University of Baghdad
College of Engineering
JOURNAL OF ENGINEERING

Journal of Engineering

journal homepage: www.joe.uobaghdad.edu.iq

Volume 29 Number 1 January 2023

Intelligent Congestion Control of 5G Traffic in SDN using Dual-Spike Neural Network

Najwan Sattar Soud*
M.sc student
University of Baghdad
Baghdad-Iraq
Najwan.saod2005M@coeng.uobaghdad.edu.iq

Nadia Adnan Shiltagh Al-Jamali
Prof. Dr.
University of Baghdad
Baghdad-Iraq
nadia.aljamali@coeng.uobaghdad.edu.iq

ABSTRACT

Software Defined Networking (SDN) with centralized control provides a global view and achieves efficient network resources management. However, using centralized controllers has several limitations related to scalability and performance, especially with the exponential growth of 5G communication. This paper proposes a novel traffic scheduling algorithm to avoid congestion in the control plane. The Packet-In messages received from different 5G devices are classified into two classes: critical and non-critical 5G communication by adopting Dual-Spike Neural Networks (DSNN) classifier and implementing it on a Virtualized Network Function (VNF). Dual spikes identify each class to increase the reliability of the classification. Different metrics have been adopted to evaluate the proposed classifier's effectiveness: accuracy, precision, recall, Matthews Correlation Coefficient (MCC), and F1-Score. Compared with a convolutional neural network (CNN), the simulation results confirmed that the DSNN model could enhance traffic classification accuracy by 5%. The efficiency of the priority model also has been demonstrated in terms of Round Trip Time (RTT).

Keywords: 5G Communications, Software Defined Networking, Virtualized Network Function, Intelligent spike neural network.

التحكم الذكي في الازدحام لحركة مرور 5G باستخدام الشبكة العصبية المزدوجة في الشبكة المعرفة برمجيا

نجوان ستار سعود*
طالبة ماجستير
جامعة بغداد كلية الهندسة

نادية عدنان شلتاغ الجمالي
أستاذ دكتور
جامعة بغداد كلية الهندسة

*Corresponding author

Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2023.01.07

Article received: 26/7/2022
Article accepted: 11/9/2022
Article published: 1/1/2023

الخلاصة

توفر الشبكات المعرفة بالبرمجيات (SDN) مع التحكم المركزي رؤية شاملة وتحقق إدارة فعالة لموارد الشبكة. ومع ذلك ، فإن استخدام وحدات التحكم المركزية له العديد من القيود المتعلقة بقابلية التوسع والأداء خاصة مع النمو المتسارع لاتصالات 5G . في هذا البحث ، تم اقتراح خوارزمية جديدة لجدولة حركة المرور لتجنب الازدحام في مستوى التحكم. يتم تصنيف رسائل Packet In المستلمة من أجهزة 5G المختلفة إلى فئات: اتصالات 5G الحرجة وغير الحرجة من خلال اعتماد الشبكة العصبيه المزدوجة (DSNN) **وتنفيذها على** وظيفة شبكة افتراضية (VNF). يتم تحديد كل فئة بواسطة طفرات مزدوجة لزيادة موثوقية التصنيف. تم اعتماد مقاييس مختلفة لتقييم كفاءة المصنف المقترح وهي: الدقة (مدى قرب القيمة من قيمتها الحقيقية) ، الدقة (مدى تكرار القياس) ، الاسترجاع ، معامل ارتباط ماثيوز (MCC) ودرجة F1. بالمقارنة مع الشبكة العصبية التلافيفية (CNN) ، أظهرت نتائج المحاكاة أن النموذج المقترح يمكن أن يعزز دقة تصنيف حركة المرور بنسبة 5٪. تم توضيح كفاءة نموذج الأولوية أيضًا من حيث وقت الرحلة ذهابًا وإيابًا (RTT).

الكلمات الرئيسية: اتصالات 5G ، الشبكات المعرفة بالبرمجيات ، وظيفة الشبكة الافتراضية ، الشبكة العصبية الذكية.

1. INTRODUCTION

With the ascending diversity of 5G services, each type of these services requires specified reliability, latency, transmission rate, etc. 5G services involve smart cities, smartphones, remote surgery, wearable devices, public safety, etc. **(Preciado-Velasco, et al., 2021)**. Different types of 5G services have different requirements; for instance, the latency metric, which is about 10 ms for non-critical communication such as smartphone use cases and about 1ms for critical use cases like healthcare and public safety (**Khan, et al., 2022**). Conventional networks do not have this capability. To fill this gap, SDN offers a promising technology that allows completely disconnecting the control layer from the data layer. However, the centralized control plane has some disadvantages in scalability and reliability. Due to the different network resource requirements of various types of 5G traffic, a precise classification algorithm is essential to realize traffic identification **(Kamath, et al., 2019)**. SDN platform provides an opportunity to adopt ML because of the central control of the network **(Mondal and Misra, 2020)** and **(Malik, et al., 2020)**. ML is a splendid choice for network analysis and automation of functions **(Morocho-Cayamcela, et al., 2019)** and **(Soud, et al., 2022).**

This paper aimed to eliminate the congestion in the centralized SDN controller and decrease the latency, especially for critical 5G communication. The enormous computation and storage requirements related to Deep learning challenge their implementation with limited network resources **(Liu, et al., 2017)**. Spiking Neural Networks (SNN) with time-based coding improves energy efficiency. It works with discrete events at defined times **(Al-Jamali, 2020), (Shi, et al., 2021),** (**Miao, et al., 2018),** and **(Shiltagh and Abas, 201).**

So that, efficient prediction algorithm called Dual-Spike Neural Networks (DSNN) classifier on Virtualized Network Function (VNF) was implemented to prioritize and schedule 5G Packet-In messages, which are forwarded to the SDN controller to create a suitable flow rule because these packets do not have flow rules in the flow tables of the switches.

(Malik, et al., 2020) implemented a new deep learning model called Deep-SDN to accurately identify the network traffic types in a short time, so it is suitable for identifying the online traffic. The simulation results confirmed that the Deep-SDN reported overall accuracy of 96%. (**Hu, et al., 2020)** proposed a CNN-based deep learning model in an SDN environment to realize the application awareness using different CNN structures; this mechanism exceeds three standards on recall ratio, stability, precision ratio, and F value.

(Raikar, et al., 2020) proposed different machine learning models, which are Support Vector Machine (SVM), nearest centroid, and Naïve Bayes (NB), to classify the network traffic. The accuracy obtained for NB is 96.79%, SVM is 92.3%, and the nearest centroid is 91.02%. **(Owusu and Nayak, 2020)** compared the efficiency of three ML algorithms: Random Forest algorithm, decision tree algorithm, and the K-Nearest Neighbor (KNN) algorithm for IOT traffic classification in SDN. The results proved that the Random Forest method achieved better accuracy of 83% with a reduced number of features.
The above works could implement the SDN-based traffic classification. However, these researches do not consider the high exhaustion of computing resources of the centralized controller. In this work, we propose a systematic framework to predict the priority of 5G services in order to forward them to the SDN controller. A new training algorithm with time-based coding called DSNN is proposed to achieve better results in power consumption, computation speed, and energy efficiency by updating the weights and threshold.
The rest of this paper is regulated as follows: Section 2 presents the proposed system model; section 3 specifies the training algorithm of the proposed DSNN. In section 4, the simulation and evaluation performance is explained. Section 6 concludes this paper.

2. THE PROPOSED SYSTEM MODEL

To prevent congestion at the SDN controller, a novel traffic scheduling algorithm was implemented in which the traffic received from different 5G devices is classified into two classes:

- High-priority traffic represents critical communication services such as healthcare, public safety, factory automation, self-driving car, etc.
- Low-priority traffic refers to non-critical communication services like smartphones, smart wearables, etc.

DSNN learning algorithm is proposed in this paper and adopted in the SDN environment's data plane. Packets to be classified will be sent to the DSNN classifier implemented at VNF standalone server to realize application-aware and prioritize the 5G traffic based on extracted features from both the network and the devices called Key Parameter Indicators (KPIs). Then, the traffic is scheduled. High-priority traffic will be sent first to the SDN controller. If two messages have the same priority, they will be based on First-In-First-Out (FIFO) technique.
The architecture of the system model is shown in **Fig. 1**.

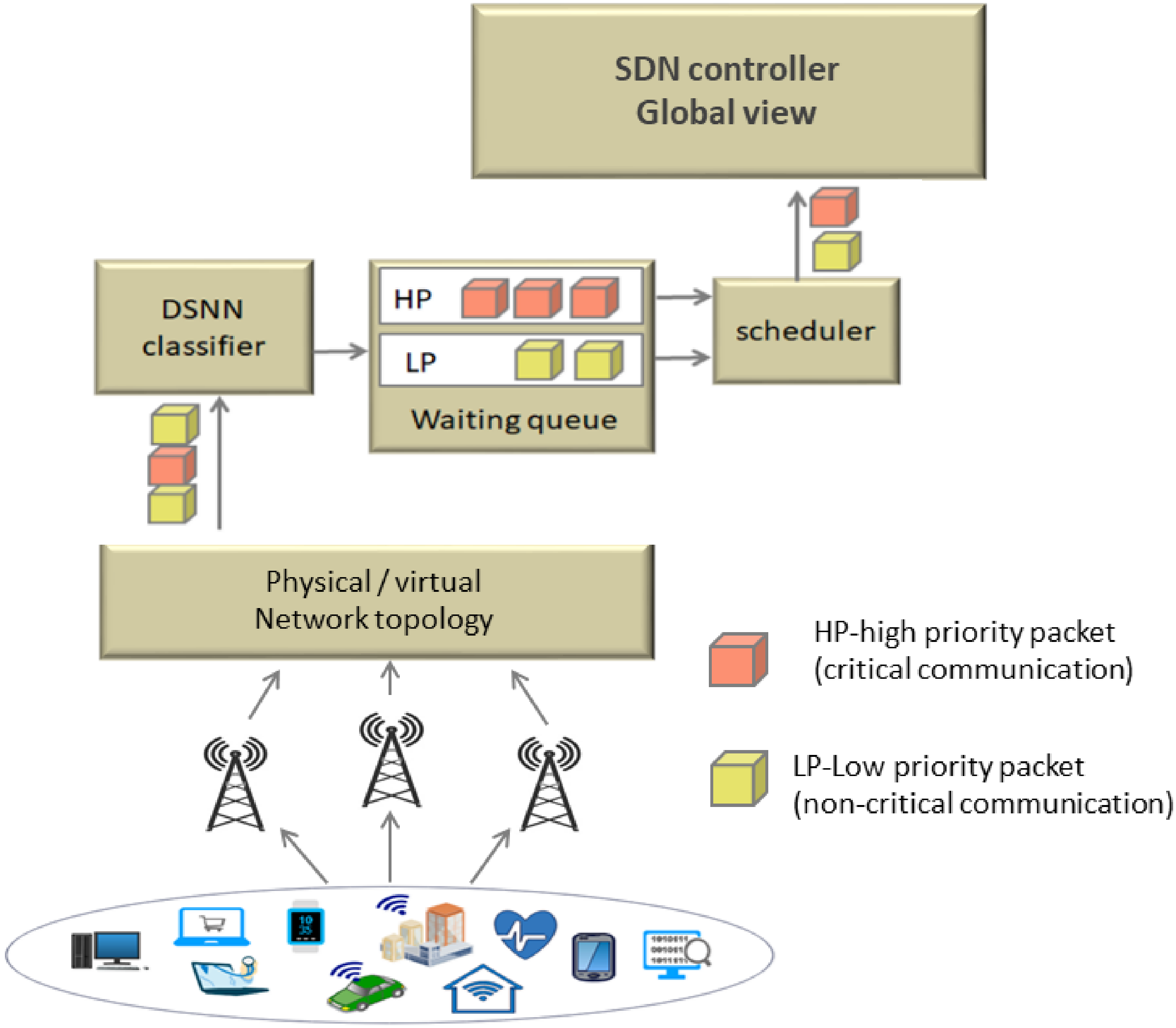

Figure 1. The proposed system architecture

3. DSRNN TRAINING ALGORITHM

The initial structure of the DNSS model has three fully connected layers with 17, 80, and 3 neurons, respectively, as shown in **Fig. 2**. DNSS learning algorithm updated only the weights of a link that reached the threshold as a base condition of emitting spikes. So that many connections and neurons have been canceled from the initial structure, resulting in low power consumption and high speed of computations. Dual spikes identify each class to increase the reliability of the classification **(Taherkhani, et al., 2019)**. Six delayed synapses (S) are used to connect the presynaptic neuron to a postsynaptic neuron, as shown in **Fig. 3.**

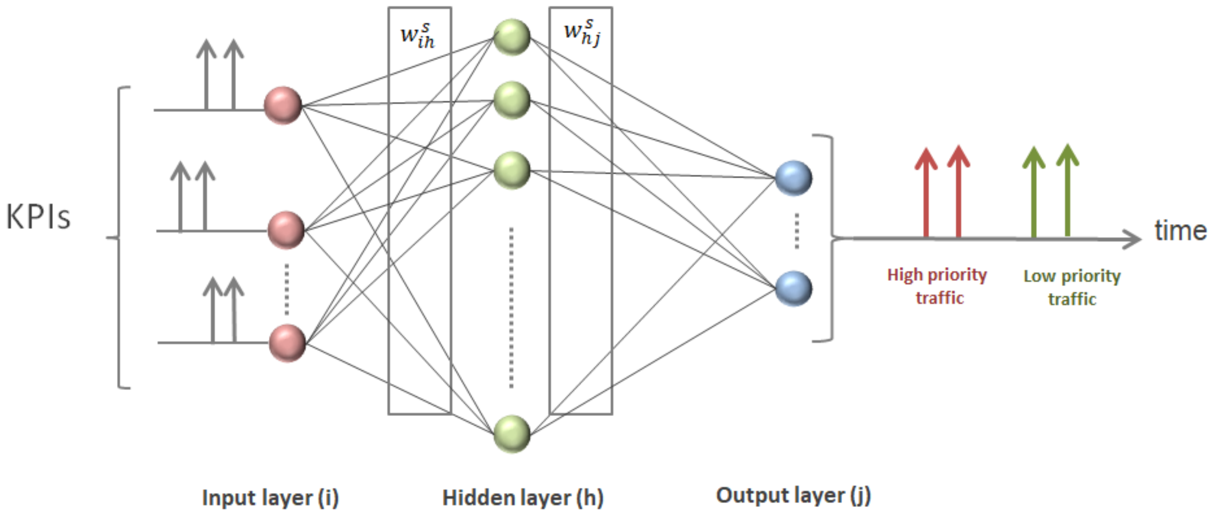

Figure 2. Structure of DSNN

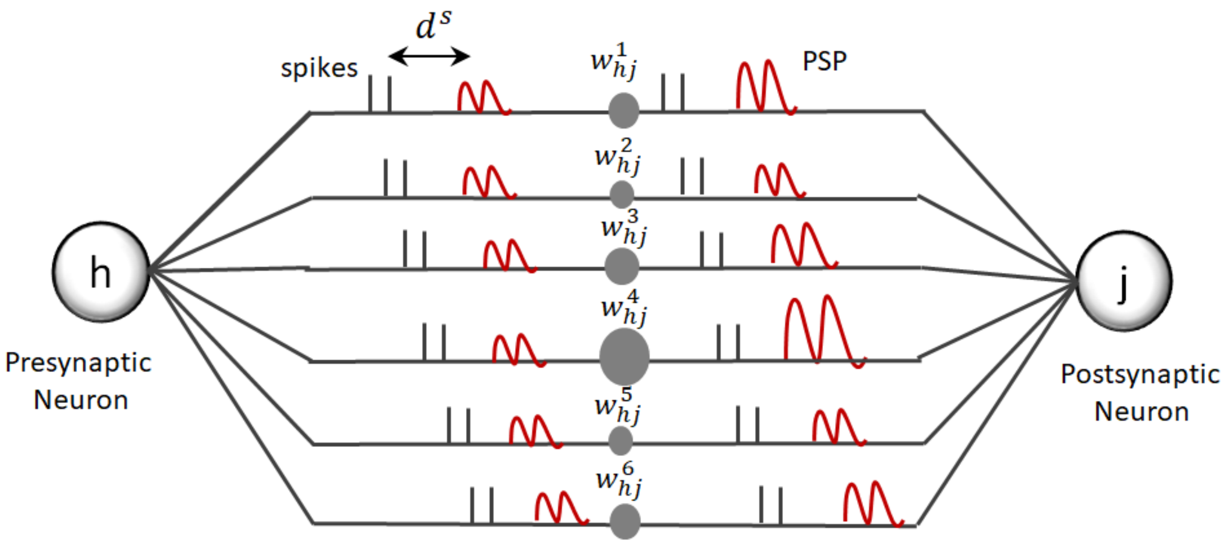

Figure 3. Sub-connection between presynaptic neuron and postsynaptic neuron

The training algorithm is based on the time encoding technique to achieve striking drooping in energy consumption. The real values of the selected features are converted to spike times as expressed in Eq. (1).

$$t_i^f = t_{max} \llcorner \frac{t_{min}(f_s(t) - f_{s_{min}})(t_{max} - t_{min})}{(f_{s_{max}} - f_{s_{min}})} \urcorner \quad (1)$$

Where, $f_{s_{max}}$ and $f_{s_{min}}$ refer to the maximum and minimum value of the selected feature, f_s is the current value of the selected feature. While t_{max} and t_{min} are the maximum and minimum interval time, respectively. The symbol $\llcorner\urcorner$ refers to the round function.

Decoding equation will be used to convert the predicted priority traffic from time to real value of priority (P) using the Eq. (2).

$$P(t_j) = \frac{(t_{max} - t_j - t_{min})(f_{s_{max}} - f_{s_{min}})}{(t_{max} - t_{min})} + f_{s_{min}} \quad (2)$$

The value of t_j denotes the output spike time.

After the implementation of the encoding process, the Spike Response Function (SRM) neuron model is adopted. At the arrival of each spike, each neuron in the hidden layer is checked if it is spiked or not. Spike response function $\epsilon(t)$ is used to compute the postsynaptic potential (PSP) as explained in Eq. (3).

$$\epsilon(t) = \begin{cases} \frac{t}{\tau_S} e^{1-\frac{t}{\tau_S}} & , \quad t > 0 \\ 0 & , \quad t \leq 0 \end{cases} \quad (3)$$

Where τ_s refers to the time decay constant of the spike response function. It is noteworthy; that the sum of the PSPs represents the Membrane Potential (MP) of the neuron, also the value of the weights of synapses that transmit these input spikes affect the PSP. During the time interval of DSNN learning, the neuron would be trained to produce two spikes when its membrane potential exceeds threshold value θ. Then, the MP decreases to zero, and the neuron cannot fire. This phase is named the Repolarization phase. MP has remained at a value less than 0, and the neuron cannot emit a spike again for a period. This phase is named the Hyperpolarization phase.

With DSNN learning, MP is not only induced by the arrived spikes from all presynaptic neurons as such single spiking neural network but also influenced by the Repolarization and Hyperpolarization phases in order to protect the neuron from damage.

As a result for that, for multi-spike neural networks, the expression of membrane potential includes the refractoriness function term $\zeta(t - t^{mrs})$, t^{mrs} value denotes the time of the most recent output spike. The function ζ (t) is explained in Eq. (4).

$$\zeta(t) = \begin{cases} -2\theta e^{\frac{-t}{\tau_r}} & , \quad t > 0 \\ 0 & , \quad t \leq 0 \end{cases} \qquad (4)$$

Where τ_r represent the time decay constant of the refractoriness function. Accordingly, Eq. (5) represents the general expression of the MP function.

$$m_j(t) = \sum_{i=1}^{N_i} \sum_{s=1}^{S} \sum_{\substack{t_i^f \in F_i \\ t_i^f + d^s > t^{mrs} + R_p}} w_{ij}^s \epsilon\left(t - t_i^f - d^s\right) + \zeta(t - t^{mrs}) \qquad (5)$$

Where N_i *is* the number of presynaptic neurons, w_{ij}^s is the weight between presynaptic neuron and postsynaptic neuron, and $\epsilon\left(t - t_i^f - d^s\right)$ denotes the spike response function. (S) refers to the number of synapses used to connect the presynaptic neuron to the postsynaptic neuron with different transmit delays (d^s) and weights. F_i refers to the number of emitted spikes by the presynaptic neuron. Rp is the length of the Repolarization period.

Mean Square Error (MSE) is used to evaluate the learning algorithm's performance

$$E = \frac{1}{2} \sum_{j=1}^{N_j} \sum_{f=1}^{F_j} (t_j^f - t_j^{\hat{f}})^2 \qquad (6)$$

Where t_j^f and $t_j^{\hat{f}}$ values refer to actual and target spike time, respectively. N_j refers to the number of neurons in the output layer, F_j denotes the number of the output spikes.

The backpropagation phase begins from the output layer to the input layer after the expiry of T interval in order to reduce the error value. The weights will be updated based on gradient descent **(Xu, et al., 2013)** between j_{th} neuron in the output layer and h_{th} neuron in the hidden layer as shown from Eq. (7) to Eq. (12).

$$w_{hj}^s(t+1) = w_{hj}^s(t) + \Delta w_{hj}^s \qquad (7)$$

$$\Delta w_{hj}^{s} = -\alpha \nabla E_{hj}^{S} \tag{8}$$

Where, α is the learning rate. w_{hj}^{s} refers to the weight between the j_{th} a neuron at the output layer and the h_{th} a neuron at the hidden layer.

$$\nabla E_{hj}^{S} = \frac{\partial E}{\partial w_{hj}^{s}} = \frac{1}{2}\sum_{f=1}^{F_j} \frac{\partial(\sum_{j=1}^{N_j}(t_j^{(f)} - t_j^{\wedge(f)})^2)}{\partial w_{hj}^{s}} \tag{9}$$

$$= \frac{1}{2}\sum_{f=1}^{F_j} \frac{\partial(\sum_{j=1}^{N_j}(t_j^{(f)} - t_j^{\wedge(f)})^2)}{\partial t_j^{(f)}} \frac{\partial t_j^{(f)}}{\partial w_{hj}^{s}}$$

$$= \sum_{f=1}^{F_j} \frac{\partial(\frac{1}{2}\sum_{j=1}^{N_j}\left(t_j^{(f)} - t_j^{\wedge(f)}\right)^2)}{\partial t_j^{(f)}} \frac{\partial t_j^{(f)}}{\partial w_{hj}^{s}}$$

$$= \sum_{f=1}^{F_j} \frac{\partial E}{\partial t_j^{(f)}} \frac{\partial t_j^{(f)}}{\partial w_{hj}^{s}} \tag{10}$$

where,

$$\frac{\partial E}{\partial t_j^{(f)}} = \frac{1}{2} \frac{\partial(\sum_{j=1}^{N_j}\sum_{f=1}^{F_j}(t_j^{(f)} - t_j^{\wedge(f)})^2)}{\partial t_j^{(f)}}$$

$$= \frac{1}{2} \frac{\partial((t_j^{(f)} - t_j^{\wedge(f)})^2}{\partial t_j^{(f)}} = t_j^{(f)} - t_j^{\wedge(f)} \tag{11}$$

As it was mentioned, DSNN is a modified type of multi-spike neural network based on time coding, where dual spikes are used to identify each class. Correspondingly, the $(\frac{\partial t_j^{(f)}}{\partial w_{hj}^{s}})$ term should be analyzed from t^1 to t^f because of the refractoriness function and effect of the most recent output spike using the recursive equation, as illustrated in Eq. (12).

$$\frac{\partial t_j^{(f)}}{\partial w_{hj}^{s}} = \frac{\partial t_j^{(f)}}{\partial m_j(t_j^{(f)})}\left(\frac{\partial m_j(t_j^{(f)})}{\partial w_{hj}^{s}} + \frac{\partial m_j(t_j^{(f)})}{\partial t_j^{(f-1)}} \frac{\partial t_j^{(f-1)}}{\partial w_{hj}^{s}}\right) \tag{12}$$

The same equations are implemented between h_{th} neuron in the hidden layer and i_{th} neuron in the input layer as explained in Eq. (13) and Eq. (14) with the same details as Eq. (7) to Eq. (12)

$$w_{ih}^{s}(t+1) = w_{ih}^{s}(t) + \Delta w_{ih}^{s}(t) \quad (13)$$

$$\Delta w_{ih}^{s} = -\alpha \nabla E_{ih}^{s} \quad (14)$$

Fig. 4 presents the Flowchart of the working mechanism

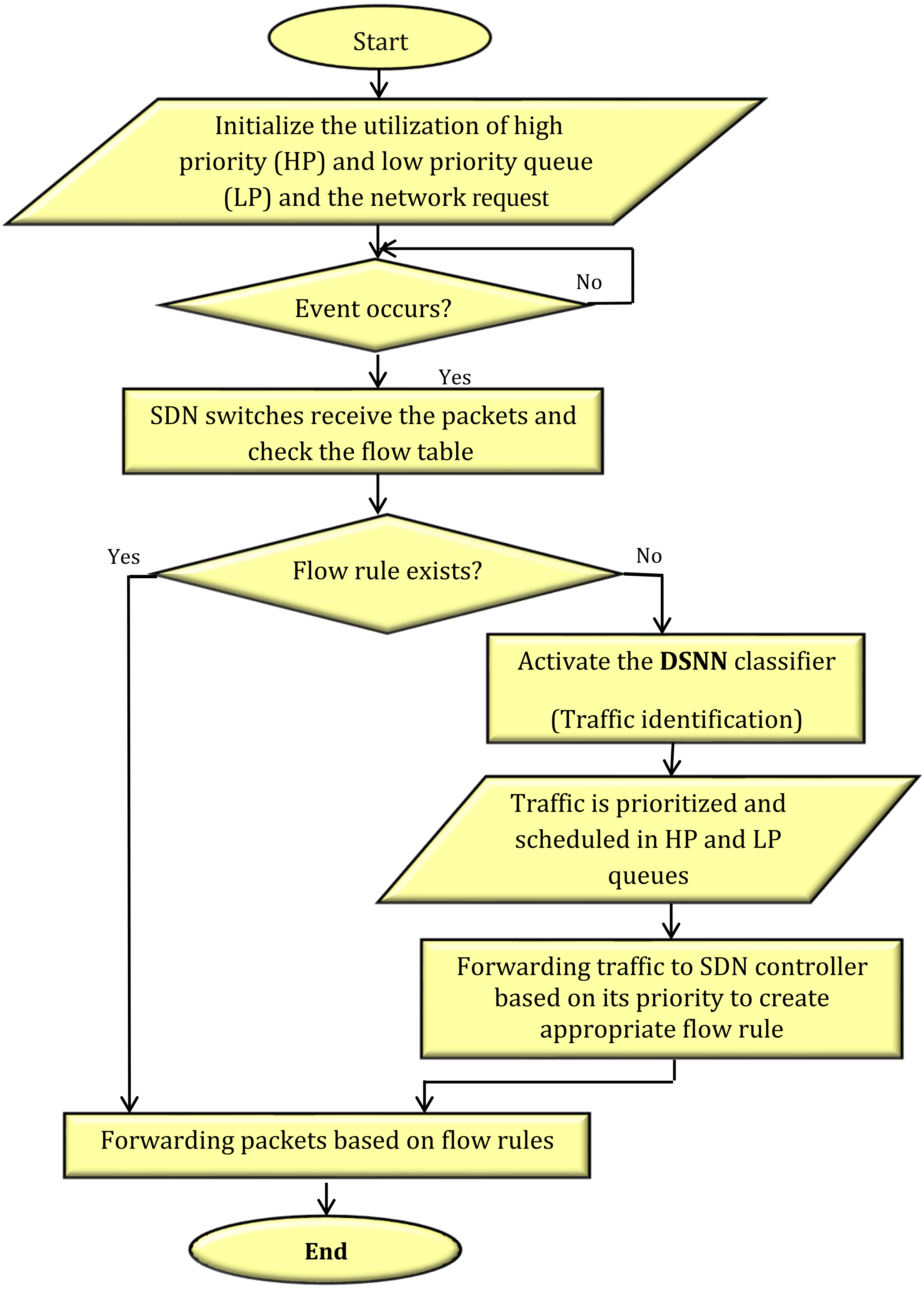

Figure 4. Flowchart of the proposed system model

4. EVALUATION PERFORMANCE

The proposed system includes two parts: the first is for SDN-based traffic collection based on the NS2 simulator, and the second is DSNN for application awareness based on MATLAB 2018a. The collected dataset contains 10000 unique items **(Thantharate, et al., 2019)**. 70% of the dataset was used for training, and 30% was used for testing. Four values derived from the confusion matrix have been used to evaluate the trained model performance, as shown in **Table 1.**

Table 1. Confusion matrix structure

<table>
<tr><td rowspan="4">Real value</td><td colspan="3">Predictive value</td></tr>
<tr><td></td><td>Positive</td><td>Negative</td></tr>
<tr><td>Positive</td><td>TP</td><td>FN</td></tr>
<tr><td>Negative</td><td>FP</td><td>TN</td></tr>
</table>

True positive (TP): the number of values classified and the classification is correct.

- True Negative (TN): the number of values classified and the classification is wrong.
- False Positive (FP): the number of values classified to wrong class, and the model considers them as positive prediction, but the prediction is incorrect.
- False Negative (FN): the number of values classified to correct class, and the model considers them as negative prediction, but the prediction is correct.

To measure the performance of the proposed model, a set of metrics can be used to evaluate the efficiency of the classification model as follows:

- Accuracy: refers to the relationship between the number of right predictions and the total number of predictions. Accuracy is defined in Eq. (15).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \qquad (15)$$

- Precision: refers to the relationship between positive class predictions that rightly join with the positive class. The Precision value is determined by Eq. 16.

$$Precision = \frac{TP}{TP+FP} * 100\% \qquad (16)$$

- Recall: refers to the relationship between the positive predictions to the all positive elements in the data set. It is computed by Eq. (17).

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (17)$$

- F1- score: refers to the balance average between the precision and recall values. The used model is more efficient if the F1-score value is large. This value is determined by Eq. (18).

$$F1\text{- } score = \frac{(2*Recall*Precision)}{Precision+Recall} \quad (18)$$

- Matthews correlation coefficient (MCC): is an evaluation metric that is more complicated than other metrics. The MCC value is between +1 and -1. The value of +1 refers to perfect classification, while the value of -1 refers to misclassification. MCC is defined in Eq. (19).

$$\mathrm{MCC} = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (19)$$

The used parameters of each neuron in the DSNN are described in **Table2.** After that, the simulation was repeated using fully connected CNN, which has one Conv1D layer, pooling layer, fully connected layers with (100, 100) neurons, and an output layer with 3 neurons. ReLU (Rectified Linear Unit) activation function is used for convolution and fully connected layers. The efficiency of the DSNN model is compared against the CNN model using the same value of learning rate. **Fig.5** shows the error rate of the DSNN model as compared with the CNN model. It is obvious that the DSNN model can learn faster than the CNN model.

Table 2. Parameters of DSNN.

Parameters	Values
τ_s	12 ms
τ_r	75 ms
α	0.005
θ	1 v
Rp	1.5 ms

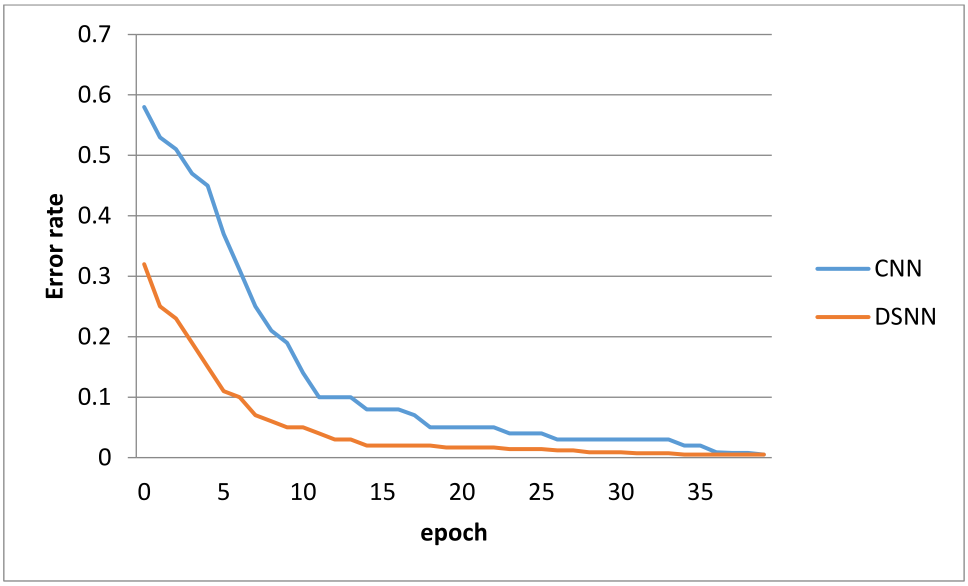

Figure5. The error rate of the DSNN model as compared with the CNN model.

The precision, accuracy, MCC, Recall, and F1-score measures are shown in **Fig. 6**, **Fig. 7**, **Fig. 8**, **Fig. 9,** and **Fig. 10,** respectively. DSNN training algorithm enhanced the results as compared with CNN in different performance metrics, and that is back to the architecture of DSNN, which has six delayed sub-connections where delays are an essential property of a real biological neural network. Furthermore, DSNN works with discrete events through precise timing of spikes, while CNN works with consistently changing events that require excessive computation and storage requirements. In addition, DSNN is a modified type of multi-spike neural network based on time coding. Dual spikes are used to identify each class, increasing the reliability of the classification process. So, the DSNN training algorithm achieved more accurate results than the CNN in different performance metrics (Accuracy, Precision, MCC, Recall, and F1-score).

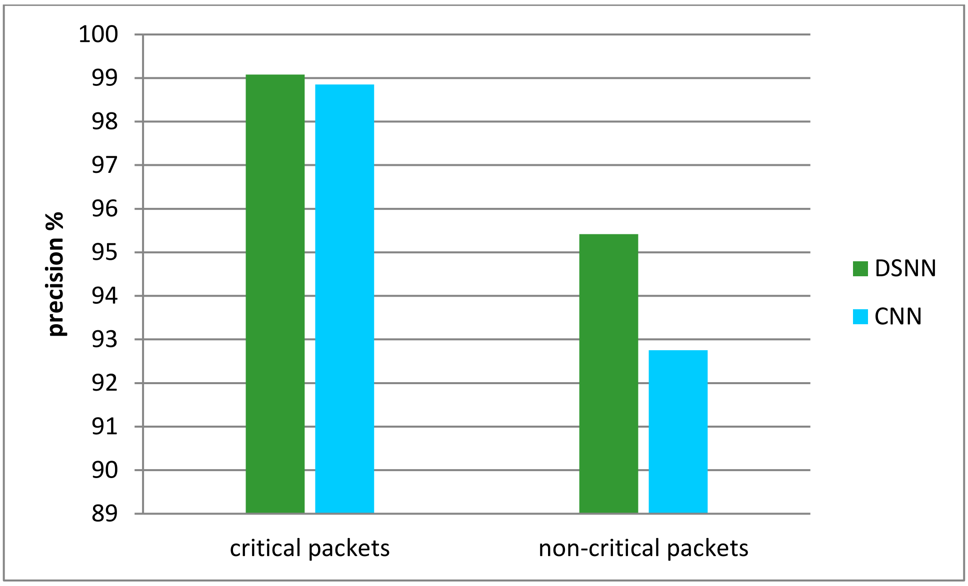

Figure 6. The precision ratio of the DSNN model and the CNN model.

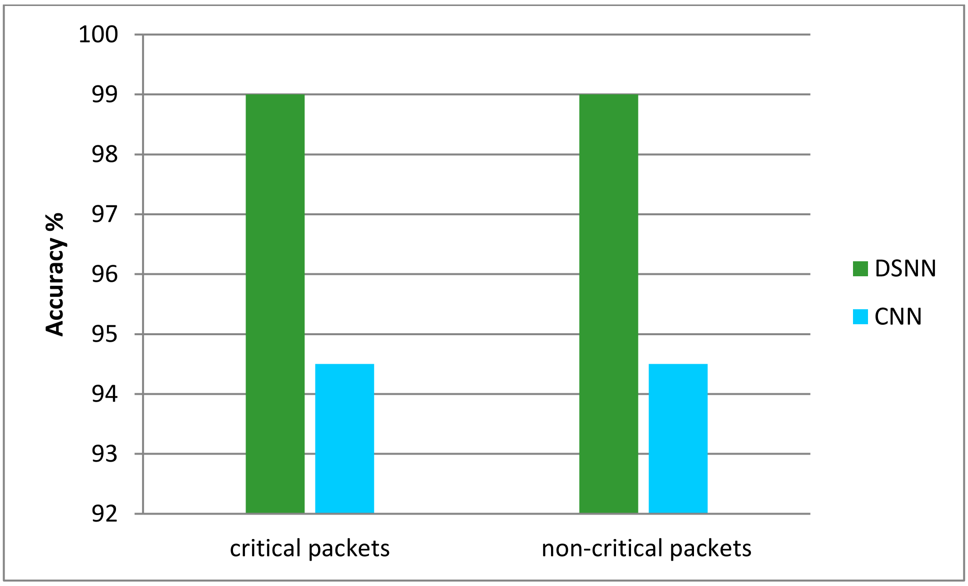

Figure 7. Accuracy ratio of the DSNN model and the CNN model.

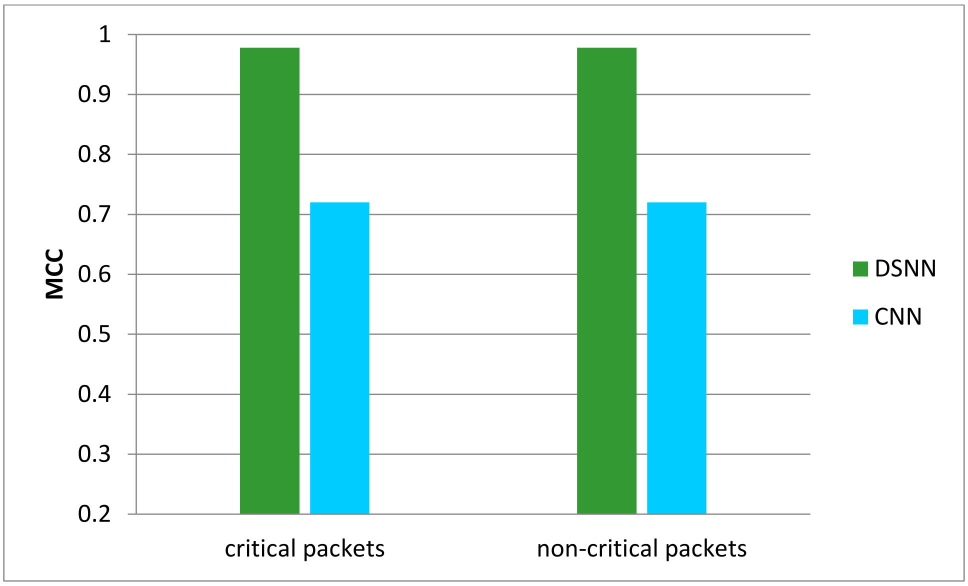

Figure 8. MCC of the DSNN model and the CNN model.

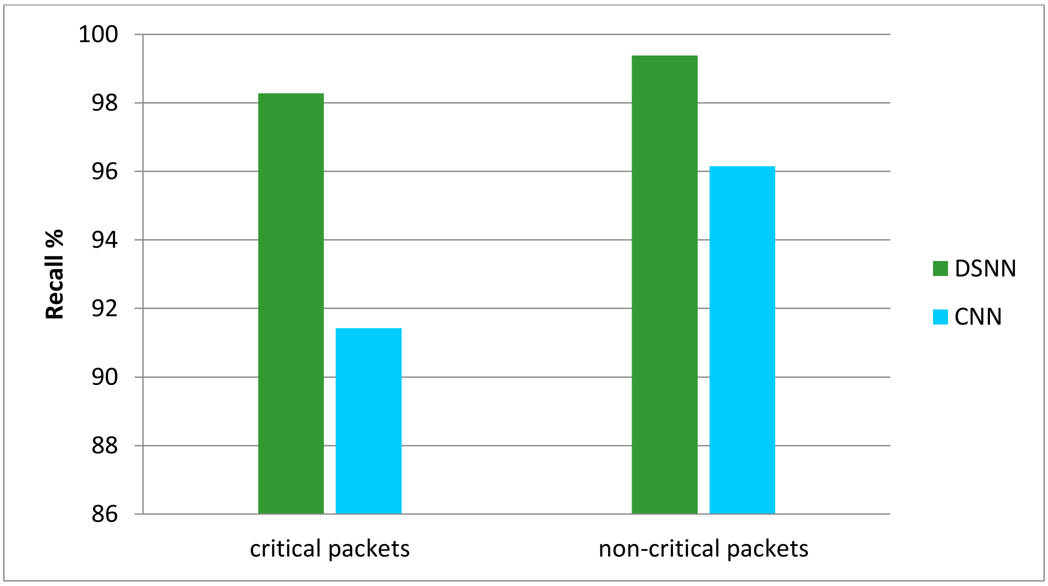

Figure 9. Recall ratio of the DSNN model and the CNN model.

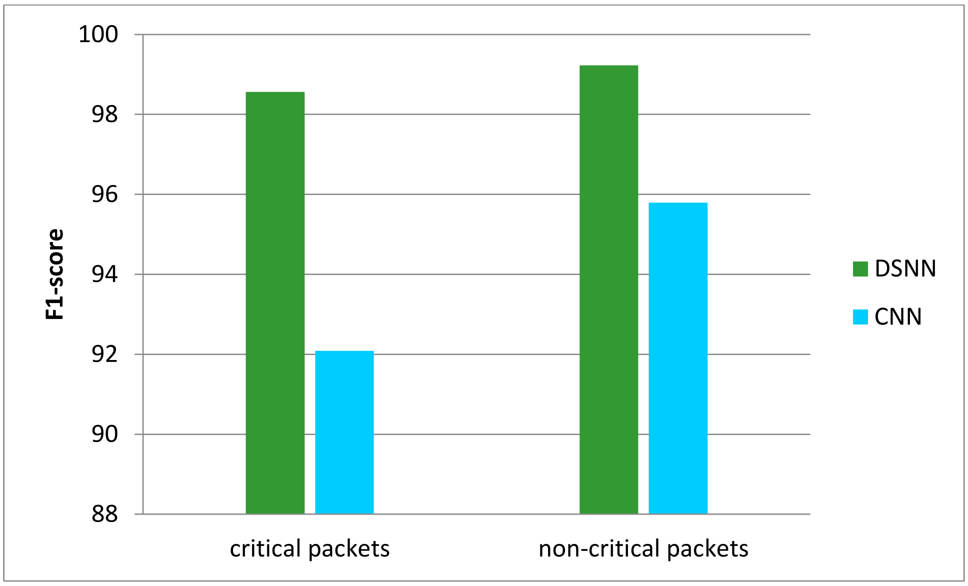

Figure 10. F1-score of the DSNN model as compared with the CNN model.

Fig. 11 compares Round Trip Time (RTT) between critical and non-critical communication. In detail, RTT is 6.82 ms in the case of critical traffic, while for non-critical traffic is 10.53 ms in the case of 1000 requests forwarded to the SDN controller. This happens because of classifying the 5G Packet-In messages into high priority and low priority traffic using the DSNN classifier. Then, these packets are handled by the SDN controller based on their priority.

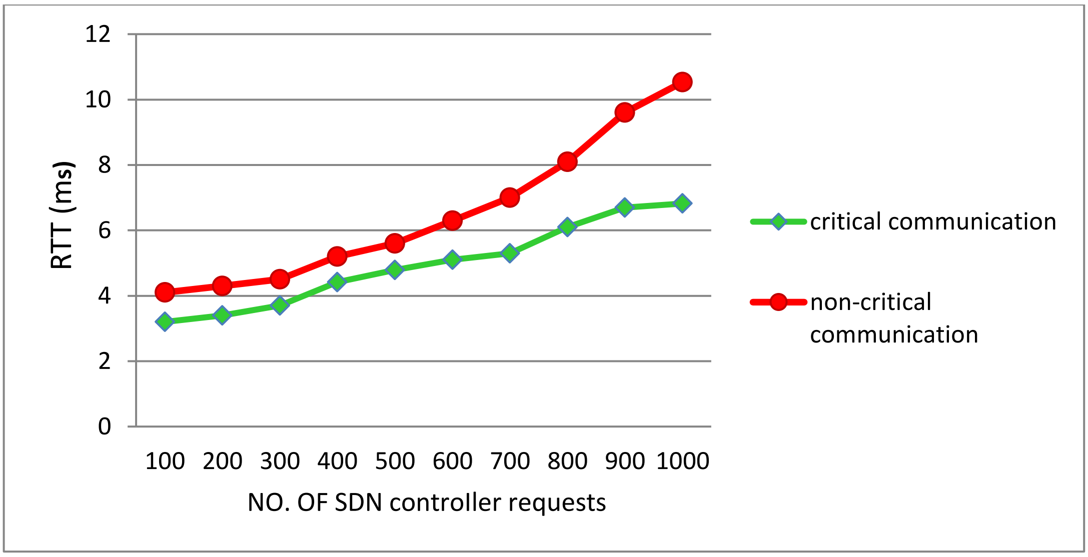

Figure 11. Round Trip Time for critical and non-critical 5G communication

CONCLUSIONS

In this paper, an efficient approach called DSNN is proposed for prioritizing and scheduling 5G traffic on a VNF in the data plane of the SDN platform. DSNN model eliminates congestion problems in the control plane to increase the network's reliability. The simulation results showed that the proposed classifier is effective in the Different metrics: precision, accuracy, recall, F1-Score, and MCC. Compared with the CNN classifier, the results confirmed that the proposed model outperformed the CNN model by 5%, which is back to the DSNN works with discrete events with time-based coding. In contrast, CNN works with consistently changing events that require costly computation and storage requirements.

Additionally, each class is identified by dual spikes increasing the reliability of the classification process. Furthermore, DSNN learned faster than CNN because not all the weights have been updated as with CNN. The DSNN updated only the weights of a link that reached the threshold. The efficiency of the DSNN priority model also has been demonstrated in terms of Round Trip Time.

REFERENCES

Al-Jamali, N. A. S., 2020. *Convolutional Multi-Spike Neural Network as Intelligent System Prediction for Control Systems, Journal of Engineering*, vol. 26, no. 11, pp. 184–194.

Hu, N., Luan, F., Tian, X., and Wu, C., 2020. A novel sdn-based application wareness mechanism by using deep learning, *IEEE Access,* vol. 8, pp. 160 921–160 930

Kamath, S., Singh, S., and Kumar, M.S., 2020. *Multiclass Queuing Network Modeling and Traffic Flow Analysis for SDN-Enabled Mobile Core Networks with Network Slicing, IEEE Access*, vol. 8, 8932489, pp. 417-430, https://doi.org/10.1109/ACCESS.2019.2959351.

Khan, S., Khan, S., Ali, Y., Khalid, M., Ullah, Z., and Mumtaz, S., 2022. Highly accurate and reliable wireless network slicing in 5th generation networks: A hybrid deep learning approach, *Journal of Network and Systems Management*, vol. 30, no. 2, pp. 1–22.

Liu, T., Liu, Z., Lin, F., Jin, Y., Quan, G., and Wen, W., 2017. Mt-spike: A multilayer time-based spiking neuromorphic architecture with temporal error back propagation, *IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE*, pp. 450–457.

Malik, A., de Fréin, R., Al-Zeyadi, M., and Andreu-Perez, J., 2020. Intelligent sdn traffic classification using deep learning: Deep-sdn, 2nd International Conference on Computer Communication and the Internet (ICCCI). IEEE, pp. 184–189.

Miao, Y., Tang, H. and Pan, G., 2018. A Supervised Multi-Spike Learning Algorithm for Spiking Neural Networks, Proceedings of the International Joint Conference on Neural Networks. IEEE, July, DOI: 10.1109/IJCNN.2018.8489175.

Mondal, S., and Misra, S., 2020. FlowMan: QoS-Aware Dynamic Data Flow Management in Software-Defined Networks, *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1366-1373, DOI: 10.1109/JSAC.2020.2999682.

Morocho-Cayamcela, M. E., Lee, H., and Lim, W., 2019. Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions, *IEEE Access*, 7. 137184-137206. 10.1109/ACCESS.2019.2942390.

Owusu, I. and Nayak, A., 2020 An Intelligent Traffic Classification in SDN-IoT: A Machine Learning Approach, IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1-6, DOI: 10.1109/BlackSeaCom48709.2020.9235019.

Preciado-Velasco, J.E., Gonzalez-Franco, J.D., Anias-Calderon, C.E., Nieto-Hipolito, J.I., and Rivera-Rodriguez, R., 2021. 5G/B5G Service Classification Using Supervised Learning, *Applied Sciences*, 11, 4942, https://doi.org/10.3390/app11114942.

Raikar, M. M., M, M. S., Mulla, M. M., Shetti, N. S., Karanandi, M., 2020. Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning, *Procedia Computer Science*, Volume 171, pp. 2750-2759.

Shi, C., Wang, T., He, J., Zhang, J., Liu, L., and Wu, N., 2021. Deeptempo: a hardware-friendly direct feedback alignment multi-layer tempotron learning rule for deep spiking neural networks, *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 68, no. 5, pp. 1581–1585.

Shiltagh, N. A. and Abas, H. A., 2014. Spiking Neural Network in Precision Agriculture, *Journal of Engineering*, 3(7), p. 17-34.

Soud, N. S., Al-Jamali, N. A. S., and Al-Raweshidy, H. S., 2022. Moderately Multispike Return Neural Network for SDN Accurate Traffic Awareness in Effective 5G Network Slicing, *IEEE Access*, vol. 10, no. June, pp. 73378–73387, doi:10.1109/ACCESS.2022.3189354.

Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., and McGinnity, T. M., 2020. A review of learning in biologically plausible spiking neural networks. Neural networks*, The Official Journal of the International Neural Network Society*, *122*, 253–272. https://doi.org/10.1016/j.neunet.2019.09.036.

Thantharate, A., Paropkari, R., V. Walunj, and Beard, C., 2019. Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5g networks, IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0762–0767.

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