IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 June 2022, accepted 5 July 2022, date of publication 7 July 2022, date of current version 18 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3189354

RESEARCH ARTICLE

Moderately Multispike Return Neural Network for SDN Accurate Traffic Awareness in Effective 5G Network Slicing

NAJWAN SATTAR SOUD[1], NADIA ADNAN SHILTAGH AL-JAMALI[1], (Member, IEEE), AND HAMED S. AL-RAWESHIDY[2], (Senior Member, IEEE)

[1]Department of Computer Engineering, University of Baghdad, Al-Jadriya, Baghdad 10071, Iraq
[2]Department of Electronic and Electrical Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, U.K.

Corresponding author: Nadia Adnan Shiltagh Al-Jamali (nadia.aljamali@coeng.uobaghdad.edu.iq)

This work was supported by the Brunel University London, U.K.

ABSTRACT Due to the huge variety of 5G services, Network slicing is promising mechanism for dividing the physical network resources in to multiple logical network slices according to the requirements of each user. Highly accurate and fast traffic classification algorithm is required to ensure better Quality of Service (QoS) and effective network slicing. Fine-grained resource allocation can be realized by Software Defined Networking (SDN) with centralized controlling of network resources. However, the relevant research activities have concentrated on the deep learning systems which consume enormous computation and storage requirements of SDN controller that results in limitations of speed and accuracy of traffic classification mechanism. To fill this gap, this paper proposes Intelligent SDN Multi Spike Neural System (IMSNS) by implementing Moderately Multi-Spike Return Neural Networks (MMSRNN) controller with time based coding achieving remarkable reduction on energy consumption and accurate traffic identification to predict the most appropriate network slice. In addition, this paper proposes another intelligent Recurrent Neural Network (RNN) controller for load balancing and slice failure condition. The current researchers have adopted the: accuracy, precision, recall and F1-Score, the simulation results revealed that the proposed model could provide the accurate 5G network slicing as compared with a convolutional neural network (CNN) by 5%.

INDEX TERMS Slicing, 5G, return neural networks, intelligent multispike neural network.

I. INTRODUCTION

With The exponential growth of the communication devices especially with rise of 5G services, these devices require reliability, low latency, high bandwidth and better QoS to achieve high service satisfaction rates. So it became necessary to adopt Network slicing and resource allocation mechanism [1]. Network slicing refers to selecting appropriate slices for the specific traffic type to provide better-performing and cost-efficient services. Identifying the traffic application types is an essential function to configure network slicing that facilitates fine-grained management and resource utilization [2], [3]. An efficient and fast classification algorithm is required to realize application awareness because of the different network resource requirements of different applications [4], [5]. Until now, the main approaches of traffic classification are port based approach, Deep Packet Inspection (DPI) and Machine Learning (ML). Due to the dynamic usage of port numbers by applications, the port-based method can no longer be used. Some protocols avoid using fixed port numbers for security reasons [6]. On the other hand, DPI is costly computation approach. Also, it is unable to inspect encrypted traffic [7]. Based on the aforementioned, ML would be considered as perfect choice for network analysis of the huge data and automation of functions in network slicing in a limited time without

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

human intervention. However, the forwarding element in traditional network maintains a limited view over the entire network. Hence, applying ML is challenge [8]. The software-Defined Networking (SDN) platform provides an opportunity to include ML into the network devices because of its centralized management and the global view of network [9]–[14]. SDN and Network Function Virtualization (NFV) are the enabling technologies of Network slicing [15], [16]. As a result of this, network operators can sell the customized slices to various tenants at different prices that result in cost effectively delivering several logical resources over the same physical infrastructure. Currently, there are three main services belong to 5G communications: Enhanced mobile broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC) [17]. Each one must meet specified requirements as shown in table 1. We aim to build SDN network model that is able to improve the QoS by implementing fast and effective prediction algorithm to assign the dedicated network slice for a specific service request with high accuracy. Deep Neural Network (DNN) and Convolutional Neural Network (CNN) have achieved attractive records on classification and recognition problems. However, the massive computation, complex neural network models and storage requirements associated with such DNN and CNN challenge their employment on platforms with limited resources. [18]. Time-based spiking neural network (SNN) would be considered as a solution for this problem; it is the third generation of Artificial Neural Network (ANN). SNN is more hardware friendly and energy-effective than traditional ANNs [19]–[22]. Due to the limitations of single spike neural networks, important information is carried by multi-spike algorithm with temporal encoding because the simple single-spike form and rate coding cannot express this essential information [23]–[25]. Also, the reliability of the classification can be increased by implementing the Multi-Spike learning rule [26], [27].

Given the above consideration, the main contributions of this paper are:

1. The system frame work of IMSNS is proposed; we introduce a Moderately Multi-Spike Recurrent Neural Network (MMSRNN) as classifier controller to predict the most appropriate network slice for a specified traffic in the proposed model. The MMSRNN is a type of Multi-spike neural network with feedback from output layer to hidden layer.

2. We suggest RNN controller to realize accurate load balancing for efficient utilization of network slices and slice failure conditions.

3. A new training algorithm is proposed to update the weights and threshold of MMSRNN model.

The rest of the paper is organized as follows: Section II presents the related works. The proposed framework in this paper is explained in Section III. Section IV presents MMSRNN learning algorithm. In Section V, simulation and evaluation are specified. Finally, section VI concludes this paper.

II. THE RELATED WORKS

Application awareness and intelligent capabilities in SDN are always attractive research areas. The most recent researches relating to the use of deep learning (DL) in traffic identifications and network slicing are introduced in this section. Abidi *et al.* [2] proposed hybrid deep learning algorithm which could influence the provision of accurate 5G network slicing as compared with the other deep learning and ML learning algorithms. Thantharate *et al.* [4] developed a DeepSlice to make smart decisions and predict the most appropriate network slice for a certain traffic, even in case of a network failure using key performance indicators (KPIs). For same goals as before, Khan *et al.* [1] developed hybrid deep learning model that consists of CNN and long short term memory (LSTM), this model achieved an overall accuracy of 95.17%. In [8] Malik *et al.* proposed a new deep SDN model for accurately identifying a wide range of traffic applications in a short time. The proposed model achieved higher overall accuracy comparing against the state-of-the-art models. Based on flow classified priority using intelligent AMPS controller, Pasca *et al.* [10] ensured a high availability for high priority flows even in loaded network. Authors in [7] adopted a CNN based deep learning mechanism to do the application-awareness in SDN environment. This proposed mechanism exceeded three benchmarks on recall ratio, precision ratio, F value and stability. Jun Xu *et al.* [12] deployed (DNN) for traffic classification on Virtualized Network Function (VNF) to solve the problems of I/O and computing resources of the SDN controller with deep learning in SDN. The proposed DNN model highly improved the network QoS. The above researches could address the SDN-based traffic classification to achieve intelligent resource allocation. However, when deep learning is adopted in SDN there are many issues such as a high consumption of computing resources of SDN controller. This paper implements time-based coding, Moderately Multi-Spike Return learning mechanism to implement the application –awareness and effective resource allocations that achieves better results in: energy efficiency, area efficiency, computation speed and power consumption. We compare our model with CNN traffic prediction model to evaluate the efficiency of the network.

III. THE PROPOSED FRAMEWORK

Novelty approach is proposed in this paper; it is based on SDN and MMSRNN supervised learning algorithm which implemented in SDN environment:

- The MMSRNN controller is used to realize application-aware, network resource allocation and network slice selection. We noticed the diversity of 5G devices. It involves smartphones, smart transportation, healthcare, wearable devices, industry 4.0, public safety, etc. Also we considered the variation in the requirements of the traffic types; (mMTC) devices require a continuous connection link with low

TABLE 1. 5G services.

Service type	QoS requirements	Applications
URLLC	Low latency $1ms$ High reliability, 10^{-9}error rate High availability	Healthcare , Factory automation, Self-driving car
eMBB	High data rate (20Gb/s) • High mobility	Smart phones • Augmented reality (AR) • Virtual reality (VR)
mMTC	High connection density (1million devise connections) • High energy efficiency	Smart cities • wearable devices • vehicle to infrastructure

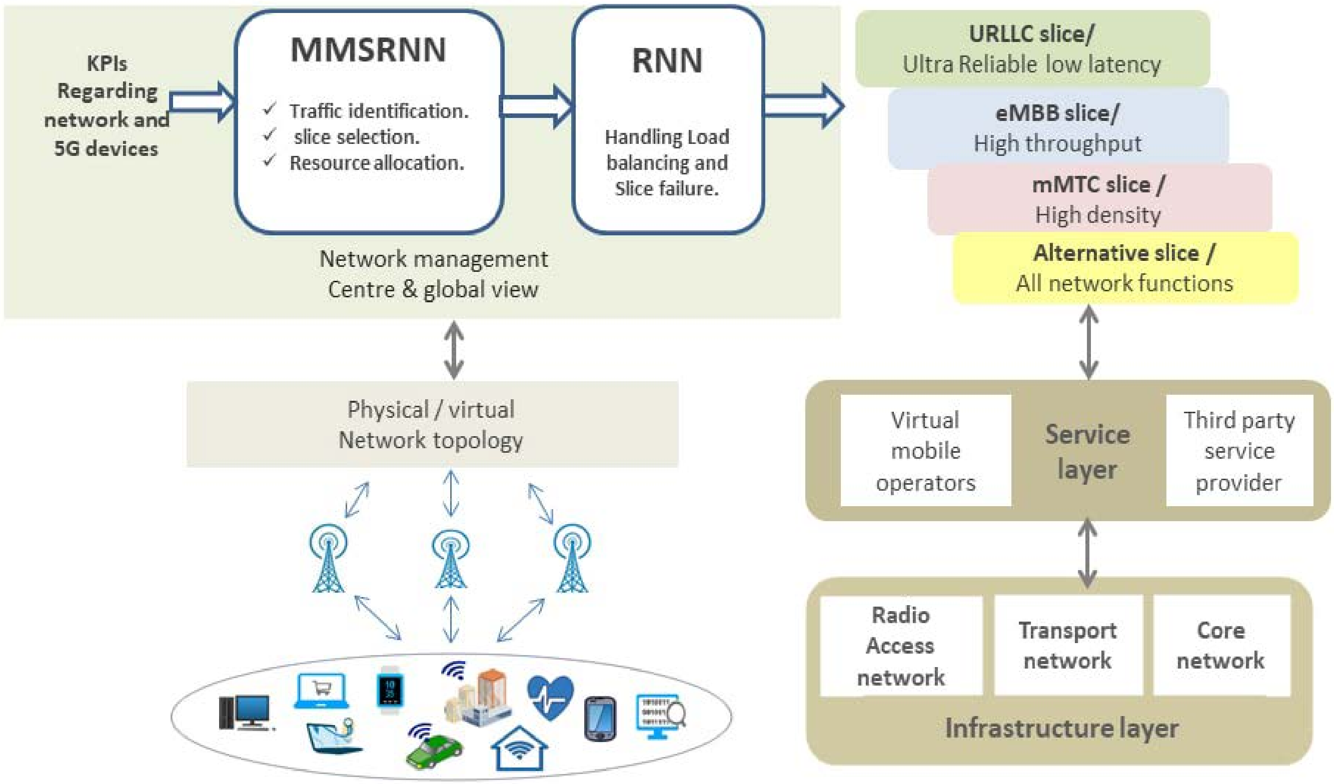


FIGURE 1. System architecture.

throughput but high density, while (eMBB) devices require a real-time and high throughput connection. As for (URLLC) devices, reliable and very low latency connection is needed. Our slice categories are (URLLC, eMBB and mMTC slice).

• Another controller which based on RNN is implemented for load balancing and slice failure conditions. Depending on the information of the preceding requests, RNN model can predict the load of each network slice. Then, the traffic will be redirected to the alternative slice when one slice would be over utilized i.e. the number of slice requests goes beyond a threshold, we consider the threshold is 95%. Also in case of slice failure, the proposed model will route all new requests which are related to the failing slice to the alternative slice and avoid any loss of service in the network. The service quality of the network and resource utilization can be improved by these criteria. The overall system architecture is shown in Fig. 1 where the SDN application layer components represent the main contribution.

The effectiveness of data classification requires three main stages before training algorithm:

1. Data collection: the traffic information of SDN network can be collected by support of the global view of SDN network without any additional overhead. The controller works as the brain of the network which can collect the packets through the secure channel with OpenFlow switches.

2. Data pre processing: From the different devices or users, the attributes will be collected, such as user device type, duration, packet loss ratio, packet delay budget, bandwidth, delay rate, speed, jitter, and modulation type. However, it is also possible that the quality of data is not acceptable, there is noise, the data is missing, the data is wrong, the dimensions are different, there are duplicates, and the amount of data is too large or too small. In order for the data to match the needs of the model, the data set needs to be pre-processed via the operation of normalization. We normalized the data to make the values of attributes between 0 and 1.

3. Traffic feature extraction: After data pre-processing stage, we need to select meaningful features to input into the model for training. That means, the selected features must be divergent and highly relevant to objectives.

IV. MMSRNN TRAINING ALGORITHM

The proposed structure of MMSRNN is fully connected feed forward network as shown in the Fig.2. All three classes

TABLE 2. The feature highlights of simulation model.

Device type	Packet delay budget (ms)	Packet loss rate	Duration(s)	Predicted slice
IoT devices	60/300	10^{-3}	50	$mMTC$
Healthcare	10	10^{-6}	200	$URLLC$
Industry 4.0	10/50	$10^{-3}/10^{-6}$	180	mMTC/URLLC
Smartphone	60/75/100/ 150/300	10^{-2}/ 10^{-3}/ 10^{-6}	300	eMBB
Smart City / Home	50/300	10^{-2}	120	mMTC
Intelligent transportation	10	10^{-6}	50	URLLC
Unknown devices	$10/50/60/75/100/150/300$	$10^{-2}/10^{-3}/10^{-6}$	$60/120/180/300$	$eMBB/mMTC/URLLC$

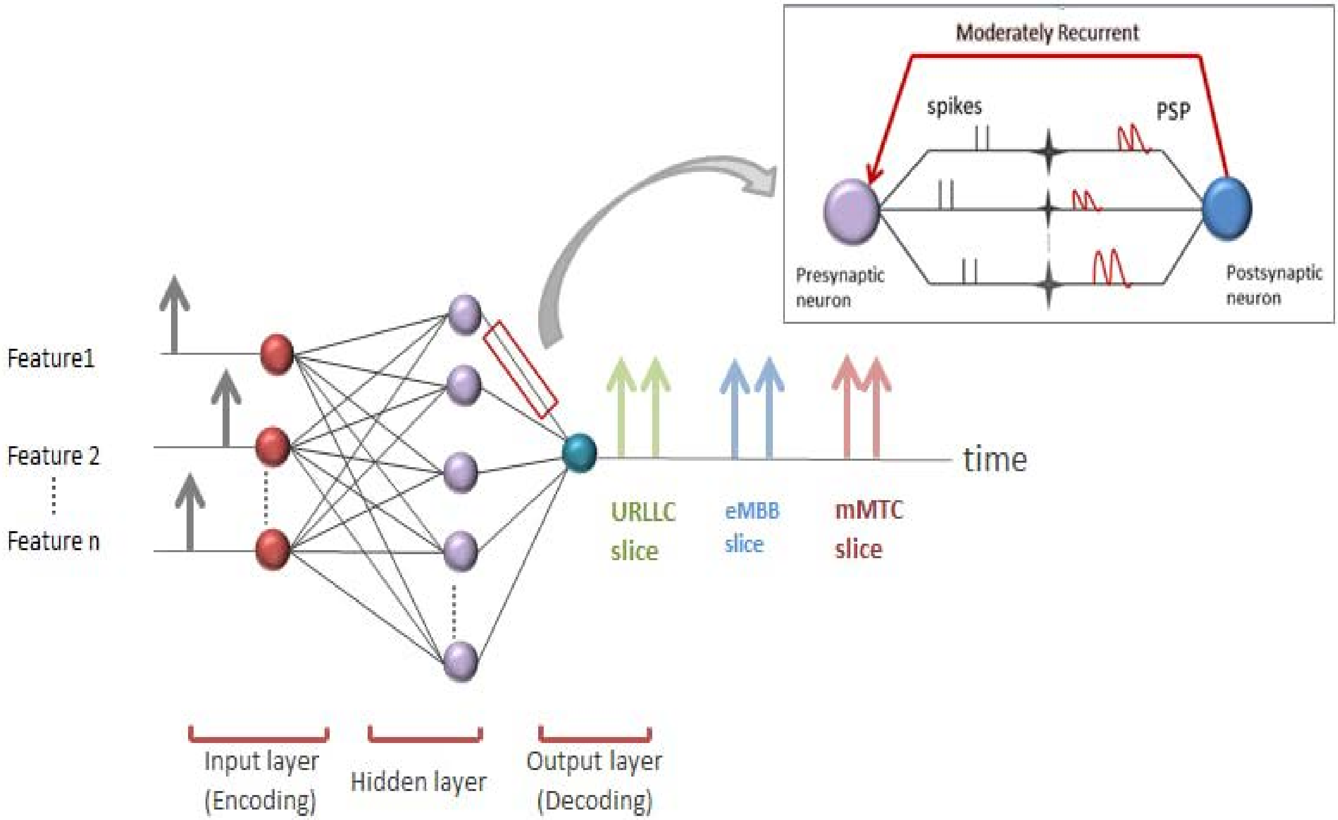


FIGURE 2. Structure of MMSRNN.

of network slices (URLLC, eMBB and mMTC Slice) are classified by MMSRNN controller with one output neuron whose output spikes times refer to the three classes. The number of neurons in the hidden layer is selected as seven. The inputs and outputs of all neurons are multiple spikes learned to be emitted at specific times. Two spikes are used to represent each class.

For the used model, we discuss three major parts, namely, encoding and decoding function, Neuron model function and modified learning algorithm.

A. ENCODING AND DECODING FUNCTION

Table 2 shows the features of our simulation model; they will be recorded and used by our intelligent model to make smart decision without human intervention. However, the values of the extracted features are real numbers. So that, encoding function is required to convert the real numbers to spike times as expounded below:

$$t_x^f = t_{max} - \lfloor \frac{t_{min}(f_e(t) - f_{emin})(t_{max} - t_{min})}{(f_{emax} - f_{emin})} \rceil. \qquad (1)$$

where, f_{emax} and f_{emin} represent the maximum and minimum certain extracted feature, while t_{max} and t_{min} are the maximum and minimum interval time, respectively. The $\lfloor\rceil$ is a round function. $f_e(t)$ refers to the extracted feature. The predicted network slice decoding $n_s(t_y)$ is explained in the equation below:

$$n_s(t_y) = \frac{(t_{max} - t_y - t_{min})(f_{emax} - f_{emin})}{(t_{max} - t_{min})} + f_{emin}. \qquad (2)$$

t_y represents the output spike time.

B. NEURON MODEL AND LEARNING ALGORITHM

The Spike Response neuron Model (SRM) is used to implement this algorithm due to its differentiated equation

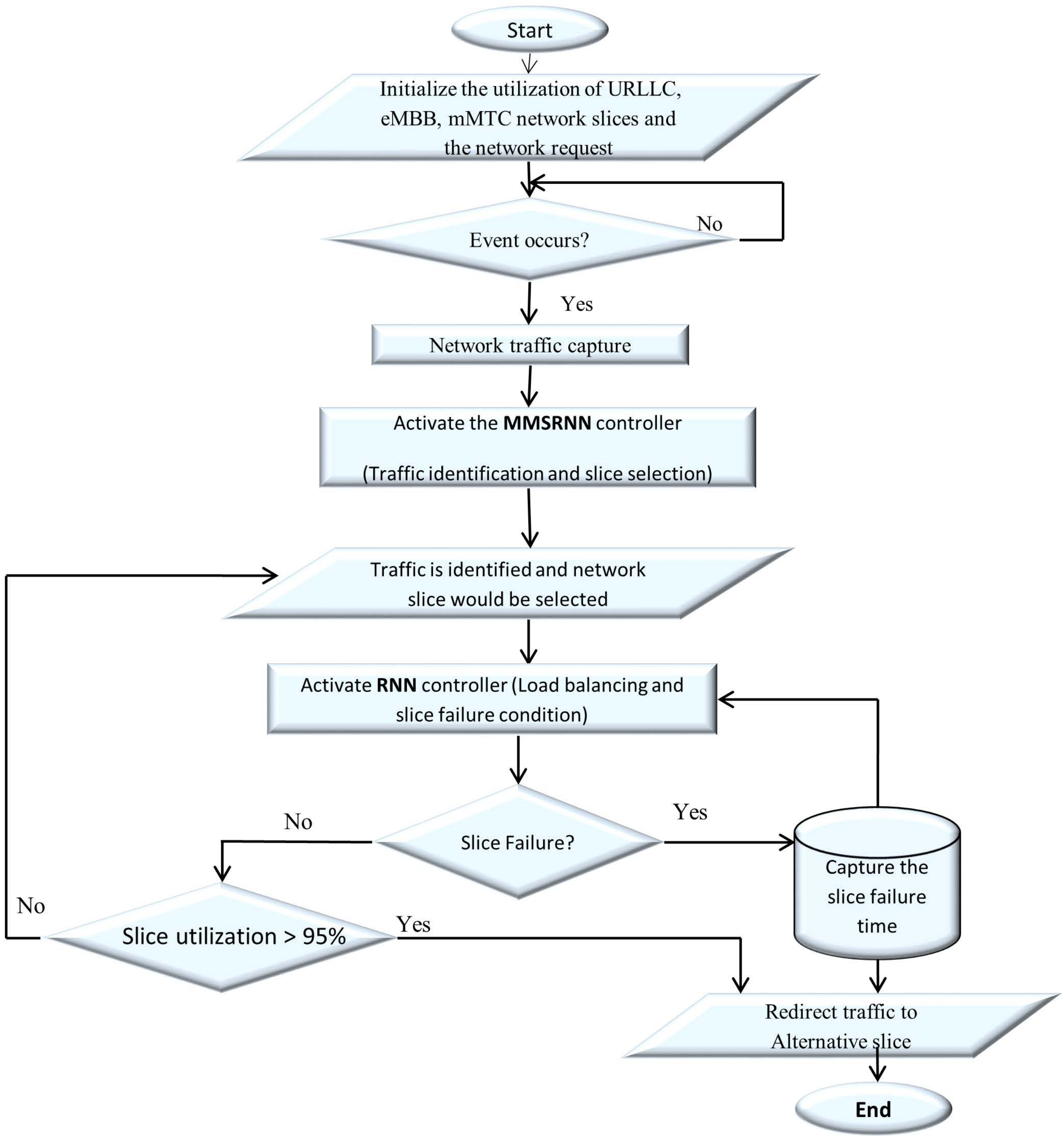


FIGURE 3. The flowchart of the proposed model.

which is suitable to learning algorithms based on gradient descent that require computation of partial derivatives. At the arrival of each spike, a postsynaptic potential (PSP) is motivated in the neuron if its membrane potential is below a threshold θ. It is important to know that membrane potential refers to the sum of the PSPs motivated by all input spikes. Also the weights of synapses that transmit these input spikes affect the value of PSP. The PSP is computed by the spike response function $\epsilon(t)$ as shown below:

$$\epsilon(t) = \begin{cases} t/\tau_1 * \exp(1 - t/\tau_1) & \text{t} > 0 \\ 0 & t \leq 0 \end{cases}. \tag{3}$$

where τ_1 represents time decay constant.

When the membrane potential exceeds the neuron threshold, the neuron fires an output spike at time t^f. At this

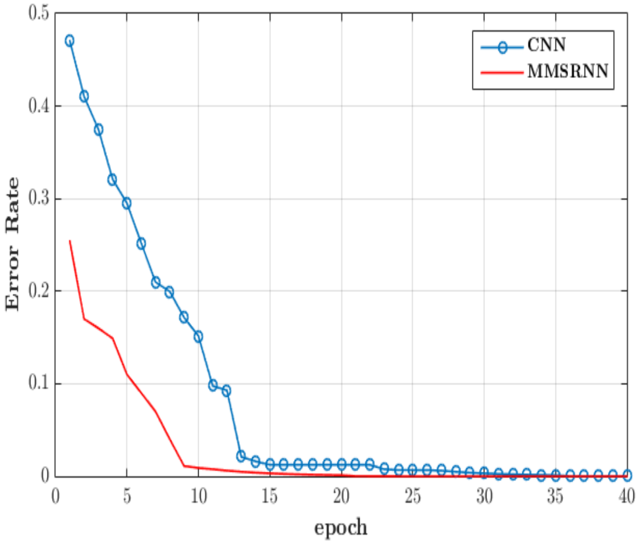


FIGURE 4. Error rate of the proposed model as compared with CNN model.

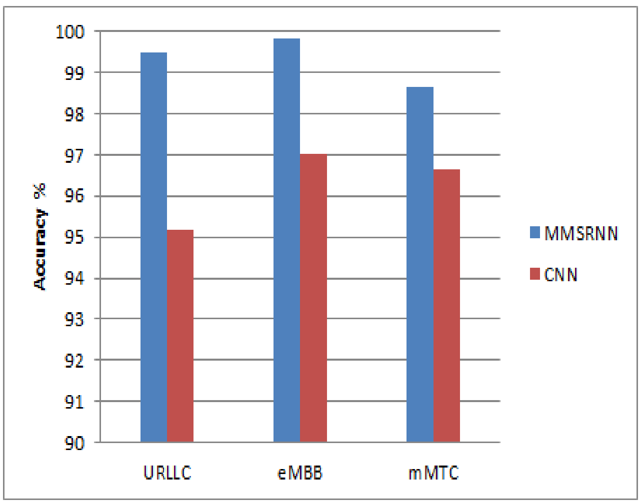


FIGURE 5. Accuracy ratio of the proposed model as compared with CNN model.

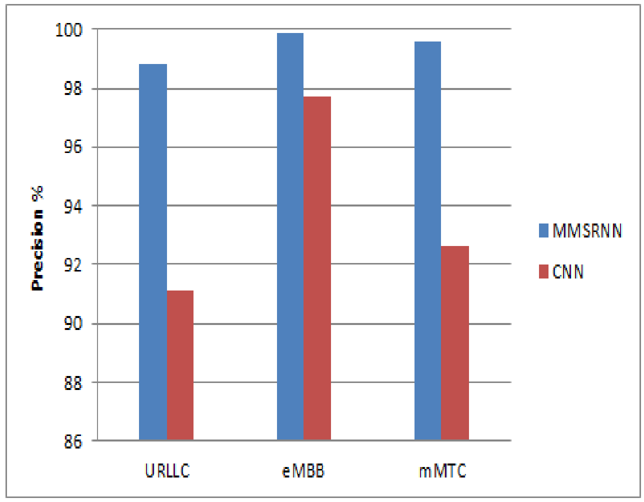


FIGURE 6. Precision ratio of the proposed model as compared with CNN model.

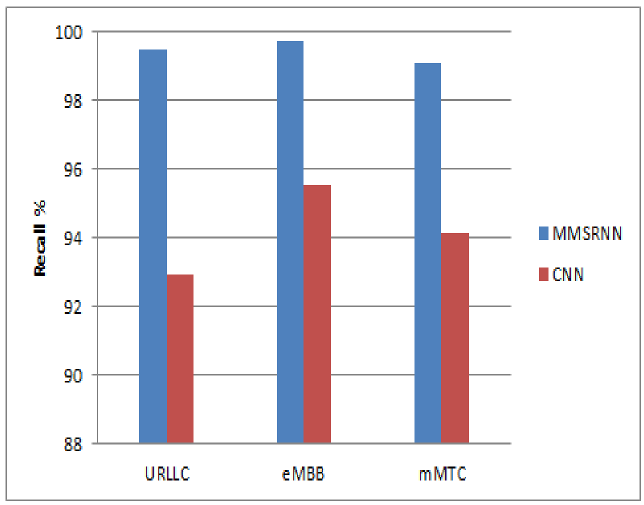


FIGURE 7. Recall ratio of the proposed model as compared with CNN model.

moment, the membrane potential decline to zero. Despite of the arrival of input spikes, the neuron cannot fire. This phase is called the absolute refractory period. After that, the membrane potential is stayed at a value lower than 0 so that the neuron cannot fire again for period of time. This phase is called the relative refractory period. Due to employment of Multi-spike learning algorithm, the membrane potential is influenced by these two phases. For this reason a refractoriness function term $\eta(t - t^{rfr})$ is added to the expression of membrane potential. The t^{rfr} is the time of the most recent output spike.

The function $\eta(t)$ is expressed as

$$\eta(t) = \begin{cases} -2\theta * \exp(-t/\tau_2) & \text{t} > 0 \\ 0 & t \leq 0. \end{cases} \tag{4}$$

where τ_2 represent time decay constant.

Membrane potential of the neuron must be determined using arrival times of input spikes after $t^{rfr} + Rp$, where Rp is the length of the absolute refractory period. To get the general expression of the membrane potential, let us consider the output layer counted as layer 1 while other layers are counted backwards starting from the output layer. In adjacent layers, (S) synapses, $S \in 1, 2, \ldots, S$ are used to connect each two neurons with different transmit delay (d^s) and weights. Suppose that MMSRNN has N_{l+1} neurons in layer $(l + 1)$ and presynaptic neuron $x(x \in 1, 2, \ldots, N_{l+1})$ has emitted a spike train that consists of F_x spikes with firing times which are represented by the set $F_x = t_x^1, t_x^2, \ldots, t_x^{F_x}$. The arrival time of spike t_x^f at postsynaptic neuron $y(y \in 1, 2, \ldots, N_l)$ in layer l through the S^{th} synapse is $t_x^f + d^s$. According to what was mentioned, the general expression of membrane potential function of a neuron with multi-spike learning can

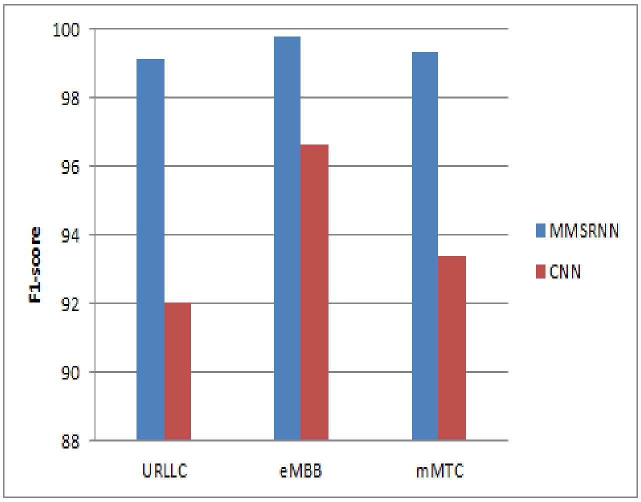


FIGURE 8. F1-Score ratio of the proposed model as compared with CNN model.

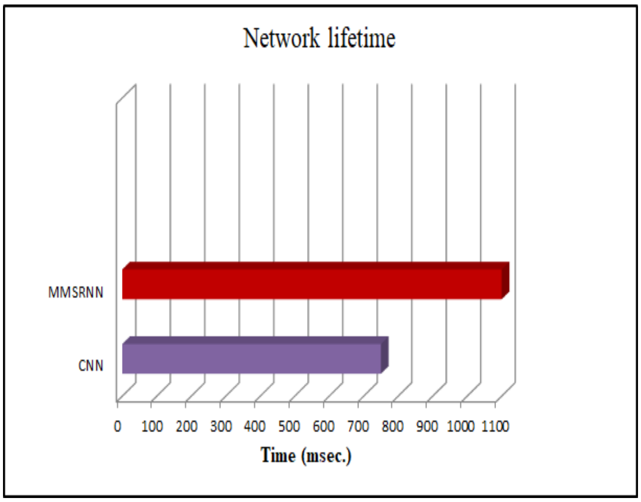


FIGURE 9. Network lifetime of the proposed model as compared with CNN model.

be expressed as:

$$m_y(t) = \sum_{x=1}^{Nl+1}\sum_{s=1}^{S}\sum_{t_x^f+d^S>t_y^{rfr}+R_p}^{t_x^f\in F_x} w_{xy}^s\epsilon(t-t_x^f-d^s) + \eta(t-t^{rfr}). \quad (5)$$

where w_{xy}^s refers to synapse weight between presynaptic and postsynaptic neuron. In view of our adoption of the return from output layer to hidden layer, another term will be added to the membrane potential function of the hidden layer. The Moderately Return term is expressed as below:

$$MR = \alpha\sum_{y=1}^{N_y}\sum_{s=1}^{S} w_{hy}^s * n_y^s(t-1) \quad (6)$$

where, α is the learning rate, w_{hy}^s refers to weights between hidden layer (h) and output layer (y) while $n_y^s(t-1)$ represents the previous output of the output layer. Thus, the membrane

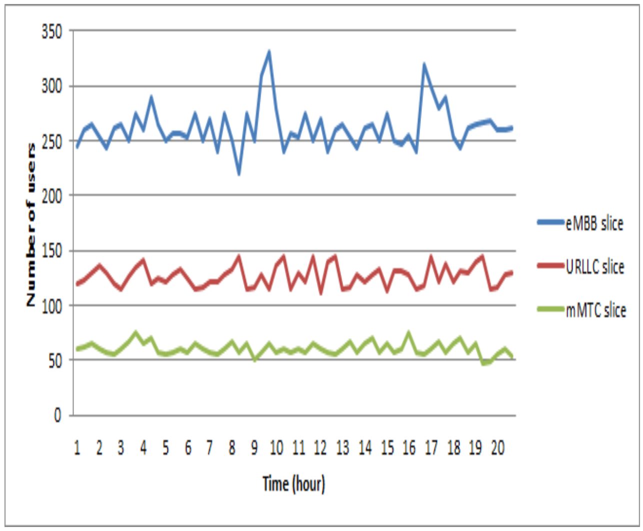


FIGURE 10. Number of active user in the network checked every 20 min.

potential of the hidden layer for our MMSRNN model is computed by the following equation:

$$m_h(t) = \sum_{x=1}^{N_x}\sum_{s=1}^{S}\sum_{t_x^f\in F_x} t_x^f + d_s > t_h^{rfr} + Rp \times w_{xy}^s\epsilon(t-t_x^f-d^s) + \eta(t-t_{rfr}) + MR \quad (7)$$

The error function is computed by the mean square error (MSE) that measures the average of the squared difference between the target and actual spike time for multi-spikes as below:

$$E = 1/2\sum_{y=1}^{N_l}\sum_{f=1}^{F_y}(t_y^f - t_y^{\hat{f}})^2 \quad (8)$$

t_y^f represents actual output spike time, $t_y^{\hat{f}}$ represents target spike time. F_y refers to the number of the output spikes while N_l represents the number of the output neurons. To minimize the error function, the S^{th} synapse weights between the h^{th} presynaptic neuron (at hidden layer) and the y^{th} postsynaptic neuron (at output layer) would be updated by taking advantage of gradient descent and expressed as:

$$w_{hy}^s(t+1) = w_{hy}^s(t) + \Delta w_{hy}^s(t)\,. \quad (9)$$

where,

$$\Delta w_{hy}^s(t) = -\alpha.\nabla E_{hy}^s. \quad (10)$$

The synaptic weights between x_{th} neuron in the input layer and h_{th} neuron in the hidden layer are modified as expressed below:

$$w_{xh}^s(t+1) = w_{xh}^s(t) + \Delta w_{xh}^s(t)\,. \quad (11)$$

where,

$$\Delta w_{xh}^s(t) = -\alpha.\nabla E_{xh}^s. \quad (12)$$

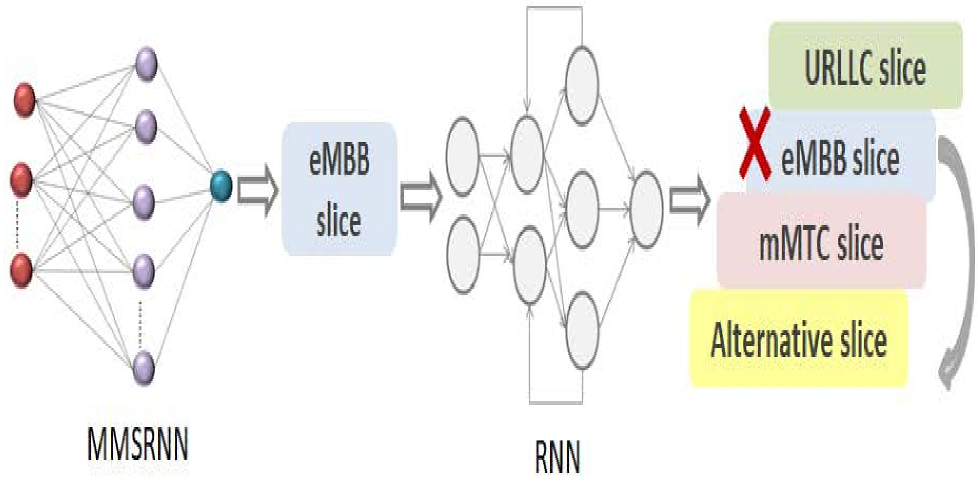


FIGURE 11. The block diagram of Scenario 1.

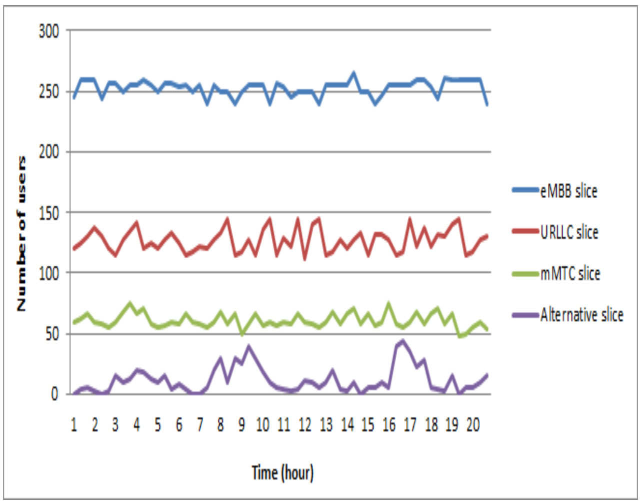


FIGURE 12. Load balancing when connection requests exceeds threshold.

Also, the synaptic delay and the threshold are updated in the following equations:

$$\Delta^{s}_{xh} = -\rho_d \sum_{(h=1)}^{(NH)} \frac{\partial E}{\partial t^{f}_{y}} \frac{\partial t^{f}_{y}}{\partial q^{s}_{x}(t)} \frac{\partial q^{s}_{x}(t)}{\partial d^{s}_{xh}}. \quad (13)$$

$$\Delta\theta_y = -\rho_\theta \sum_{(h=1)}^{(NH)} \frac{\partial E}{\partial t^{f}_{y}} \frac{\partial t^{f}_{y}}{\partial q^{s}_{x}(t)} \frac{\partial q^{s}_{x}(t)}{\partial \theta_y}. \quad (14)$$

For abbreviation, we defined the function $\epsilon(t - t^{f}_{x} - d^{s})$ as q^{s}_{x}. ρ_d and ρ_θ represent the learning rate of the synaptic delay and synaptic threshold respectively. The Flowchart of the proposed model is shown in Fig.3.

V. EVALUATION PERFORMANCE

We executed the proposed 5G network slicing model in Python. The collected dataset [4] consists of 65000 unique items. To train the model, 70% of input dataset was used while 30% was used for testing. The training data set have about 8 Key Parameter Indicators (KPIs) from both the networks and the devices to identify the requested services and assign the correct network slice for all incoming slice request. In this paper we focus on measuring the efficiency of the proposed MMSRNN controller based on evaluation metrics: accuracy, precision, recall, and F1-score. For MMSRNN, the used parameters of each neuron are specified in the table 3. Any presynaptic neuron is connected to a postsynaptic neuron by six synapses with delays of 1,4,7,10,13 and 16 ms. Then, we compared the performance of the proposed model against the other controller based on CNN controller using the same value of learning rate which is used by MMSRNN model to achieve fair comparing. The CNN is fully connected model and includes one convolutional layer and a ReLU (Rectified Linear Unit) activation function layer. It should be noted that the RNN model which is used for load balancing has two hidden layers using ReLU activation function. Fig.4 shows the error rate of the proposed model as compared with CNN model. It is clear that the proposed model can learn faster and be more accurate than CNN model. Fig 5, Fig 6, Fig 7 and Fig 8 Show the accuracy,precision,Recall and F1-score measures respectively of the proposed model in network slicing as compared with CNN model for the three network slices.

TABLE 3. The parameters of MMSRNN.

Parameters	Values
τ_1	10 ms
τ_2	70 ms
R_p	1 ms
θ	1 v
α	0.001

The key classification metrics are determined as follows:

• Accuracy: represents the rate of correct samples classification.

• Recall: specifies the ratio of the positive class prediction to the all positive samples in the data set.

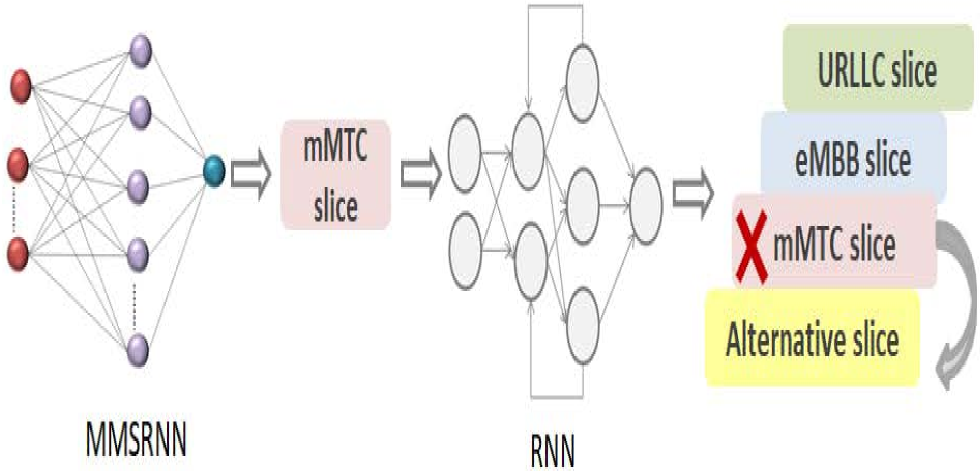


FIGURE 13. The block diagram of Scenario 2.

TABLE 4. Overall performance analysis of the proposed MMSRNN and CNN.

Performance metrics	MMSRNN	CNN
Accuracy %	99.492 %	94.415 %
Precision %	99.430 %	93.83 %
Recall %	99.431 %	94.209 %
F1-score %	99.423	94.01

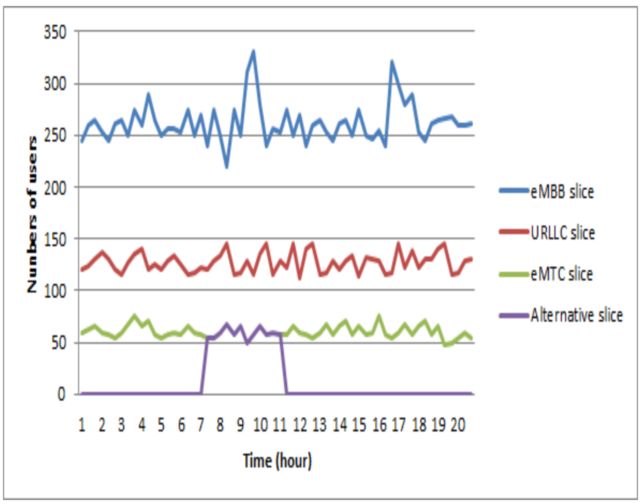


FIGURE 14. mMTC Slice failure and redirection to alternative slice.

• Precision: specifies the number of positive class predictions that truly belong to the positive class.

• F1- score: it is a valuable score that balances the precision and recall values, the larger F1-score means that the implemented approach is more efficient.

From table 4, we can see that the overall accuracy of MMSRNN model is 5% advanced than CNN. Fig.9 shows the network lifetime of the proposed model as compared with CNN. It is clear that the lifetime of MMSRNN model is better than CNN model.

About three hundred thousand requests were generated in 20 hours simulation that consists of 30% URLLC, 45% eMBB and 25% mMTC. Fig.10 shows the simulated model run for 20 hours.

The IMSNS model realized load balancing by identifying overloading case and automatically routed all the incoming requests to the Alternative slice when the number of connections exceeds a threshold which is 95% usage in our case. In the simulation, eMBB slice would be over utilized. IMSNS system model identified the overloaded slice and automatically redirected the new eMBB incoming requests to the alternative slice as specified in Fig.11. Compared to Fig.10, the overloaded traffic is handled by the alternative slice as shown in Fig.12.

Fig.13 specifies another scenario where the mMTC slice failed from 4hr to 7 hr; the proposed model redirected all incoming requests that belong to the failed slice to the alternative slice to avoid any packet loss. Fig.14 shows the handling of the mMTC slice failure. However, the connection can be lost due to sudden slice failure. This is recorded by the system especially the time and date of the slice failure to avert loss of all connections in the next time.

VI. CONCLUSION

We have proposed IMSNS system for 5G Network Slicing, which is an effective way to divide resources based on particular applications in order to provide QoS guarantees on a per-service basis. We have proposed two intelligent SDN controllers for accurate predicting the appropriate network slice with handling network load balancing and network slice failure conditions. The simulation results proved that the performance metrics which are Accuracy, Precision, Recall, F1-score are enhanced in IMSNS based on MMSRNN learning model. Through the full analysis, we could conclude that the proposed MMSRNN is more accurate and faster than CNN. This is largely owing to the fact that only neurons which exceed the threshold value would update their weights.

REFERENCES

[1] S. Khan, S. Khan, Y. Ali, M. Khalid, Z. Ullah, and S. Mumtaz, ''Highly accurate and reliable wireless network slicing in 5th generation networks: A hybrid deep learning approach,'' *J. Netw. Syst. Manage.*, vol. 30, no. 2, pp. 1–22, Apr. 2022.

[2] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed, W. Ameen, and T. R. Gadekallu, ''Optimal 5G network slicing using machine learning and deep learning concepts,'' *Comput. Standards Interfaces*, vol. 76, Jun. 2021, Art. no. 103518.

[3] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, ''Deep reinforcement learning for resource management in network slicing,'' *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[4] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, ''DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks,'' in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 0762–0767.

[5] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, ''Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1709–1717.

[6] F. Xie, D. Wei, and Z. Wang, ''Traffic analysis for 5G network slice based on machine learning,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–15, Dec. 2021.

[7] N. Hu, F. Luan, X. Tian, and C. Wu, ''A novel SDN-based application-awareness mechanism by using deep learning,'' *IEEE Access*, vol. 8, pp. 160921–160930, 2020.

[8] A. Malik, R. de Fréin, M. Al-Zeyadi, and J. Andreu-Perez, ''Intelligent SDN traffic classification using deep learning: Deep-SDN,'' in *Proc. 2nd Int. Conf. Comput. Commun. Internet (ICCCI)*, Jun. 2020, pp. 184–189.

[9] A. Mondal and S. Misra, ''FlowMan: QoS-aware dynamic data flow management in software-defined networks,'' *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1366–1373, Jul. 2020.

[10] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, ''AMPS: Application aware multipath flow routing using machine learning in SDN,'' in *Proc. 23rd Nat. Conf. Commun. (NCC)*, Mar. 2017, pp. 1–6.

[11] A. I. Owusu and A. Nayak, ''An intelligent traffic classification in SDN-IoT: A machine learning approach,'' in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, May 2020, pp. 1–6.

[12] J. Xu, J. Wang, Q. Qi, H. Sun, and B. He, ''Deep neural networks for application awareness in SDN-based network,'' in *Proc. IEEE 28th Int. Workshop Mach. Learn. for Signal Process. (MLSP)*, Sep. 2018, pp. 1–6.

[13] C. Roy, R. Saha, S. Misra, and D. Niyato, ''Soft-health: Software-defined fog architecture for IoT applications in healthcare,'' *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2455–2462, Feb. 2022.

[14] M. Hayes, B. Ng, A. Pekar, and W. K. G. Seah, ''Scalable architecture for SDN traffic classification,'' *IEEE Syst. J.*, vol. 12, no. 4, pp. 3203–3214, Dec. 2018.

[15] S. D. A. Shah, M. A. Gregory, and S. Li, ''Cloud-native network slicing using software defined networking based multi-access edge computing: A survey,'' *IEEE Access*, vol. 9, pp. 10903–10924, 2021.

[16] F. Naeem, G. Srivastava, and M. Tariq, ''A software defined network based fuzzy normalized neural adaptive multipath congestion control for the Internet of Things,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2155–2164, Oct. 2020.

[17] A. J. Ramadhan, ''T-S3RA: Traffic-aware scheduling for secure slicing and resource allocation in SDN/NFV enabled 5G networks,'' 2021, *arXiv:2107.05056*.

[18] T. Liu, Z. Liu, F. Lin, Y. Jin, G. Quan, and W. Wen, ''MT-Spike: A multilayer time-based spiking neuromorphic architecture with temporal error backpropagation,'' in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 450–457.

[19] A. D. Majeed and N. A. S. Al-Jamali, ''Spike neural network as a controller in SDN network,'' *J. Eng.*, vol. 27, no. 9, pp. 64–77, Sep. 2021.

[20] N. A. S. Al-Jamali and H. S. Al-Raweshidy, ''Intelligent traffic management and load balance based on spike ISDN-IoT,'' *IEEE Syst. J.*, vol. 15, no. 2, pp. 1640–1651, Jun. 2021.

[21] C. Shi, T. Wang, J. He, J. Zhang, L. Liu, and N. Wu, ''DeepTempo: A hardware-friendly direct feedback alignment multi-layer tempotron learning rule for deep spiking neural networks,'' *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 5, pp. 1581–1585, May 2021.

[22] N. A. S. Al-Jamali and H. S. Al-Raweshidy, ''Modified Elman spike neural network for identification and control of dynamic system,'' *IEEE Access*, vol. 8, pp. 61246–61254, 2020.

[23] Y. Xu, X. Zeng, L. Han, and J. Yang, ''A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks,'' *Neural Netw.*, vol. 43, pp. 99–113, Jul. 2013.

[24] B. E. Yellakuor, A. A. Moses, Q. Zhen, O. E. Olaosebikan, and Z. Qin, ''A multi-spiking neural network learning model for data classification,'' *IEEE Access*, vol. 8, pp. 72360–72371, 2020.

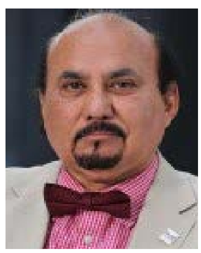
[25] N. A. S. Aljamali, ''Convolutional multi-spike neural network as intelligent system prediction for control systems,'' *J. Eng.*, vol. 26, no. 11, pp. 184–194, Nov. 2020.

[26] R. Hu, S. Chang, H. Wang, J. He, and Q. Huang, ''Efficient multispike learning for spiking neural networks using probability-modulated timing method,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1984–1997, Jul. 2019.

[27] X. Lin and G. Shi, ''A supervised multi-spike learning algorithm for recurrent spiking neural networks,'' in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 222–234.

NAJWAN SATTAR SOUD received the B.Sc. degree in computer engineering from the University of Baghdad, in 2004, where she is currently pursuing the M.Sc. degree with the Department of Computer Engineering. Her research interests include computer control, intelligent systems, and wireless networks.

NADIA ADNAN SHILTAGH AL-JAMALI (Member, IEEE) received the B.Sc. degree in control and systems engineering, the M.Sc. degree in control engineering, and the PhD. degree in computer engineering from the University of Technology, Baghdad, Iraq. Her research interests include computer control, wireless sensor networks, intelligent systems, neural networks, and robotics.



HAMED S. AL-RAWESHIDY (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the Postgraduate Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991, all in electronic engineering. He was at the Space and Astronomy Research Center, Baghdad, Perkin Elmer, Waltham, British Telecom, Oxford University, Manchester Metropolitan University, and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network and Communications Centre, Brunel University London, U.K.