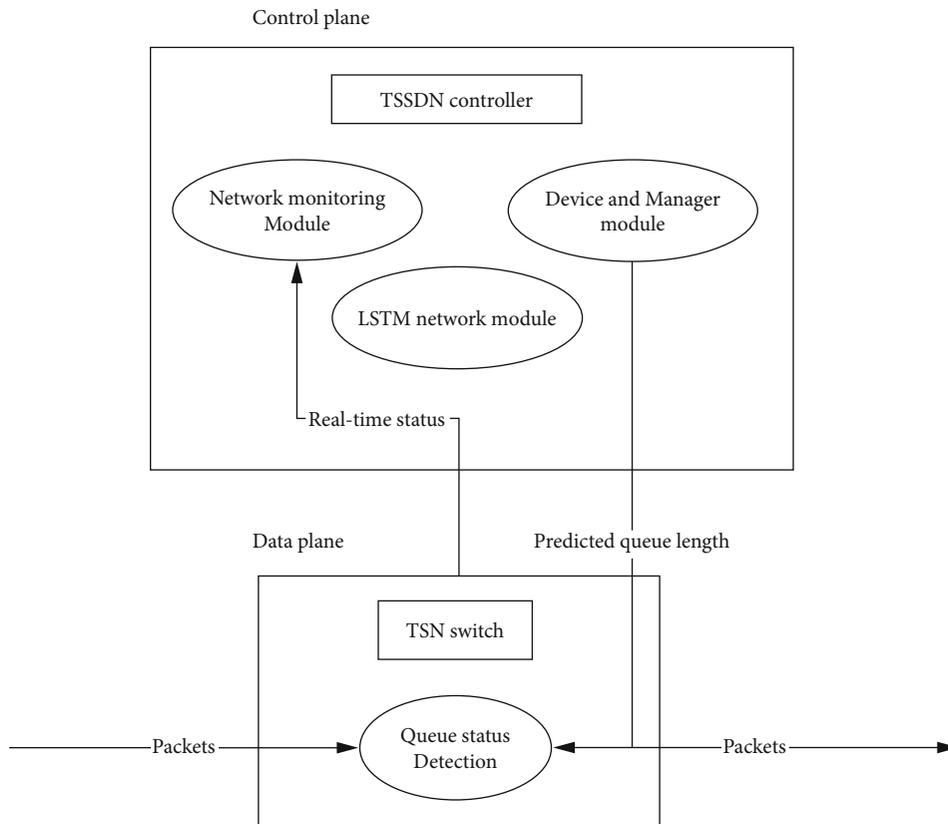


FIGURE 2: Hierarchical design of the control and data plane.

on multiple metrics, such as the switch queue length and topology. Unlike the shortest path algorithm, the queue prediction-based routing scheme does not select the shortest path for routing but dynamically and intelligently selects the next hop to avoid packet loss and long wait delays in the TSN switch buffer.

**3.3. PCA-Based Feature Extraction.** In queue prediction-based routing schemes, the connectivity representation of the switches in the underlying network is critical for the neural network to learn the characteristics of the network topology. In order to be able to represent more easily and accurately the connectivity of the switch in the network topology, we can use PCA algorithm to extract the switch connectivity. The switch node score in the network represents the connectivity of the switch nodes, which can represent the switch connectivity instead of the degree of the switch nodes.

The principle of the PCA [17] algorithm is linear mapping, and in data analysis, it means that the principal components of the data (containing informative dimensions) are retained and the components that are not important for data description are ignored. Based on this, PCA can be performed on the adjacency matrix to gain basic information about the TSN topological connections. PCA is performed on the adjacency matrix, and the scores of the analysed principal components are used to represent the topological information of the network, which achieves matrix dimensionality reduction and uses it as an input variable for prediction.

The connection information of the TSN switch is represented by the matrix  $X$ .

$$X_{N \times N} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{bmatrix}. \quad (1)$$

When there is a link between switches  $i$  and  $j$ ,  $x_{ij} = 1$ ; otherwise,  $x_{ij} = 0$ .  $X_i$  reflects the connection of switch  $i$  to other switches. Next, the covariance matrix is calculated as

$$\sum_{N \times N} = \frac{1}{N-1} X^* (X^*)^T. \quad (2)$$

According to the definition of PCA, each sample will have a corresponding principal component score, which can be expressed as follows:

$$\begin{cases} P_1 = l_1^T X = l_{11}X_1 + l_{12}X_2 + \cdots + l_{1N}X_N, \\ P_2 = l_2^T X = l_{21}X_1 + l_{22}X_2 + \cdots + l_{2N}X_N, \\ \vdots \\ P_N = l_N^T X = l_{N1}X_1 + l_{N2}X_2 + \cdots + l_{NN}X_N. \end{cases} \quad (3)$$

The principal component score of each TSN switch is represented as the connectivity of the switch, which can

accomplish the dimension reduction of the network adjacency matrix well, and the score can also reflect the connectivity relationship of the switch in the substrate network well.

Since the variance reflects the degree of deviation of the data from its mean, it is assumed that the larger the variance, the more informative the data are. The variances are arranged in the order from maximum to minimum, and the one with the highest variance is the first principal component, and so on. And each principal component should be guaranteed to be orthogonal to each other. The PCA algorithm is as follows.

## 4. TSN Switch Queue Prediction Model Design

This section discusses the prediction of TSN switch queue length based on neural networks. The prediction process can be separated into initialization, training, and operation phases. Through accurate prediction, the ideal output value of the length of each queue of the TSN switch at the next time slot is obtained.

**4.1. Input and Output.** To describe the research problem, we designed a network consisting of TSN switches. In the network, we select routing paths for packets based on their current location and destination in the packet-switched network. In order to be able to give an accurate next-hop selection, we first need to make a reasonable prediction of the queue length of the TSN switches. If the TSN switch is predicted to be congested at the next time slot, then the TSSDN controller can dynamically reselect the next hop address to avoid congestion.

We selected a time series-based neural network and included features of the network topology for TSN switch queue length prediction.

- (1) Topology:  $P^{<t>} = [p^{<1>}, p^{<2>}, \dots, p^{<N>}]$ ,  $p^{<i>}$  indicates the score of the switch in the network after the PCA algorithm has been downscaled. The score indicates the connectivity of the switch and  $N$  indicates the number of switches in the network. The higher the switch node score, the wider the connectivity of the switch
- (2) Traffic types: there are three traffic types in TSN, TT, AVB, and BE; each of which has a different priority, and traffic of different priority enters different queues in the switch. Therefore,  $T_m^{<t>} = [t_m^{<1>}, t_m^{<2>}, \dots, t_m^{<t-x>}]$  denotes the number of packets arriving at the TSN switch in the previous time interval.  $t_m^{<t>}$  is the number of packets in queue  $m$ 's buffer of the switch at moment  $i$
- (3) Queue length:  $Q_m^{<t>} = [q_m^{<1>}, q_m^{<2>}, \dots, q_m^{<t-x>}]$  indicates the current queue state in the network switch buffer, and  $q_m^{<i>}$  is the number of normal packets in switch queue  $m$  at moment  $i$ . Each TSN switch has eight priority queues, so the queue length of each switch is considered separately

Input:  $X_{N \times N} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{bmatrix}$  //sample sets

Output:  $\begin{cases} P_1 = l_1^T X = l_{11}X_1 + l_{12}X_2 + \cdots + l_{1N}X_N, \\ P_2 = l_2^T X = l_{21}X_1 + l_{22}X_2 + \cdots + l_{2N}X_N, \\ \vdots \\ P_N = l_N^T X = l_{N1}X_1 + l_{N2}X_2 + \cdots + l_{NN}X_N. \end{cases}$

- 1:  $\bar{x}_j = \sum_{i=1}^N x_{ij}/N$ ; //sample means
- 2:  $\sigma_j^2 = \sum_{i=1}^N (x_{ij} - \bar{x}_j)^2/N$ ; //sample variance
- 3: for  $j$  from 1 to  $N$  do
- 4: for  $i$  from 1 to  $N$  do
- 5:  $x_{ij}^* = (x_{ij} - \bar{x}_j)/\sigma_j$ ;
- 6: end for
- 7: end for
- 8:  $\sum_{N \times N} = (1/(N-1))X^*(X^*)^T$  //calculate the covariance matrix of the sample
- 9:  $\sum_{N \times N} l_i = \lambda_i l_i$  //eigenvalue decomposition of the covariance matrix
- 10:  $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$  //the first  $M$  maximum eigenvalues are taken and normalized to the contribution rate
- 11:  $l_1, l_2, \dots, l_M$  //unit eigenvectors
- 12:  $P_{1 \times N} = P_{N \times 1}^T = l_1^T X$  //the individual principal component scores of each evaluation object, i.e., the sum of the scores of the dimensionality-reduced dataset

ALGORITHM 1: PCA: network topology downscaling.

**4.2. Initialization Phase.** In the initialization phase, we should train the parameters of our proposed neural network based on the dataset. The topology, traffic pattern, and queue length state are used as inputs, and the neural network should process the TSN switch queue length at the next time slot. The queue length in the switch buffer is used to represent the utilization of the switch queue. In addition, since the traffic type also contains the number of packets arriving in the switch buffer, it should be added to the queue length for queue length updates. Because there are eight priority queues in the TSN switch, we can use a time series of eight switch queue lengths to represent the inputs and outputs of the prediction model. However, it is important to note that the sequence length of the input is larger than the sequence length of the output because the effect of the network topology on the switch queue length in the network is considered in the input. The mathematical expression of the input and output in the neural network is as follows:

$$X_m^{<t+N>} = [P_m^{<N>}, Q_m^{<t>} + T_m^{<t>}], \quad (4)$$

$$Y_m^{<t>} = [Q_m^{<t>}], \quad (5)$$

where  $X$  denotes the input value of the prediction model,  $Y$  denotes the output value of the prediction model,  $m$  denotes the number of queues of switches in the whole network, and  $N$  denotes the number of switches in the network. Each switch has eight priority queues, so there are  $8N$  priority queues in the whole network that need to be predicted. Therefore,  $m = 8N$ .

**4.2.1. Training Model.** A recurrent neural network (RNN) [18] is a neural network used to process serial data whose output is associated with the historical input at each moment. It has the same prediction principle as BP neural networks, with forward propagation to produce predictions and backward propagation for parameter optimization. However, it has only short-term memory and cannot predict data accurately in the long term in time series data prediction.

LSTM networks [19] use a special gate structure to cleverly combine short-term memory and long-term memory to learn long-term dependencies by addressing, to some extent, the problem of gradient disappearance that exists in RNNs. However, the structure of LSTM networks is mostly a pattern with equal input and output. Even if there are cases of unequal input-output lengths, the network waits for all the inputs to finish before determining the outputs, e.g., in Chinese-English translation. This is not consistent with the real situation of our real-time network traffic prediction problem. If the input and output lengths are equal, only the influence of the time factor is considered, ignoring many other influencing factors of flow; if we wait for all the inputs before outputting, it will increase the prediction time of network traffic, thus increasing the network latency. Therefore, we design a network structure with unequal input and output lengths by considering the network topology features to address the above problem of network traffic prediction that does not match the real situation. The model structure is shown in Figure 3.

In this model, we consider the effect of the topology on the switch queue length. Therefore, it is no longer a traditional network structure with equal input and output, but

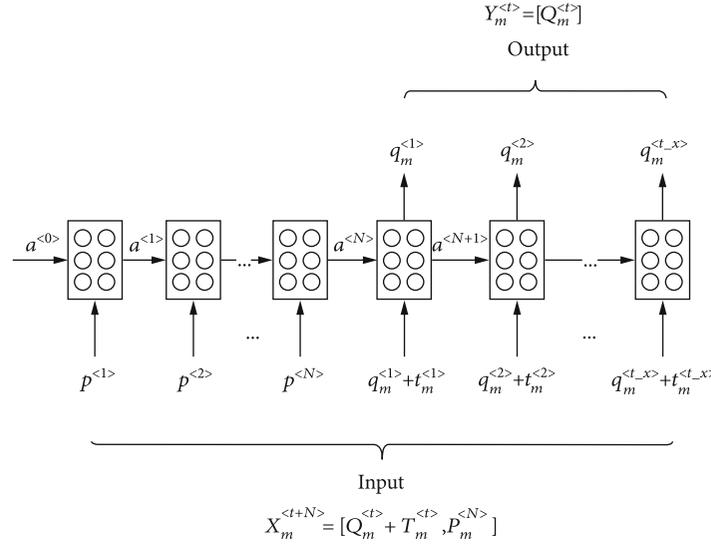


FIGURE 3: The improved LSTM structure.

the input sequence is longer than the output sequence. The sequence of network topology scores after PCA is considered; that is,  $P^{<t>} = [p^{<1>}, p^{<2>}, \dots, p^{<N>}]$ .

**4.2.2. Training Dataset.** To obtain the corresponding training data, we simulate some datasets and extract information about the topology, traffic patterns, and queue lengths. In addition, we run a traditional routing algorithm based on a given network traffic pattern and record the queue length of the switch for each time interval. We create a training set using the recorded queue lengths of the shortest path algorithm. The neural network is then trained on the switch by the TSN switch queue length prediction model.

**4.3. Training Phase.** Our proposed training of an LSTM neural network considering topological effects can be processed on a TSSDN controller. In this case,  $W_{ax}$  manages the input parameters from the input layer to the hidden layer, and the same parameters are used for each time step;  $W_{aa}$  manages the connections between each time step; similarly,  $W_{ay}$  manages the connections between the hidden layer and the output layer. The training is divided into two parts: (1) random initialization of the parameters  $\theta, w, b$  and (2) an adaptive estimation algorithm based on a low-order moment to adjust the parameters.  $\theta, w, b$  can be obtained through the training phase in the procedure below.

The forward propagation equation is

$$\begin{aligned} a^{<t>} &= \tanh(W_{aa}a^{<t-1>} + W_{ax}x_m^{<t>} + b_a), \\ y^{<t>} &= \text{sigmoid}(W_{ya}a^{<t>} + b_y). \end{aligned} \quad (6)$$

The activation functions we selected for the hidden layer and the output layer are  $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$  and  $\sigma(x) = 1/(1 + e^{-x})$ , respectively.

In TSNs, the current moment queue length of the switch is influenced by the input and output of the data stream at a historical moment. In other words, there is a complex tem-

poral correlation between TSN switch queue lengths and a variety of information lengths. Therefore, a gating unit was added inside the neuron, and the internal structure of the neuron is shown in Figure 4, with the aim of making this neural network memory capable of learning from historical data and making predictions.

In a supervised ANN, the inputs and outputs of the network model are the features and labels of the training samples. The network model obtains the predicted values after the corresponding computation of the features. The queue length of the next time slot is a mapping of the current queue length. We use the LSTM network to learn the patterns of historical data from the training samples. The parameters are adjusted to tap the relationship between topology, traffic patterns, and queue length. Also, the mapping is determined by the joint participation of the parameters of the neural network and the activation function.

We use the mean squared error (MSE) as the objective function of the neural network because it is able to indicate how close the predicted value is to the true label. Its mathematical representation is

$$\text{MSE} = L(\hat{y}, y) = \frac{1}{N} \sum_{t=1}^{t-x} (y^{<t>} - \hat{y}^{<t>})^2. \quad (7)$$

The gradient descent solution is obtained by the calculation of the MSE values, and the values of each network parameter are finally obtained by learning. The LSTM neural network is optimized by Adam when performing gradient descent operations, which is a stochastic gradient optimization algorithm.

**4.4. Running Phase.** The run phase is a forward propagation process in which the controller can predict the switch queue length for the next time slot based on the current traffic pattern. After the predicted queue length is computed, the controller assigns the switch queue length vector to each switch.

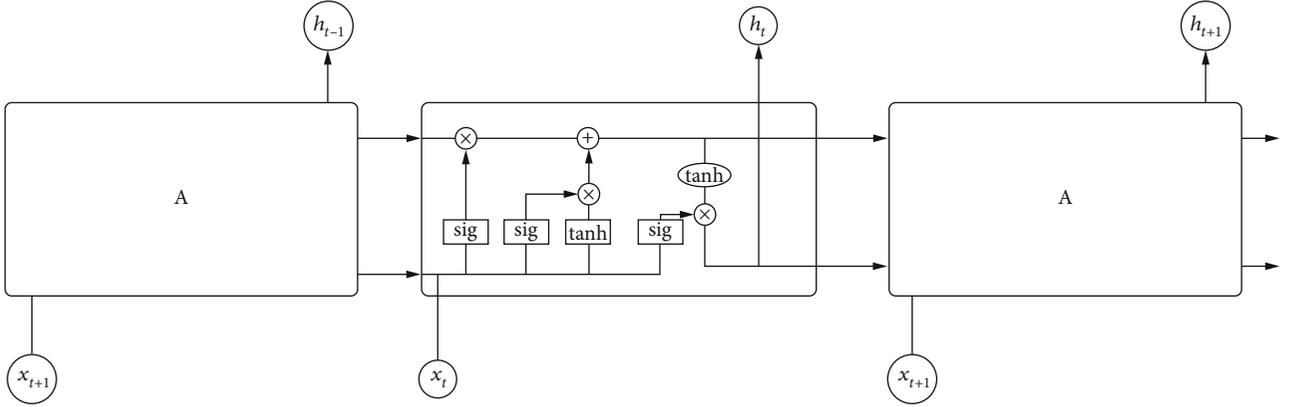
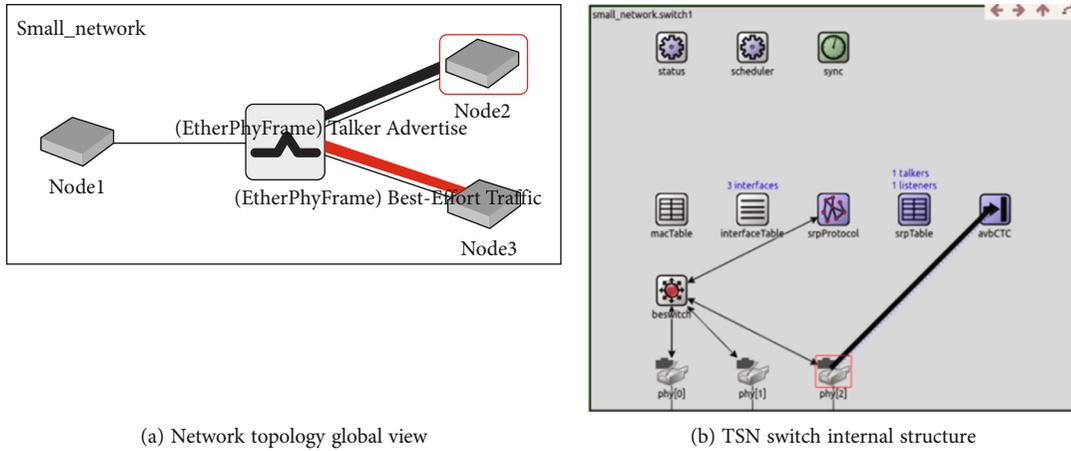


FIGURE 4: The internal structure of the neuron.



(a) Network topology global view

(b) TSN switch internal structure

FIGURE 5: TSN simulation environment.

TABLE 1: The parameter settings of LSTM.

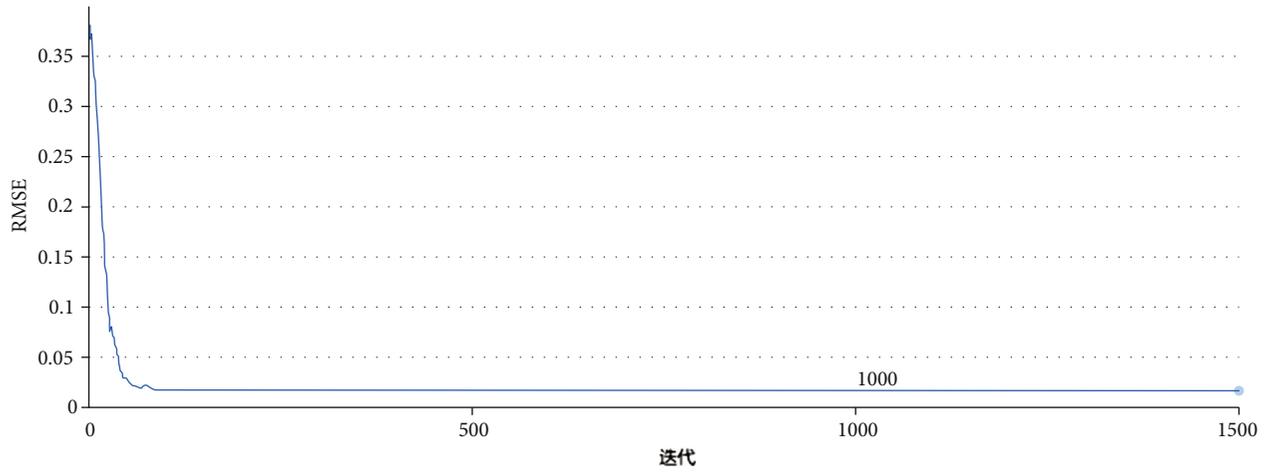
Input layer	Connection layer				Output layer
	Queue 1		Queue 2		
	Layer	1	Layer	1	
	HiddenUnits	200	HiddenUnits	400	
	InitialLearnRate	0.008	InitialLearnRate	0.005	
	LearnRateDropPeriod	100	LearnRateDropPeriod	100	
	LearnRateDropFactor	0.1	LearnRateDropFactor	0.2	
Length: 103	Queue 3		Queue 4		Length: 100
	Layer	1	Layer	1	
	HiddenUnits	150	HiddenUnits	180	
	InitialLearnRate	0.01	InitialLearnRate	0.008	
	LearnRateDropPeriod	100	LearnRateDropPeriod	100	
	LearnRateDropFactor	0.2	LearnRateDropFactor	0.2	

Queue 1 is taken as an example to demonstrate the training process, as shown in Figure 6.

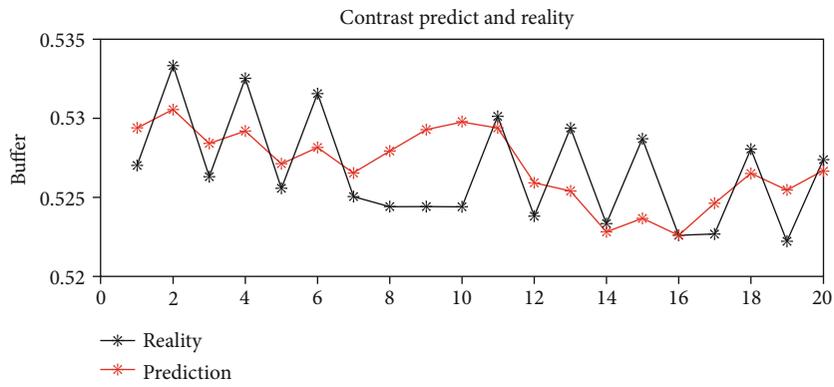
In the operation phase, the controller can make routing decisions based on the predicted value of the switch queue length for the next time slot to achieve network load balancing and assign the routing table to each switch.

## 5. Experiment

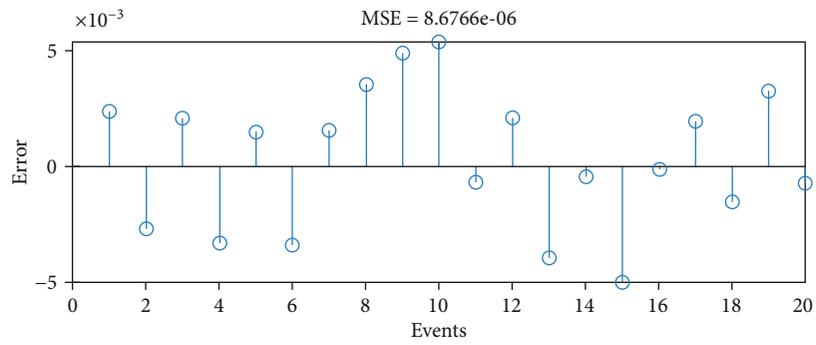
5.1. *Datasets.* We built a simulation environment for the TSN by using the INET and CORE4INET models in



(a) The MSE convergence process during training

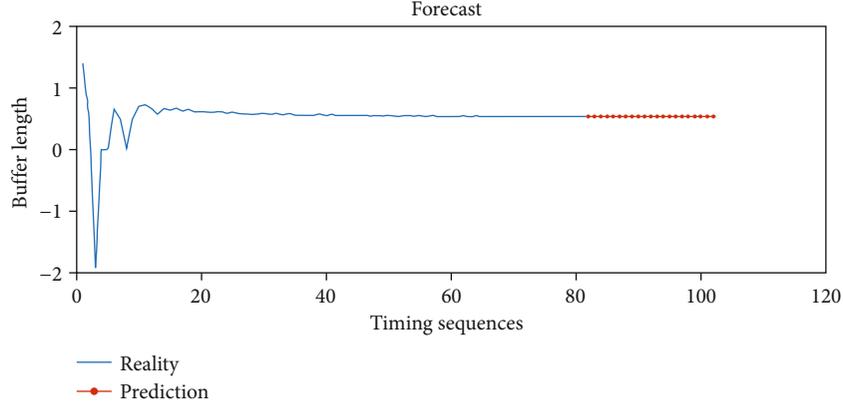


(b) Comparison between the real and predicted values of the switch queue length

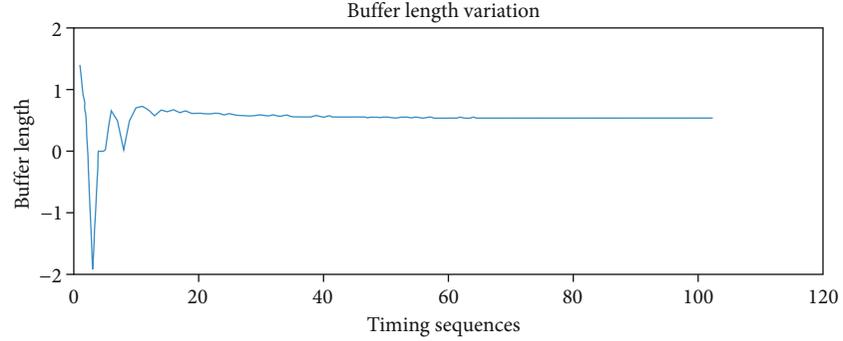


(c) MSE per test sample

FIGURE 6: Continued.



(d) Change in the switch queue length after prediction



(e) Change in the switch queue length

FIGURE 6: The training performance of Queue 1.

Omnet++. Omnet++ is an extensible, modular, component-based C++ emulation library and framework primarily for building network simulators. CORE4INET is an event-based implementation of the Ethernet emulation framework in INET, who provides real-time Ethernet protocols. Currently, CORE4INET supports Ethernet (AS 6802), IEEE 802.1 audio/video bridging/time-sensitive networking, IEEE 802.1Q, and priority. In this environment, there are three hosts and one switch, which send three types of flows, TT, AVB, and BE, at regular intervals, as illustrated in the figure. The topology of this network can be expressed as a complete graph of three nodes because there are three input and output interfaces in one switch that interconnect with each other. Additionally, we indicate the length of the buffer by the number of the buffer, and when the buffer is filled, the packets are discarded randomly.

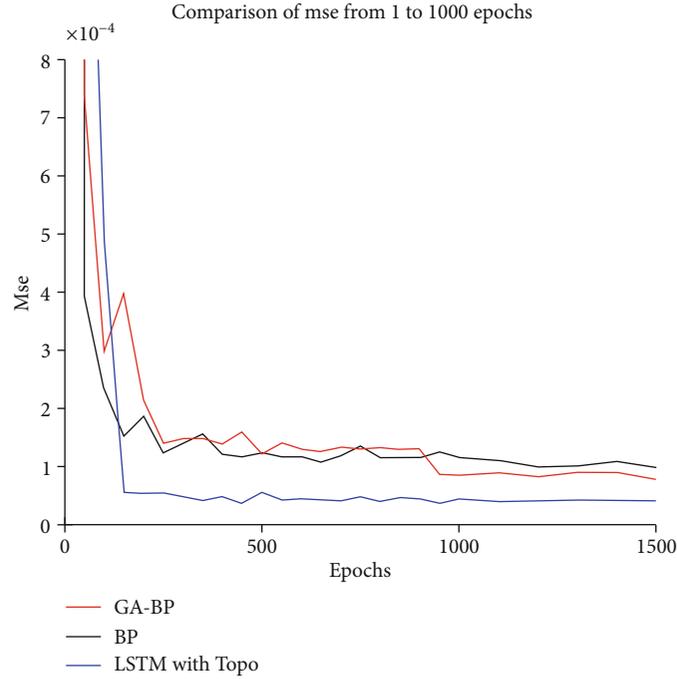
In the simulations performed, we set the time interval for switch queue length updates to once every 100 events. During this time, packets are forwarded and processed. The recorded traffic patterns and switch queue lengths are then used as the training set for our proposed neural network to predict the queue length for the next time slot. We use the first 80% of the data for training and learning and the remaining 20% for prediction testing. Due to the good deterministic trial transmission performance of the TSN and because the time of the simulation for data collection reaches approximately 20 min, which is not particularly long, not all queues within the switch are congested. Moreover, bursts of traffic occur in the initial

stage of scheduling, so the dataset also contains cases of burst traffic. Based on this, only the queues with nonzero queue lengths are predicted, for a total of four queues. The information from the reduced dimensionality of the principal components of the network topology is considered the feature input. The input timing length of the prediction training model considering the topology case is 103 (because the network topology is a complete graph with 3 nodes) and the output length is 100.

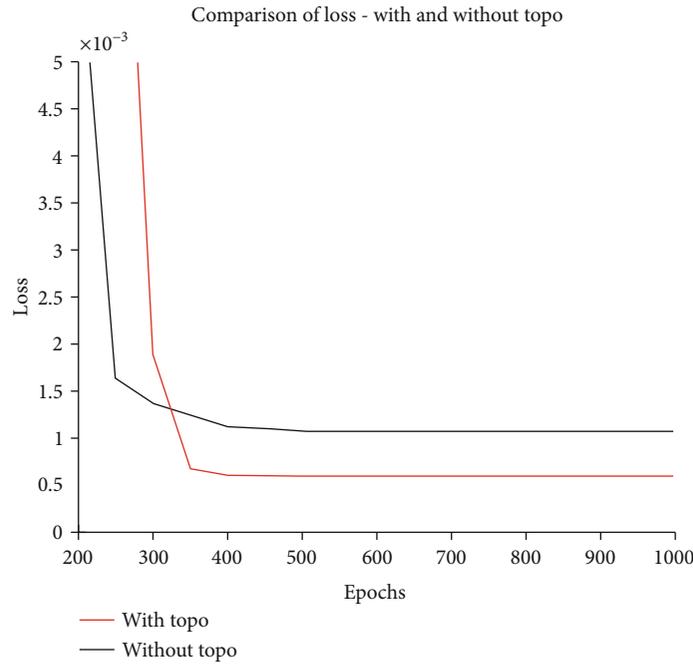
The queue lengths before and after each time interval are recorded as the input and output. The test set contains queue length information for 2000 events. First, we improve the input-output structure of the proposed neural network training model and optimize the network parameters during the training phase. Then, we use the obtained network parameters to perform length prediction on the test set. The network in the simulation environment is shown in Figure 5.

Currently, only one switch is used in the TSN emulation network. There are three physical interfaces in the switch, and each interface has 8 queues. Then, the connection relationship between the physical interfaces in the switch can be represented by a complete graph with a node number of 3. That is,

$$X = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (8)$$



(a) Performance on BP/GA-BP/LSTM with Topo



(b) Comparison of the performance when considering and not considering the topology

FIGURE 7: Demonstration and comparison of the predicted performance.

5.2. *Training Settings.* We build the neural network model by using MATLABR2021a programming environment. The parameters are optimized by the Adam algorithm. As for the setting of network parameters, the output value is the length of the queue in the switch buffer. The activation functions of the hidden layer and the output layer are  $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$  and  $\sigma(x) = 1 / (1 + e^{-x})$ , respectively. Then, we choose MSE which can be used as a loss function for learning in the gradient descent method to reduce the

difference between the predicted result and the output result. Additionally, the learning rate decreases continuously during the training process to slow down the learning speed to avoid skipping the optimal values. Finally, the number of epochs is determined by overall observation. The parameter settings of the LSTM neural network are specified in Table 1.

Figure 6(a) shows the process of the MSE of the network gradually converging with the increase of the number of iterations during the training process; Figure 6(b) shows the

comparison between the real data in the test set and the data predicted by the network model; Figure 6(c) shows the MSE of each data point in the test set; Figures 6(d) and 6(e) show the comparison between the real queue length and the predicted queue length. From Figure 6, the difference is very narrow.

**5.3. Training Results.** MSE is a metric to evaluate the accuracy of prediction. The learning rate and the number of iterations determine the convergence of the training process, so an appropriate value should be chosen. As seen from the above figure, the number of iterations is chosen to be 1000 to accomplish the training well.

The LSTM considering the influence of topological features is an optimized upgrade of an RNN, which has a natural structural advantage over conventional fully connected neural networks when dealing with sequential data. Therefore, to compare our proposed neural network model and the traditional neural network model for predicting the TSN switch queue length, we selected a BP neural network and a genetic optimization-based BP (GA-BP) neural network to compare the effect with our proposed prediction model. As the routing planning was carried out for the deterministic network, Haipeng et al. had used BP neural network to predict the router queue and achieved good results. Meanwhile, the prediction effect of BP neural network will be further improved after adding genetic algorithm. Therefore, we choose two algorithms, BP and GA-BP, to compare with our proposed prediction algorithm to reflect the prediction effect of the improved LSTM. From Figure 7(a), it can be seen that the error of LSTM with Topo during training is the smallest at the maximum number of iterations of 1500. Moreover, the training convergence speed of BP and GA-BP is slower than that of LSTM with Topo, and the MSE of LSTM with Topo converges at approximately 200 iterations with almost no fluctuation subsequently, while BP and GA-BP show significant fluctuation after convergence at approximately 300 iterations. The MSE of BP and GA-BP prediction is twice as high as that of LSTM with Topo. Therefore, our proposed prediction model can perform well for TSN switch queue length prediction compared to the traditional neural network prediction model.

Figure 7(b) shows that when we perform queue length prediction for TSN switches, the loss value of the switch queue length prediction when the network topology is considered is smaller than the loss value when the network topology is not considered. Therefore, the effect of the network topology on switch queue length cannot be ignored when performing queue prediction for switches. Although the convergence speed is slightly slower after considering the effect of topology, its accuracy is significantly improved. The loss value without considering topology is 0.00106, while the loss value considering the influence of topology is 0.00059.

In conclusion, when predicting sequential data such as the switch queue length, neural network models that consider historical inputs, such as RNN and LSTM, should be preferred. In addition, when predicting sequential data, the

influencing factors should be considered comprehensively. In this paper, after considering the influence of network topology on the switch queue length, the prediction accuracy is significantly higher than when the influence of the network structure is not considered; it is approximately double.

## 6. Conclusion and Future Prospects

Based on the LSTM network, the output result of switch queue length is not only affected by the current traffic flow to the switch but also depends on the previous switch queue length information, which is consistent with the actual situation of TSN transmission. In addition, when determining the input and output characteristics, we not only consider the time variation of switch queue length but also include the important feature of global network topology. Therefore, we further improve the accuracy of the switch queue length prediction results by changing the input and output structures of the original LSTM network. In this paper, we also demonstrate the effectiveness and enhancement of our proposed TSN switch queue length prediction method considering topology in simulation experiments.

Additionally, this reasonable and accurate prediction provides the next step for the TSSDN controller to perform intelligent dynamic routing based on the prediction results. Future dynamic routing based on queue prediction is composed of two main parts: (1) the TSSDN controller predicts the length of the next time slot switch queue; (2) the controller sends down the routing table to the switch to complete the routing. When no congestion occurs as a result of the prediction, the TSN switches can complete the scheduling based on their own routes during a period when the TSSDN controller continuously updates the queue lengths of each switch. If a queue is predicted to be congested soon, the controller will reasonably allocate the next hop for the data flow according to the global network situation, thus reducing the waiting time in the buffer and achieving load balance in the network, increasing the throughput of the whole network.

In the future, the TSN intelligent routing algorithm based on switch queue length prediction can be improved in the following respects. (1) Dynamic routing requires the TSSDN controller to collect the switch queue length for prediction in real time. This process is time-consuming, and we need to compare this time with the time that the data stream waits in the buffer. If the prediction time is long, it means that the dynamic routing algorithm does not optimize the network transmission delay; on the contrary, if the waiting time is long, it can only show the progress of the dynamic routing algorithm. (2) There are 8 TSN switch priority queues, and predicting all of them will increase the burden on the controller. We can also reduce the dimension of these eight queues in a way similar to PCA to simplify them before prediction, or we can predict only the exit traffic of the switch when only one type of traffic flows through, which greatly reduces the computation of the controller. (3) Future intelligent routing based on queue prediction can model the complex relationship between multiple network metrics to achieve a better load-balanced network.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Key R&D Program of China (2018YFB1700103) and the National Natural Science Foundation of China (61903356).

## References

- [1] S. Wang, B. Wu, C. Zhang, Y. Huang, T. Huang, and Y. Liu, "Large-scale deterministic IP networks on CENI," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, Vancouver, BC, Canada, 2021.
- [2] A. Arsalan and R. A. Rehman, "Distance-based scheme for broadcast storm mitigation in named software defined vehicular networks (NSDVN)," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–4, Las Vegas, NV, USA, 2019.
- [3] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location based routing in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1509–1522, 2005.
- [4] Z. Jintao, *Simulation Study on Predictable Service Flow Scheduling Mechanism in Time-Sensitive Networks*, Xi'an University of Electronic Science and Technology, 2020.
- [5] M. A. Ojewale, P. M. Yomsi, and B. Nikolić, "Multi-level pre-emption in TSN: feasibility and requirements analysis," in *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 47–55, Nashville, TN, USA, 2020.
- [6] A. M. Kentis, M. S. Berger, and J. Soler, "Effects of port congestion in the gate control list scheduling of time sensitive networks," in *2017 8th International Conference on the Network of the Future (NOF)*, pp. 138–140, London, UK, 2017.
- [7] N. G. Nayak, F. Dürr, and K. Rothermel, *Routing Algorithms for IEEE802.1Qbv Networks. RTN Workshop*, ECRTS, 2018.
- [8] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*, Elsevier, 2004.
- [9] K. A. Nsiah, K. Alkhouri, and A. Sikora, "Configuration of wireless TSN networks," in *2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pp. 1–5, Dortmund, Germany, 2020.
- [10] M. Erel, E. Teoman, Y. Özçevik, G. Seçinti, and B. Canberk, "Scalability analysis and flow admission control in mininet-based SDN environment," in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 18–19, San Francisco, CA, USA, 2015.
- [11] M. Baddeley, R. Nejabati, G. Oikonomou, S. Gormus, M. Sooriyabandara, and D. Simeonidou, "Isolating SDN control traffic with layer-2 slicing in 6TiSCH industrial IoT networks," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 247–251, Berlin, Germany, 2017.
- [12] N. G. Nayak, F. Dürr, and K. Rothermel, "Software-defined environment for reconfigurable manufacturing systems," in *2015 5th International Conference on the Internet of Things (IOT)*, pp. 122–129, Seoul, Korea (South), 2015.
- [13] M. Boehm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser, "Time-sensitive software-defined networking: a unified control-plane for TSN and SDN," in *Mobile Communication - Technologies and Applications; 24. ITG-Symposium*, pp. 1–6, Osnabrueck, Germany, 2019.
- [14] H. Xu and C. Jiang, "Deep belief network-based support vector regression method for traffic flow forecasting," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2027–2036, 2020.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] N. G. Nayak, F. Dürr, and K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2066–2075, 2018.
- [17] S. Rajendran, A. Kaul, R. Nath, A. S. Arora, and S. Chauhan, "Comparison of PCA and 2D-PCA on Indian faces," in *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*, pp. 561–566, Ajmer, India, 2014.
- [18] D. K. Jain, R. Jain, X. Lan, Y. Upadhyay, and A. Thareja, "Driver distraction detection using capsule network," *Neural Computing and Applications*, vol. 33, no. 11, pp. 6183–6196, 2021.
- [19] J. S. Rahhal and D. Abualnadi, "IOT based predictive maintenance using LSTM RNN estimator," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–5, Istanbul, Turkey, 2020.