Iraqi Journal of Science, 2025, Vol. 66, No. 11, pp: 5191- 5203
DOI: 10.24996/ijs.2025.66.11.39

ISSN: 0067-2904

5G Network Slice Prediction using Adaptive Neuro-Fuzzy Inference System

Yousif S. Alash*, Rana F. Ghani
Department of Computer Science, University of Technology, Iraq

Received: 24/6/2024 Accepted: 23/10/2024 Published: 30/11/2025

Abstract

Resource utilization in computer networks is a key area of research. Due to the growing numbers of communication devices and data being processed in the network, today's 5G networks are no exception. Network slicing provides a data-driven, programmable solution by enabling the use of virtual segments over the same underlying physical network, using techniques such as Software Defined Networking and Network Function Virtualization. This paper focuses on predicting the most suitable network slice for network tenants by utilizing the adaptive neuro-fuzzy inference system as a multiclassifier for incoming network slice requests. Each tenant is assigned a suitable slice based on the requesting device's type and characteristics of the required communication channel. The slices considered are massive machine-type communications, enhanced mobile broadband, and ultra-reliable low-latency communications. Our evaluation of the model and simulation results highlights the effectiveness of the Adaptive Neuro-Fuzzy Inference System in selecting the most suitable network slice type, achieving a prediction accuracy of 99.97% using unseen data with a total increase of bandwidth utilization of 12.6% during simulation. Displaying similar or superior performance to existing Deep Learning and Machine Learning approaches.

Keywords: 5G Cellular Networks, Artificial Intelligence, Machine Learning, Network Slicing, Neuro-Fuzzy Inference System.

التنبؤ بشرائح شبكة الجيل الخامس باستخدام نظام الاستدلال العصبي الضبابي التكيفي

يوسف سمير علش*, رنا فريد غني
قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة

يُعد توظيف موارد شبكات الحاسوب مجالًا رئيسيًا للبحث. ومع تزايد عدد أجهزة الاتصال وكميات البيانات المعالجة في الشبكات، فإن شبكات الجيل الخامس الحالية ليست استثناءً. يقدم تقطيع الشبكة حلاً برمجيًا قائمًا على البيانات من خلال تمكين استخدام القطاعات الافتراضية على نفس الشبكة المادية الأساسية، باستخدام تقنيات مثل الشبكات المعرفة بالبرمجيات ووظائف الشبكة الافتراضية. يركز هذا البحث على التنبؤ بأكثر شريحة شبكية ملائمة لمستأجري الشبكات باستخدام نظام الاستدلال العصبي الضبابي التكيفي كمصنف متعدد لطلبات تقطيع الشبكة الواردة. يتم تخصيص شريحة ملائمة لكل مستأجر بناءً على نوع الجهاز المقدم للطلب وخصائص قناة الاتصال المطلوبة. تشمل الشرائح التي تم النظر فيها اتصالات

*Email: yousif.s.abdulameer@uotechnology.edu.iq

ضخمة من نوع الآلة، والنطاق العريض المتنقل المحسن، والاتصالات فائقة الموثوقية منخفضة التأخير. تسلط نتائج تقييم النموذج والمحاكاة الضوء على فعالية نظام الاستدلال العصبي الضبابي التكيفي في اختيار نوع الشريحة الشبكية الأنسب، حيث حقق دقة تنبؤ تبلغ 99.97٪ باستخدام بيانات غير معروضة مسبقاً على النموذج وزيادة إجمالية في استغلال النطاق الترددي بنسبة 12.6٪ أثناء المحاكاة، مما يظهر أداءً مشابهاً أو متفوقاً بالمقارنة مع الطرق المعتمدة على التعلم العميق وتعلم الآلة.

1. Introduction

Digital communications are increasingly becoming an essential part of not only the average person's life but also of various industries, enterprises, and organizations. This is largely driven by the exponential growth of mobile devices, IoT devices, smart homes, smart security devices, healthcare, and much more [1]. It is quite clear that the network service providers aim is to offer a vital communication channel for these devices. Therefore, many researchers have tackled the issue of improving the management of network communication resources [2][3][4]. Various network applications require stable and highly available communication channels. These communication channels should be served with high throughput, high bandwidth, and reliable channels personalized to the needs of each device. Since much of their functionality is tangled with continuous network access, service providers are constantly striving to fully exploit 5G technologies to provide such services to millions of user devices [5].

While 5G technologies offer a seamless mobility experience, connectivity, and higher network speeds than their predecessors, utilizing available network infrastructure remains an important factor in network management and service delivery. For 5G service providers to achieve high standards of quality of service (QoS) to their users and fully utilize the available network infrastructure, network slicing became an influential part of 5G technology implementation [6], enabling the use of virtual networks on top on the physical infrastructure, which are termed network slices (NS) [1]. These virtual networks are tailored to the requirements set by the use case of the connecting device; such ability is gained by employing network softwarization. Paradigms such as software-defined networking (SDN) and network function virtualization (NFV) [7] that in turn, provide swift reconfiguration, programmability and flexibility to create the aforementioned slices, networked devices such as the Internet of Things (IoT) gadgets could communicate through a separated network slice from other devices such as vehicular communication or critical healthcare-based devices [8]. Although network slicing increases the QoS of the network by separating communications to different portions of the network, constant monitoring of network traffic and decision-making is required to sufficiently assign network slices to devices.

Based on the use case of the network, network slices could be classified into [9]: enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (uRLLC), and massive machine-type communications (mMTC). Each service provides various QoS to users based on business requirements. Moreover, a QoS class identifier (QCI) could be added to further select the most suitable slice or sub-sub slice for a specific use case. In some cases, uRLLC service has extremely low latency, resilience, and security whereas an eMBB service may have high data rates and acceptable latency with nomadic mobility. It's possible that a mMTC connection won't need characteristics like mobility management and handover, nor will it need strict latency requirements. the eMBB service could also be further classified as low-cost, high-speed, ultra-high-speed, or high-mobility [10].
Leveraging the power of artificial intelligence (AI) techniques in mobile networking is not a new approach, it shows a potent ability to maximize network performance ranging from wireless networking [11] to network softwarization [12] and in network specific functions

[13] [14]. In turn, AI can be incorporated into many aspects of 5G networking, one of which is network slicing. Previously trained machine learning (ML) and deep learning (DL) techniques could be used to assess the massive amount of traffic that flows through the network in real-time [15], which opens the doors to improving heterogeneous network's performance, adaptiveness, reconfiguration, and higher efficiency in resource utilization [16]. In this research, we utilize the Adaptive Neuro-Fuzzy Inference System (ANFIS) for its ability to combine both fuzzy logic and neural networks, producing a powerful adaptive system for calculating predictions and decision-making. The main contribution of our research is integrating the ANFIS into a 5G network scenario, where the ANFIS predicts the most suitable network slice to enhance bandwidth utilization of the 5G network by allocating the appropriate network resources to satisfy QoS requirements of user equipment (UE), we also compare the outcome in terms of prediction accuracy with relevant research, where more sophisticated hybrid deep learning approaches are utilized as well as other machine learning techniques. The model receives incoming device traffic characteristics from the network and returns a device-specific slice to assign. This is achieved by utilizing various use cases of 5G networks and other characteristics, such as packet loss rate, packet delay time, and connection duration, which are readily available in the network.

The remaining sections of this paper are organized as follows: In section 2, related literature in the field of the paper's research is discussed. Then the next section briefly outlines the basics of ANFIS, and the dataset used to train the model. In sections 4 and 5, the model's training results, evaluation, and performance in a simulation are shown, discussed, and compared with relevant works. Finally, section 6 concludes the research with closing remarks.

2. Related Works

Artificial intelligence is a powerful tool that enhances network performance and configuration. This is why it is becoming the target for academic research in this field.
The authors of [17] proposed a hybrid model composed of a convolutional neural network (CNN) and long short-term memory (LSTM) to enhance the quality of service of wireless 5G and 6G networks by predicting appropriate network slices and inherently optimizing the reconfiguration of the network while addressing other issues in 5G networks, such as load balancing and slice failure handling.

The authors of [18] developed a DL model and termed it DeepSlice. It leverages ML and DL to predict correct slices as well as optimize network load efficiency and availability. The model was used to analyze incoming traffic and predict the appropriate network slice for known and unknown devices while also addressing slice failure. CNN and random forest (RF) were both employed in the proposed model.

In [19], the authors targeted network resource distribution among slices. Their approach utilizes software defined networking (SDN) controller architecture combined with spiking neural networks. the proposed controller and AI model optimize network performance through dynamic resource allocation, which is achieved by monitoring the network traffic and assigning bandwidth slices according to the type of demand present in the network. Simulations were run in various network demand scenarios to measure the model's effectiveness, which showed better overall network throughput.

The authors of [20] presented a machine learning powered slice prediction model for network traffic. Multiple ML techniques were deployed, namely: support vector machine (SVM), k-nearest neighbor (KNN), and decision tree (DT). The authors considered latency, reliability, availability, and throughput in their test simulation. From the techniques used, kNN and RF achieved higher slice prediction accuracy.

In [21], the authors proposed a machine learning and deep learning model to address the problems of dynamic resource scheduling, slice link provisioning, and optimal network slicing. Using a hybrid meta-heuristic model and Glowworm Sawm-based Deer Hunting Optimization algorithm, the authors were able to accurately classify network slices based on device attributes using deep belief and neural networks. The network's weights were optimized using the aforementioned meta-heuristic model.

The authors of [22] built an ML framework for dynamic network resource slicing based on ML techniques. The authors addressed the challenge of serving heterogeneous demands in 5G networks by implementing a resource orchestration framework with the goal of guaranteed QoS. Their approach involved traffic classification, dynamic slicing of network resources, and formalization of admission control and slice scheduler modules. The authors' simulation results indicated that ML-based approaches, specifically regression trees, outperform the other techniques in terms of prediction accuracy and throughput. Although at an increased training time.

In [23], the authors employed various ML algorithms for traffic classification of network slicing for 5G. The authors utilized Naïve Bayes, Support Vector Machine (SVM), Neural Network (NN), Gradient Boosting Trees (GBT), and RF algorithms to classify traffic flows. The results indicated high accuracy across all algorithms, with GBT and RF achieving nearly 100% accuracy in classifying traffic flows. Their 5G experiment showed increased network performance and bandwidth for connected user equipment.

3. Methodology

3.1 Adaptive Neuro-Fuzzy Inference System

The authors in [24] laid the groundwork for ANFIS, a Takagi-Sugeno fuzzy inference system (FIS) combined with the principles of artificial neural networks, exploiting the adaptive nature of artificial neural networks to generate and fine-tune reasonable fuzzy if-then rules that approximate a specific dataset. The model could approximate membership functions in the absence of prior human knowledge or be defined for each feature accordingly [25]. A two-input ANFIS architecture is depicted in Fig. 1. ANFIS comprises five layers. A hybrid learning approach tunes the models' output in two passes: a forward pass and a backward pass.

The inputs of the ANFIS, called premise parameters, are tuned using the gradient descent (GD) method. Consequent parameters are defuzzified weighted values for a particular rule, calculated by using the least squares estimate method (LSE). This combined learning technique produces better results while also tuning the aforementioned parameters for membership functions. The ANFIS model has a single output, calculated using weighted average defuzzification. The following describes the roles of each layer within the ANFIS as proposed by the authors in [24].
Layer 1: Each node of this layer is adaptive. The parameters in this layer are called premise parameters. Membership grades are generated based on premise parameters.

$$O_i^1 = \mu_{A_i}(x) \tag{1}$$

O_i^1 denotes the membership function grade and the association of the input (x) to the assigned linguistic label A_i. In our experiment, we chose bell membership functions for inputs to the ANFIS model with a maximum equal to 1 and a minimum of 0.

Layer 2: Nodes in this layer are denoted with the symbol Π. In this node, incoming signals from the previous layers' nodes are multiplied, and the product is sent to the next layer. These values represent each rule's firing strength.

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \quad (2)$$

Layer 3: Nodes in this layer are labeled N, and each firing strength is averaged against the sum of all firing strengths. The output of this layer is called normalized firing strength.

$$\bar{w_i} = \frac{w_i}{w_1 + w_2} \quad (3)$$

Layer 4: Each node in this layer performs the below operation. Taking in the output of the previous layers and a set of parameters, which are called the consequence parameters.

$$O_i^4 = \bar{w}(p_i x + q_i y + r_i) \quad (4)$$

Layer 5: A single node layer, this node is not adaptive and is labeled with *Σ*. It computes the overall output as the summation of all incoming signals, thus producing a single output.

$$O_i^5 = \sum_i \bar{w_i} f_i = \frac{\sum \bar{w_i} f_i}{\sum w_i} \quad (5)$$

In the case of multi-class classification, the output of ANFIS is not necessarily a single integer value; rounding it to the closest integer provides the predicted integer class [26]. This approach will be utilized in our experiment as well.

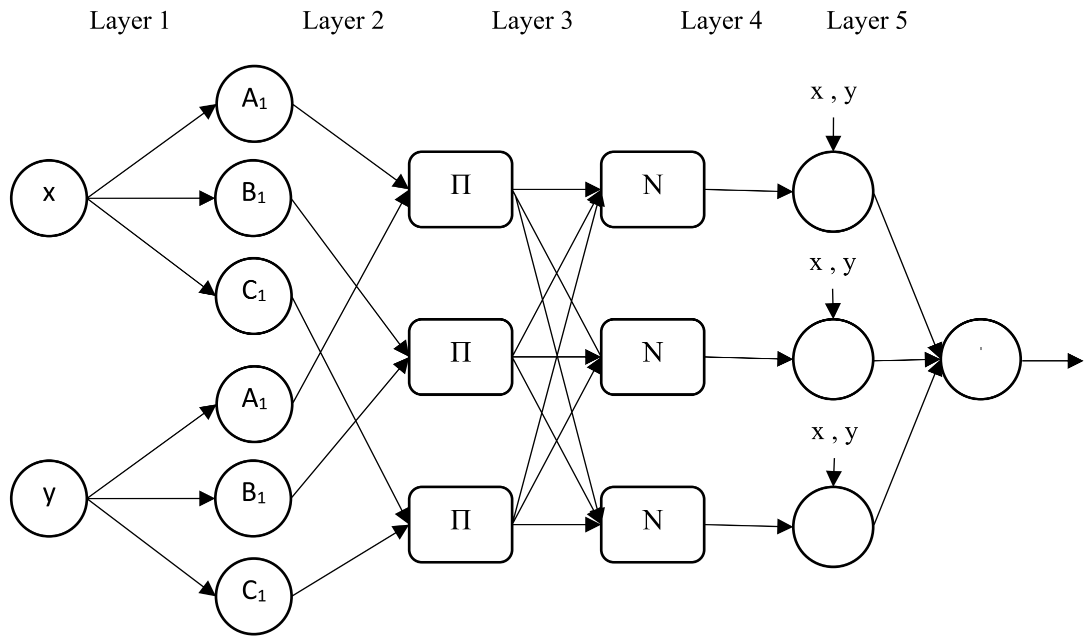


Figure 1: ANFIS layers [21]

3.2. Dataset

The dataset [27] used to train the model contains all the necessary KPIs in 63,000 unique input combinations, which are available in today's networks for both the network and connected devices. One such feature is the use-case of the network that includes the type of devices that are requesting network access. This includes AR/VR/Gaming, IoT Devices, Public Safety, Smart City & Home, Smart Transportation, and Smartphones. The dataset includes other KPIs such as packet delay budget, communication duration, packet loss rate budget, Healthcare, Industry 4.0, and guaranteed bit rate (GBR) requirement. All the aforementioned features are considered in our model. Since the model executes classification alongside the network, capturing these features is feasible. Target classes are slice types: mMTC, eMBB, and uRLLC. The dataset was adapted according to the model's requirements; label encoding was done by one-to-one mapping categorical values to numeric values. The dataset was then randomly split into three segments: training segment, testing segment, and simulation segment, to evaluate the model's performance using unseen data. A sample of the dataset is shown in Table 1.

Table 1: Sample of the dataset used for slice prediction

Use Case	LTE/5g Category	Technology Supported	Time (seconds)	GBR	Packet Loss Rate	Packet delay Budget (ms)	Slice Type
AR/VR/Gaming	1	LTE/5G	6	Non-GBR	0.001	50	eMBB
Public Safety	1	IoT(LTE-M, NB-IoT)	5	Non-GBR	0.000001	10	URLLC
Smart Transportation	1	IoT(LTE-M, NB-IoT)	7	Non-GBR	0.000001	10	URLLC
Smartphone	1	LTE/5G	2	Non-GBR	0.01	100	eMBB
Industry 4.0	6	IoT(LTE-M, NB-IoT)	15	GBR	0.001	50	mMTC

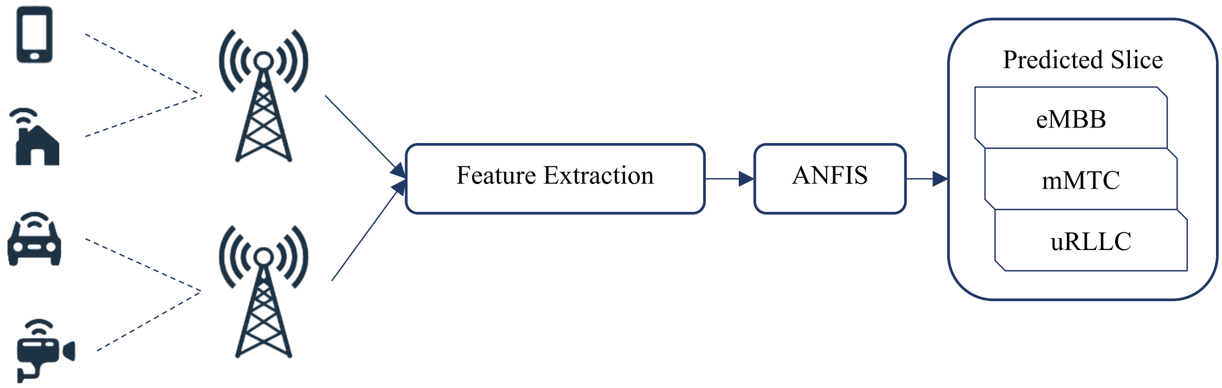


Figure 2: General overview of the proposed work

4. Proposed Work

With the massive increase in mobile network data, communication and machine learning models are becoming a viable option for statistical analysis, data forecasting, decision-making, and many other applications. Our proposed model employs the ANFIS as a network slice predictor, incoming data traffic is analyzed for key performance indicators (KPIs), and then a slice type is determined from the input that is based on incoming connection requests.
To analyze ANFIS classification performance. Slicesim [28], a Python discrete event simulator is utilized. Slicesim simulates the behavior of 5G devices and base stations in a prespecified geographical area. Simulation parameters are highlighted in Table 2. The simulation in this research involves 20 base stations in a 1750 KM area. Each client's connection request, mobility state, handover, and network consumption per slice are tracked and recorded. Three slices are implemented, eMBB, URLLC, and mMTC. Each client is assigned a different device type and can request a slice per initiation of communication of the device's traffic. The client's request is accepted by the closest base station. The client is then subscribed to the most suitable slice based on traffic type. As each client moves based

on a prespecified mobility pattern, the connection is handed over to another base station alongside the connected slice. We observed slice utilization and bandwidth usage of the simulation before and after implementing our model in the network.

The model was implemented and evaluated using Tensorflow in a Python environment. It is set up to accept six features. With the slice type being the target label, the network to make smart decisions to reserve network resources based on the importance and requirements of user equipment.

In Table 3, we summarize implementation parameters. The model is evaluated using root mean squared error (RMSE), detailed below.

- Root Mean Squared Error (RMSE): Calculating RMSE measures the average squared error rate between the actual value and the predicted value, as shown in Eq. (6):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (6)$$

for the above formula, n is the number of test samples, y_i denotes the actual value while $\hat{y}_i$ indicates the predicted value from test samples i.

Table 2: Simulation Parameters

Simulation Parameters	
Geographical Area	1750 m^2
Simulation duration (seconds)	3600, 43200
Number of Clients	5000, 500
Network Slices	eMBB, URLLC and mMTC
Mobility Types	Car, walking, stationery, and tram

Table 3: Implementation Parameters

Parameters	Values
Dataset Overview	
Number of network slices (class)	3 [0-2]
Number of samples	63168
Number of input variables	6
ANFIS Model Parameters	
Learning Tool	Tensorflow
Number of layers	5
Batch size	1
Optimizer	Adam
Membership function	Generalized Bell membership function
Loss function	RMSE
Number of epochs	5 epochs

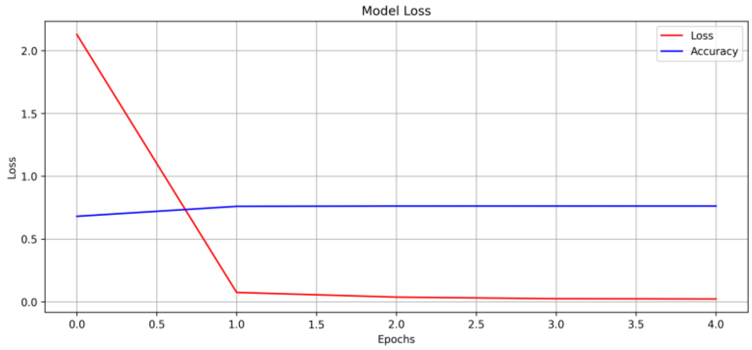
5. Experiment

5.1. Implementation Setup

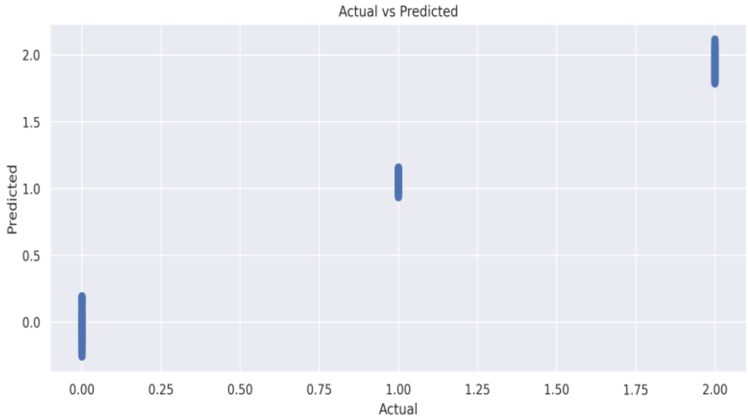
The model's training, 5G simulation, and evaluation were conducted on a desktop computer equipped with an AMD 5600X 6-core CPU, 32 GB of RAM, and an NVIDIA 8GB GTX 1070 GPU.

5.2. Training and Simulation Setup

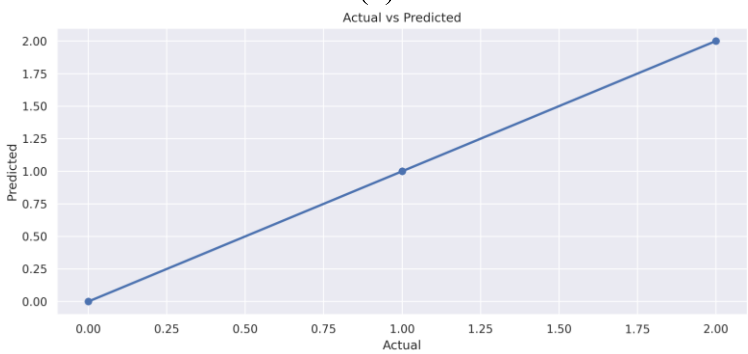
The dataset was split into three segments: training, testing, and simulation segments. We allocated 70% of the dataset to train the model, 15% to test the model, and 15% to use as KPIs for clients in the simulation.



(a)



(b)



(c)

Figure 3: Training results: (a) Evaluation of training phase (b) predicted vs. actual labels before rounding operation (c) predicted vs. actual labels following round operation

The model's output is flattened to produce the final integer slice class. Our experiment shows very promising and accurate results with optimal accuracy, recall, and an F1 score of 100% for each slice type in the dataset following the rounding operation. The model achieves 76.73% prediction accuracy before rounding application, as depicted in Fig. 3a, with RMSE value of 0.1748 for training data and 0.1591 for testing data. Fig. 3b and Fig. 3c highlight the effect of rounding the prediction output.

We ran the slicing simulation in two scenarios. A 12-hour scenario with a population of 500 clients and a 1-hour scenario with a population of 5000 clients. The simulation segment of the dataset was used to generate clients with the prediction of slices and its allocation left to the implemented ANFIS model.

5.2. Results

We observed that the ANFIS model quickly adapts the membership functions and weights over the short training duration, producing a minimum amount of loss and higher accuracy in predicting the designated slice type for each type of batch of input. Regarding prediction performance, the ANFIS takes between 8 ms to 13 ms with an average of 8.56 ms to return the predicted slice type for a single request during the simulation. In contrast, batch processing the requests benefits from Tensorflow's optimizations for simultaneous processing of samples, significantly reducing the prediction time to a range of 314 µs to 624 µs with an average of 349 µs.

In Fig. 4a and Fig. 4b, the simulation setup and client distribution per base station and slices at the end of the simulation are depicted. In Fig. 5, we showcase the results of bandwidth utilization in total. An increase of 12.6% is gained in total average bandwidth utilization as a whole and a 50% increase in each slice based on the initial simulation and then the simulation with ANFIS installed in the network in the first scenario as depicted in Fig. 5a and Fig. 5b. While in the second scenario, depicted in Fig. 5c and Fig. 5d. An increase of 10.77% is gained in total average bandwidth utilization and 45% increase in each slice with the ANFIS implementation.

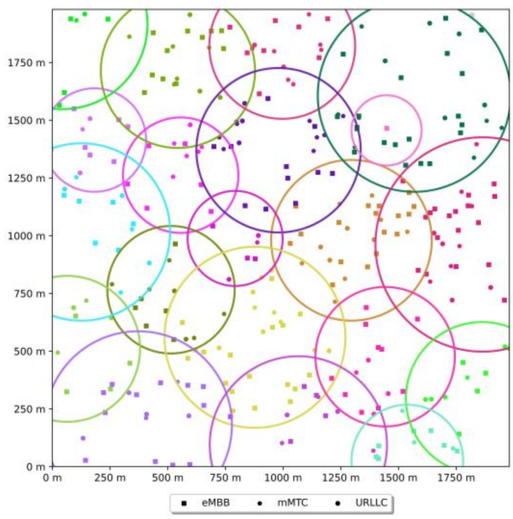


Figure 4: (a) Simulation setup with 500 clients

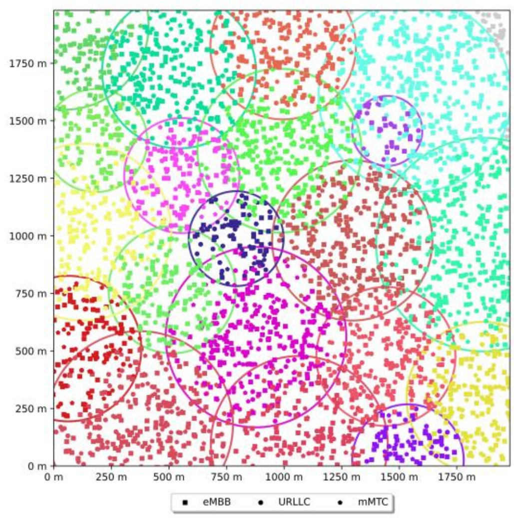


Figure 4: (a) Simulation setup with 5000 clients

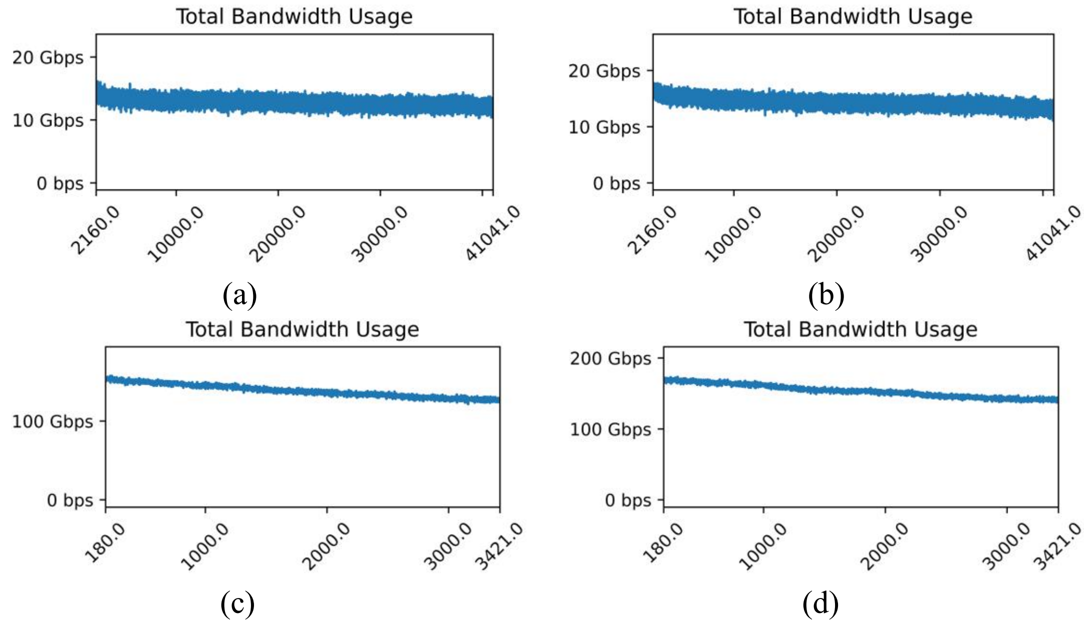


Figure 5: Bandwidth utilization during simulation: (a) 500 clients without the ANFIS model for 12 hours (b) 500 clients with the ANFIS model for 12 hours (c) 5000 clients without the ANFIS model for one hour (d) 5000 clients with the ANFIS model for one hour

The results show that ANFIS is an effective model for implementing a dynamic fuzzy inference system that can adapt to incoming network requests by predicting the most suitable network slice. This will result in an even distribution of network resources based on the indicated request type and flow characteristics used to train the model.

5.3. Comparison to Relevant Works

In this section, we compare the ANFIS prediction accuracy with other approaches taken in recent research. The authors of [18] proposed DeepSlice hybrid model, which achieved 90.62% to 95% utilizing the same KPIs used in our paper. Also using the same KPIs, the authors of [17] proposed a hybrid model that achieved 95.17% prediction accuracy using a combination of CNN and LSTM. Compared to DL models, the ANFIS offers higher prediction performance at a lower complexity and resource consumption. This stems from the model's streamlined 5-layer architecture that is based on neural networks and integrated fuzzy logic. This leads to a relatively small number of layers and parameters. In the case of our experiment, the total number of parameters is 484, 36 of which are premise parameters.

In [20], the authors employed multiple techniques that include RF, SVM, KNN, and DT to tackle slice allocation problems. The techniques achieved prediction accuracies of 97%, 92.49%, 99.89%, and 94.08%, respectively. Although the ANFIS achieved 76.73% prior to rounding the last layer's prediction results, testing and simulation results show an accuracy of 99.97% to an optimal 100% against unseen data. A summary of the comparison is shown in Table 4.

Table 4: Comparison Summary

Model	Approach	Prediction Accuracy
DeepSlice [18]	Deep learning with hybrid approach	90.62% - 95%
Hybrid CNN and LSTM Model [17]	Combination of CNN and LSTM	95.17%
RF [20]	Ensemble of decision trees	97.97%
SVM [20]	SVM classifier	92.49%
KNN [20]	Instance-based learning	99.89%
DT [20]	Tree-based classifier	94.08%
ANFIS	Neuro-fuzzy inference system	99.97%

6. Conclusions

During our research, we presented ANFIS as a possible solution to network slice prediction in 5G networks. The analysis of recent research in this area has proven that ML techniques could play a vital role in enhancing network performance on many levels of the network's operation. Our evaluation of the ANFIS and simulation further highlights neuro-fuzzy systems as a viable model for creating automated and adaptive rule sets to process device slice requests. The model is fed with KPIs from the network itself as well as device type to predict network slices, namely eMBB, mMTC, and URLLC with optimal accuracy.

The simulation results showed increased bandwidth usage and improved network resource utilization with the model's implementation. While we noted the efficiency of the model, there is a need to include additional applications and KPIs to further reflect the growing number of applications and devices. In the future, our work could be expanded by using different KPIs and employing the model in a real test environment or integrated into a 6G testbed. Another way to improve the model could be done by enlarging the processed feature set to include extra network characteristics as well as introducing reliability through estimating missing features.

Disclosure and conflict of interest

The authors declare that they have no conflicts of interest.

References

[1] R. El Hattach and J. Erfanian, “5G White Paper (2015) - NGMN.” Accessed: Jan. 10, 2024. [Online]. Available: https://www.ngmn.org/work-programme/5g-white-paper.html

[2] M. Attaran, “The impact of 5G on the evolution of intelligent automation and industry digitization,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 5977–5993, May 2023, doi: 10.1007/S12652-020-02521-X/TABLES/1.

[3] M. A. Mohammed, M. Abdulmajid, B. A. Mustafa, and R. F. Ghani, “Queueing theory study of round robin versus priority dynamic quantum time round robin scheduling algorithms,” *2015 4th International Conference on Software Engineering and Computer Systems, ICSECS 2015: Virtuous Software Solutions for Big Data*, pp. 189–194, Nov. 2015, doi: 10.1109/ICSECS.2015.7333108.

[4] H. W. Oleiwi, N. Saeed, H. L. Al-Taie, and D. Mhawi, “An Enhanced Interface Selectivity Technique to Improve the QoS for the Multi-homed Node,” *Engineering and Technology Journal*, vol. 40, no. 8, pp. 1006–1013, Aug. 2022, doi: 10.30684/ETJ.2022.133066.1165.

[5] P. Parastar, A. Lutu, Ö. Alay, G. Caso, and D. Perino, “Spotlight on 5G: Performance, Device Evolution and Challenges from a Mobile Operator Perspective,” in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10. doi: 10.1109/INFOCOM53939.2023.10229108.

[6] P. Subedi *et al.*, “Network slicing: a next generation 5G perspective,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, p. 102, 2021, doi: 10.1186/s13638-021-01983-7.

[7] S. Zhang, “An Overview of Network Slicing for 5G,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, Jun. 2019, doi: 10.1109/MWC.2019.1800234.

[8] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017, doi: 10.1109/MCOM.2017.1600935.

[9] P. S. Saravanakumar, E. Mahendran, and V. Suresh, “Adaptive End-To-End Network Slicing on 5G Networks,” in *Big Data and Cloud Computing*, 2023, pp. 319–329. doi: 10.1007/978-981-99-1051-9_20.

[10] V. K. Choyi, A. Abdel-Hamid, Y. Shah, S. Ferdi, and A. Brusilovsky, “Network slice selection, assignment and routing within 5G Networks,” *2016 IEEE Conference on Standards for Communications and Networking, CSCN 2016*, Dec. 2016, doi: 10.1109/CSCN.2016.7784887.

[11] C.-X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, “Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges,” *IEEE Wireless Communications*, vol. 27, no. 1, pp. 16–23, 2020, doi: 10.1109/MWC.001.1900292.

[12] O. Nassef, W. Sun, H. Purmehdi, M. Tatipamula, and T. Mahmoodi, “A survey: Distributed Machine Learning for 5G and beyond,” *Computer Networks*, vol. 207, p. 108820, 2022, doi: https://doi.org/10.1016/j.comnet.2022.108820.

[13] R. M. Yas and S. H. Hashem, “Intelligent Approaches for Enhancing Networked Routing Protocol,” *Iraqi Journal of Science*, vol. 62, no. 11, pp. 4121–4147, Nov. 2021, doi: 10.24996/IJS.2021.62.11.32.

[14] R. F. Ghani and L. Al-Jobouri, “Packet Loss Optimization in Router Forwarding Tasks Based on the Particle Swarm Algorithm,” *Electronics 2023, Vol. 12, Page 462*, vol. 12, no. 2, p. 462, Jan. 2023, doi: 10.3390/ELECTRONICS12020462.

[15] A. K. Mousa and M. N. Abdullah, “A Survey on Load Balancing, Routing, and Congestion in SDN,” *Engineering and Technology Journal*, vol. 40, no. 10, pp. 1284–1294, Oct. 2022, doi: 10.30684/ETJ.2022.132886.1150.

[16] B. Han and H. D. Schotten, “Machine Learning for Network Slicing Resource Management: A Comprehensive Survey,” *ZTE Communications*, vol. 17, no. 4, pp. 27–32, Jan. 2020, doi: 10.12142/ZTECOM.201904005.

[17] 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,” *Journal of Network and Systems Management*, vol. 30, no. 2, p. 29, 2022, doi: 10.1007/s10922-021-09636-2.

[18] 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 *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 762–767. doi: 10.1109/UEMCON47517.2019.8993066.

[19] M. Al-Yasari and N. Al-Jamali, “Smart Dynamic Resource Allocation Controller in Bandwidth Slicing Based on Spiking Neural Networks,” *Utilitas Mathematica*, vol. 120, pp. 368–381, Jun. 2023, [Online]. Available: https://utilitasmathematica.com/index.php/Index/article/view/1663

[20] R. K. Gupta and R. Misra, “Machine Learning-based Slice allocation Algorithms in 5G Networks,” in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, IEEE, Dec. 2019, pp. 1–4. doi: 10.1109/ICAC347590.2019.9036741.

[21] M. H. Abidi *et al.*, “Optimal 5G network slicing using machine learning and deep learning concepts,” *Computer Standards & Interfaces*, vol. 76, p. 103518, Jun. 2021, doi: 10.1016/j.csi.2021.103518.

[22] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, “Machine learning based resource orchestration for 5G network slices,” *2019 IEEE Global Communications Conference,* *GLOBECOM 2019 - Proceedings*, Dec. 2019, doi: 10.1109/GLOBECOM38437.2019.9013129.

[23] L. V. Le, B. S. P. Lin, L. P. Tung, and D. Sinh, "SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G," *IEEE 5G World Forum, 5GWF 2018 - Conference Proceedings*, pp. 20–25, Oct. 2018, doi: 10.1109/5GWF.2018.8516953.

[24] J. S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans System Management and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[25] C. T. Sun and J. S. Jang, "Neuro-fuzzy classifier and its applications," *1993 IEEE International Conference on Fuzzy Systems*, pp. 94–98, 1993, doi: 10.1109/FUZZY.1993.327457.

[26] A. N. Toosi, M. Kahani, and R. Monsefi, "Network intrusion detection based on neuro-fuzzy classification," in *2006 International Conference on Computing & Informatics*, 2006, pp. 1–5. doi: 10.1109/ICOCI.2006.5276608.

[27] A. Thantharate and V. Walunj, "Deepslice dataset." Accessed: Jan. 09, 2024. [Online]. Available: https://github.com/adtmv7/DeepSlice

[28] A. Dilmac, M. E. Gure, and T. Tugcu, "Slicesim: A simulation suite for network slicing in 5g networks." 2019. [Online]. Available: https://github.com/cerob/slicesim