Jurnal Kejuruteraan 38(2) 2026: 937-956
https://doi.org/10.17576/jkukm-2026-38(2)-36

5G Network Slicing Using Deep Learning for Hospital of The Future

Izzati Thaqifah Zulkifli[a], Nurul Asyikin Mohamed Radzi[a*], Norazizah Mohd Aripin[a], Kaiyisah Hanis Mohd Azmi[b], Nayli Adriana Azhar[a] & Faris Syahmi Samidi[a]

[a]*Institute of Power Engineering (IPE), Dept. of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia,*

[b]*Department of Electronic Systems Engineering, Malaysia- Japan International Institute of Technology, Universiti Teknologi Malaysia*

[*]*Corresponding author: asyikin@uniten.edu.my*

Received 21 May 2024, Received in revised form 25 September 2025
Accepted 25 November 2025, Available online 30 March 2026

ABSTRACT

Effective health management is essential, yet hindered by challenges in traditional healthcare systems and an uneven physician-to-population ratio. The integration of 5G networks improves communication in healthcare. This paper delves into integrating deep learning (DL) within network slicing to provide tailored solutions for the Hospital of the Future (HoF). To the best of our knowledge, this paper presents the first instance of classification techniques being used in network slicing using DL demonstrated via OMNeT simulations. We evaluate three scenarios namely, network slicing using DL, network slicing without DL, and unsliced network in terms of throughput and delay. Throughput result for URLLC network slicing using DL shows approximately a 33.33 times improvement compared to network slicing without DL and unsliced network, while eMBB network slicing using DL exhibits approximately a 10 times improvement. Additionally, mMTC network slicing using DL demonstrates a 53% improvement. Regarding delay, URLLC network slicing using DL exhibits the lowest delay compared to network slicing without DL and unsliced network, while in eMBB, network slicing using DL shows the second lowest delay. In mMTC slice, network slicing using DL shows an overlapping performance with unsliced networks, and network slicing without DL exhibits the lowest delay. It's noteworthy that the differences in delay among eMBB, mMTC, and URLLC slices compared network slicing without DL and unsliced network slices are minimal, approximately less than 1%. The intelligent distribution of resources by DL makes it ideal for critical healthcare applications, surpassing alternatives in heterogeneous networks.

Keywords: Deep learning; network slicing; OMNeT; hospital of the future; 5G

INTRODUCTION

The effective management of healthcare plays a pivotal role in ensuring that patients are provided with timely and enough access to medical specialists and facilities. Challenges arise from the conventional healthcare setup and the uneven ratio of physicians to the population, both of which constrain the provision of medical care. Specifically, the average number of doctors per 1000 persons in 30 countries in 2019 was 3.37 (WHO, n.d.; "Doctors per 1,000 People - Country Rankings," n.d.; WHO 2023). The surge in patient numbers during the pandemic has also placed significant strain on healthcare facilities and workers. According to World Health Organization (WHO) data as of September 2023, there have been 770,437,327 confirmed cases of COVID-19 globally and 6,956,900 deaths worldwide (WHO 2023). In 2020, after the WHO announced the global outbreak as a pandemic, most countries introduced comprehensive "lockdowns" to their societies — with devastating effects (Onyeaka et al. 2021). And its lingering health impacts have presented unprecedented logistical hurdles for hospitals managing COVID-19 cases as well as noncoronavirus patients seeking care. In response to previous pandemics and progress within the field, telemedicine has seen a significant increase in use, especially in large industrial nations.

However, the current telemedicine infrastructure may not be sufficient to handle the increasing demand for remote healthcare services. To address this issue, there has been a growing interest in leveraging the capabilities of 5G networks to enhance telemedicine and establish the Hospital of the Future (HoF). HoF is a technologically advanced healthcare facility designed to improve patient care. This enables telehealth professionals to evaluate remotely, diagnosis, and treatment of patients (Haleem et al. 2021). 5G's inclusion in the medical field creates the potential for a trusted communications foundation among patients, physicians and medical equipment. Telemedicine applications such as remote surgery, remote monitoring and connected ambulances are considered promising use cases for 5G network slicing in the HoF (Peralta-Ochoa et al. 2023).

The concept of network slicing is integrated into the network architecture due to the inadequacy of a one-size-fits-all approach in accommodating the diverse services of the HoF. In a study by Ericsson and ADL (Ericsson and Little 2021), approximately 25-30% of the potential 5G use cases necessitate network slicing. Their research highlights that the top six industry segments contribute to 90% or more of the potential for network slicing, underscoring its pivotal role in sustaining the industry's progress. Healthcare takes the lead among these segments, closely followed by government and transportation (Ericsson and Little 2021). Network slicing involves dividing a physical network into multiple virtual networks, each tailored to meet specific requirements of different applications or services (S. Zhang 2019). For the HoF, 5G network slicing can provide dedicated slices of the network with specific characteristics and resources required for telemedicine applications.

In the context of 5G and wireless communication, network can be sliced into three main services based on their specific requirements and applications (Osseiran, Monserrat, and Marsch 2016):

1. Enhanced Mobile Broadband (eMBB): eMBB is a category of 5G services primarily designed to provide high data rates and improved mobile broadband connectivity. It aims to deliver significantly faster internet speeds, higher capacity, and enhanced network performance.
2. Massive Machine Type Communications (mMTC): mMTC focuses on connecting a massive number of devices to the 5G network simultaneously. It is optimized for scenarios where a large number of low-power, low-data-rate Internet of Things (IOT) devices need to communicate efficiently and reliably.
3. Ultra-Reliable Low Latency Communications (URLLC): URLLC is designed to provide extremely reliable and low-latency communication. It is essential for applications where low latency and high reliability are critical, such as industrial automation, remote surgery, and other real-time applications. URLLC ensures that data transmission is highly dependable and occurs with minimal delay.

Numerous researchers (Tian et al. 2023; Ghadi et al. 2024) in the field of telemedicine have explored a wide range of techniques and methodologies to implement network slicing. However, there is still a substantial lack of research on the applicability of DL techniques in network slicing for telemedicine applications. DL, emerging as a powerful tool, has the potential to revolutionizing telemedicine by utilising its capabilities for thorough dataset analyses in order to refine the categorisation of slices.

This research provides valuable contributions to the field of healthcare communication and network slicing in the HoF setting. The key contributions include:

1. Comprehensive 5G Network Performance Dataset: The major contribution lies in the development and curation of the extensive 5G network performance data from different network slices in the hospital. This dataset serves as the primary input for our machine learning models, including those implemented in RapidMiner, as well as for our DL algorithm tailored for 5G network slicing in the HoF scenario. It provides a valuable resource to understand the specific healthcare application needs and performance metrics in the HoF.
2. Design of a Dynamic Deep Learning Algorithm: Another key contribution is the development of a dynamic and adaptive DL framework tailored for 5G network slice in the HoF scenario. This neural network learning algorithm optimally allocates the network resources, improving configurations and adapting its performance according to the requirements of the healthcare services for maximizing the efficiency of the network usage.
3. Simulation and Performance Analysis: The paper includes vast simulations and comprehensive performance analyses of the DL algorithm, specially tailored for the HoF. Insights drawn

from these cases provide valuable information regarding the efficiency and potential enhancements of 5G network slicing in the healthcare sector. In addition, these simulations use network simulator, OMNeT, which captures Quality of Service (QoS) metrics.

To the best of our knowledge, this paper presents the first instance of classification techniques being used in network slicing using DL demonstrated via OMNeT simulations. The paper's structure is organized as follows: Section II discusses related works within the proposed field, providing context for the study. Section III presents the HoF proposed method along with the experiment setup and methodology details. In Section IV, we thoroughly analyze the performance of the proposed model, followed by our results and discussion. Finally, the last section concludes our study.

RELATED WORKS

In recent years, there has been a growing interest in exploring the potential of 5G network slicing for various industries and sectors, including healthcare. The HoF, in particular, stands to benefit significantly from the implementation of network slicing technology. This literature review aims to explore the role of network slicing, a technology enabled by 5G, and its integration with various techniques. According to the definition of the Next Generation Mobile Network (NGMN) ("NGMN Alliance, NGMN 5G WHITE PAPER," n.d.): A 5G network slice is a capability allowing an end-to-end network to be partitioned into multiple independent logical networks which are optimised for specific services. This makes the network easily adaptable for various use cases and corresponding performance requirements. These slices encompass the core network, transport network, and radio access network. The radio access network plays a critical role in end-to-end network slicing. Radio Access Network slicing is a crucial component of the overall network slicing architecture. This slicing process leverages technologies like Software-defined Networking (SDN) and Network Function Virtualization (NFV) to create these slices (Li et al. 2017). Consequently, 5G networks gradually transform into a versatile and adaptable network architecture. The network slicing in our proposed model occurs at the radio access network (RAN) level which is depicted in Figure 1.

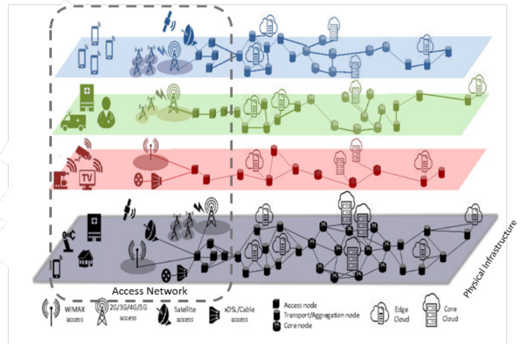


FIGURE 1. The conceptual illustration of network slicing (Wint Yi Poe and Huawei, n.d.)

This deliberate network slicing operation serves as a fundamental aspect of our approach, enabling us to tailor and optimize various network resources and services for specific applications and user groups, thereby enhancing overall network performance and efficiency.

There are various innovative methodologies in the field of 5G networks research to deal with the challenges of resource allocation and network slicing.

In (González et al. 2022) the authors introduced the Dynamic Radio Access Selection and Slice Allocation (DASA) algorithm. DASA is proposed to provide solutions for resource selection and overload scenarios under dynamic allocation of network slices. It uses technologies such as SDN and NFV. The algorithm implements multi-attribute decision making (MADM) and Analytic Hierarchy Process (AHP) to rank the services according to throughput, delay, and energy dissipation. DASA also makes use of cooperative game theory to balance the load between base stations and network slices by giving a priority for preferred users.

Taking a different approach, in (Thantharate et al. 2019), Thantharate et al. presented "DeepSlice", a framework which merges DL with network slicing in 5G networks. DeepSlice uses a DL neural network (DLNN) model to predict the network slice allocation based on traffic type and depending on KPIs of the network. Its versatile enough to support various devices in order to properly allocate resources and balance loads. Another contribution comes from Ding et al. in their work (Ding et al. 2017), where they proposed a new way of data transmission in a mobile hospital system. They used 5G network slicing along with non-orthogonal multiple access (NOMA), an approach in which users are grouped and scheduled for transmission on shared spectrum resources. The authors divided the NOMA slicing system into two slices: for eMBB also for URLLC.

In a different context, the study in (Abidi et al. 2021) introduced an efficient 5G network slicing model with three phases: data collection, Optimal weighted feature extraction (OWFE) and slicing classification. OWFE was obtained by scaling attributes with a weight function, then optimizing it by Hybrid GS-DHOA. Each of these devices was labeled (eMBB, mMTC, URLLC) via a hybrid classifier that combined Deep Belief Networks (DBN) and Neural Networks (NN).

Furthermore, the paper (Filali et al. 2022) presented a two-tier optimization approach for the radio resource allocation in an SDN based RAN. This involves allocation of resources both at SDN level and at the gNodeBs level, in order to be able to adapt according to user's growing needs. The work designs and solves an elaborate data rate optimization problem by combining the EXP3 algorithm at the SDN layer with DQL algorithm leveraging advanced techniques for gNodeB allocation.

Singh et al. (Singh et al. 2020) proposed an Internet of Things (IoT) network sub-slicing framework using machine learning for different application domains including automotive, industry, healthcare, and smart grids. This framework deals with the complexities of 5G networks, categorization of services based on its Functions/Features, and Optimization in Resource Utilization. In this paper (Guan et al. 2018), the authors use an analytical model to allocate network slice demand into the infrastructure network. Authors highlight service-oriented deployment for various slice classes such as eMBB, mMTC, and uRLLC. Topological data is collected using complex network theory, and detailed simulations are run to evaluate the effectiveness of the placement strategy.

It is essential to highlight a key difference between the model proposed in reference (Thantharate et al. 2019) and our approach, which centers around addressing network slicing challenges. While the authors of (Thantharate et al. 2019) validated their DL model in Python to confirm its functionality through simulations, our model was tested within a network simulator, offering insights into its performance within a 5G environment. Testing the model in a 5G environment allows us to more accurately replicate real-world conditions, providing an understanding of how our model performs in a 5G environment. This approach ensures the model's adaptability to the complexities of 5G networks, which surpasses the capabilities of Python-based simulations.

The works presented in reference (Abidi et al. 2021; Filali et al. 2022) and (Wint Yi Poe and Huawei, n.d.) utilized an analytical method or mathematical programming techniques for assessing their proposed model, setting it apart from our approach. Conversely, in reference (Singh et al. 2020), the author leveraged machine learning techniques to slice applications across various domains, including industry and automotive, while our proposed method exclusively concentrates on healthcare applications. In Table 1, we conduct a comparative analysis of network slicing methodologies employing neural networks and OMNeT as a simulator, highlighting the difference between our proposed model and existing studies.

The study in (Bouroudi, Outtagarts, and Hadjadj-Aoul 2023) addresses the challenges of resource allocation in shared substrate networks by employing Algorithm Selection (AS) and Deep Reinforcement Learning (DRL) techniques. The methodology involves the development of a simulation test-bed architecture using microservices. OMNeT simulation was used to simulate the solution. Notably, the paper does not utilize a classification technique dissimilar from our proposed model. The paper also utilizes a test bed simulation to evaluate DRL approaches. The performance metrics employed are different from those in our proposed model, where they evaluate algorithm usage percentage and acceptance ratio.

In (Fontana, Desogus, and Murroni 2020), the authors present the SMASH algorithm, which integrates reputation-based methods, network slicing, and neural networks to optimize network selection in heterogeneous 5G scenarios. The methodology involves algorithm development, OMNeT simulation of smart city scenarios, and the integration of neural networks into SMASH for processing complex data. The difference with our proposed model lies in several aspects: the paper does not utilize URLLC, eMBB, and mMTC slices, employs neural network simulation using MATLAB, implements hierarchical selection based on service user habits using unsupervised learning.

This paper (Rahali, Sanner, and Rubino 2020) introduces the TOM solution for overlay networks monitoring, focusing on topology selection, simulation environment setup using OMNeT, data generation, neural network training, evaluation metrics, and comparison with an SVD-based reference method. Different with our proposed model, the paper does not use a classification technique for slicing applications, instead inferring addictive metrics and conducting end-to-end measurements at the network edge.

Our proposed method, unlike the aforementioned papers, incorporates classification techniques for network slicing applications using DL. It specifically targets URLLC, eMBB, and mMTC slices and demonstrated via OMNeT simulation. This classification method is crucial because it allows for more efficient and dynamic resource allocation tailored to the distinct requirements of each slice. By leveraging DL, our approach can adapt to varying network conditions ensuring optimal performance for each service type. This enhances the overall efficiency and reliability of the network, providing a robust framework for managing diverse 5G applications.

PROPOSED METHODOLOGY

In this section, we describe the HoF application requirements and the proposed HoF network slicing model.

5G HOF APPLICATION REQUIREMENTS

5G network slicing holds great promise as a technology to meet the performance demands of supporting HoF applications, including remote surgery, remote monitoring, and connected ambulances. In this section, we outline the methodology of our proposed model for implementing 5G network slicing in the HoF environment. To initiate this process, we generate a dataset tailored to HoF applications, aligning with the 5G service requirements derived from numerous studies as shown in Table 1.

TABLE 1. Summary of Methodologies and Points of Differentiation from the Proposed Method in Reviewed Papers

No.	Title	Methodology	Difference
1.	Dynamic Machine Learning Algorithm Selection for Network Slicing in Beyond 5G Networks (Bouroudi, Outtagarts, and Hadjadj-Aoul 2023)	Utilizes Algorithm Selection (AS) and Deep Reinforcement Learning (DRL) techniques Algorithm Selection Test-Bed Architecture: -Simulation test-bed architecture with microservices -Orchestrator uses TensorFlow for DRL models - OMNeT++ for network simulation.	-Does not utilize a classification technique. -Test-bed simulation was used to evaluate DRL approaches. -Performance metrics are measured in terms of algorithm usage percentage and acceptance ratio.
2.	SMASH: a SMArt Slicing Heterogeneous 5G network selection algorithm(Fontana, Desogus, and Murroni 2020)	-Develops SMASH algorithm integrating reputation-based methods, network slicing, and neural networks -Simulates smart city scenario using OMNeT++, comparing SMASH with TYDER in key QoS metrics. -Integrates neural networks into SMASH for processing complex data	-Does not utilize the URLLC, eMBB, and mMTC slices. -Neural network simulation using MATLAB. -Implement hierarchical selection based on the service user's habits. -Uses unsupervised learning. -Compares the proposed model SMASH with TYDER.
3.	TOM: a self-trained Tomography solution for Overlay networks Monitoring(Rahali, Sanner, and Rubino 2020)	-Topology Selection: Two network topologies, A and B, were chosen from existing datasets. -Utilized the Omnet++ network emulator for traffic simulation and metric computation. -Generates simulated traffic data for training the Neural Network -Implements feedforward Neural Network architecture with ReLU activation function and Adam optimizer for training -Evaluates estimation accuracy using relative error percentage between estimated and real metrics	-Does not use a classification technique for slicing applications. -Solutions are for inferring addictive metrics. -End-to-end measurements at the network edge. -Compares the proposed method TOM with SVD

5G network slicing holds great promise as a technology to meet the performance demands of supporting HoF applications, including remote surgery, remote monitoring, and connected ambulances (Chowdary et al. 2023). In this section, we outline the methodology of our proposed model for implementing 5G network slicing in the HoF environment. To initiate this process, we generate a dataset tailored to HoF applications, aligning with the 5G service requirements derived from numerous studies as shown in Table 1.

Remote surgery, connected ambulances, and remote monitoring stand as critical components of the HoF. Remote surgery involves a surgeon performing an operation on a patient from a different location, offering a solution to geographical barriers that often hinder conventional surgery. As highlighted in Table 1, remote surgery falls within the URLLC slice, emphasizing the need for ultra-reliable and low-latency communication to ensure the precision and responsiveness required in critical applications. Remote surgery necessitates data rates of 128-400kbps for haptic feedbacks and less than 10Mbps for visual multimedia (Hollensen, Kotler, and Opresnik 2023).

Connected ambulances play a crucial role in the HoF, potentially reducing patient mortality and disability rates. These ambulances transmit various types of data, including vital signs, audio, and video. Vital signs like temperature, blood pressure, heart rate, respiration rate, and oxygen saturation typically require data rates of 10 kbps, while more complex data like ECG and EEG demand 72-86.4 kbps. Low latency, around 10ms, is vital for these connected ambulances, which utilize eMBB for data transmission (Alssaheli et al. 2022; Singh et al. 2020; Rahali, Sanner, and Rubino 2020).

Healthcare workers, including nurses, are grappling with an overwhelming number of patients due to the pandemic, making remote monitoring a lifeline. In this context, mMTC is essential in supporting a number of IoT devices like sensors and wearables. Vital signs, such as temperature, blood pressure, heart rate, respiration rate, and oxygen saturation, constitute the primary data monitored. Wearables, among other applications, serve as invaluable monitoring devices for patients in the comfort of their homes. Remote monitoring requires latency and data rate of 1000ms and 128-400kbps respectively (Singh et al. 2020), (Hollensen, Kotler, and Opresnik 2023). The mobility requirement for this project is defined based on the Report ITU-R M.24010 and 5G white paper. Table 2 presents the clear overview of 5G services requirements.

This dataset will encompass critical performance metrics such as (Hollensen, Kotler, and Opresnik 2023; Alssaheli et al. 2022; “NGMN Alliance, NGMN 5G WHITE PAPER,” n.d.):

1. Latency can be described as the time it takes for data to travel from one point to another. It holds significant importance in applications like remote surgery, where low latency is crucial.
2. Data rate refers to the amount of data that can be transmitted within a specific time frame.
3. Jitter relates to the variability in the delay of received data packets within a network.
4. Mobility refers to the ability to maintain a reliable and stable network connection for users in motion.
5. Packet size defines the volume of information that can be transmitted within a single network packet
6. Bit error rate quantifies the number of incorrectly received bits in a communication system.

HOF SIMULATION

Building on the concepts introduced in Figure 2 and Figure 3, the proposed HoF Methodology depicted in Figure 5, consists of three key phases: RapidMiner simulation, the implementation of DL algorithms, and OMNeT++ simulation shown in Figure 4.

PHASE 1: RAPID MINER SIMULATION

The subsequent first step involves comparing various machine learning models to identify the most suitable model for the HoF dataset. For this purpose, we utilize RapidMiner to address classification challenges because its versatility, user-friendliness, extensive preprocessing tools, model comparison capabilities, AutoModel features, visualization, and compatibility. RapidMiner is a data science and machine learning platform that offers an intuitive drag-and-drop interface, numerous preprocessing tools, built-in algorithms, model evaluation functions.

Network slicing is performed based on the three primary 5G network services: eMBB, URLLC, and mMTC, each employing distinct deployment strategies tailored to specific application services. The prediction category is employed to address classification problems, with available models including DL. The simulation results within the prediction category will provide insights into the performance of various supervised machine learning models.

In the RapidMiner simulation, we utilized the AutoModel feature. The background process of this simulation is illustrated in Figure 2. The process includes six stages where each stage comprises of its operator and functional elements that are interconnected.

1. Stage 1: Basic Processing: In the initial phase entails three operations: 1) Load and Process

Data, where the dataset is accurately loaded, and essential preprocessing tasks are carried out, ensuring that both labeled and unlabeled data points are prepared for subsequent stages. 2) Create Validation Set, where a training set and a validation set are precisely made, with the latter being designated for performance evaluation. 3) Basic Feature Engineering begins, addressing basic feature engineering tasks such as missing data handling and encoding.

2. Stage 2: Feature Engineering and Modeling: As the data emerges from its basic phase, it proceeds to a more advanced stage. 4) Handle Text Columns becomes crucial, especially if text data is present. It also creates a model for text processing. 5) Automatic Feature Engineering supplements the prior feature engineering, delving deeper into automatic feature creation while keeping in mind text processing, date handling, encoding, and other essentials. 6) Train Model signifies the start of actual model training, including the possibility of automatic hyperparameter tuning for optimized model performance.

3. Stage 3: Transform Validation and Scoring Data: The validation and scoring data, once well-prepared, enter a transformative phase. 7) Transform Validation Data applies the same preprocessing and feature engineering procedures to the validation data, aligning it with the modelling data. 8) Transform Scoring extends this transformation to scoring data, which lacks known target values, ensuring consistency and data integrity.

TABLE 2. 5G Services Requirements

Slice	Application	Use cases	Latency	Mobility(NGMN Alliance 2015)	Data rate	Packet size	Bit error	Jitter
eMBB	Connected ambulances	vital signs	250 ms (Singh et al. 2020)	10-120 km/h	10kbps/86.4kbps (Singh et al. 2020)	1024 bytes (Fontana, Desogus, and Murroni 2020)	10-6 to 10-1 (Dhatchayeny et al. 2017)	<20 ms (Alssaheli et al. 2022)
eMBB		visual multimedia	150 ms (Guan et al. 2018)	10-120 km/h	15/30 Mbps (Bouroudi, Outtagarts, and Hadjadj-Aoul 2023)	1500 bytes, 1024 bytes, 512 byte (Hollensen, Kotler, and Opresnik 2023)	10-7 to 10-3 (Nallappan et al. 2018)	<20 ms (Alssaheli et al. 2022)
mMTC	Remote monitoring	smart wearables	1000 ms (Guan et al. 2018)	0-50 km/h	1-100 kbps (Singh et al. 2020)	100 bytes (Alssaheli et al. 2022)	1-6 (Cai et al. 2019)	< 25 ms (Fontana, Desogus, and Murroni 2020)
URLLC	Remote surgery	Haptic feedback (kinesthetic)	3-10 ms (Q. Zhang, Liu, and Zhao 2018)	0-10 km/h	128-400 kbps (Guan et al. 2018)	100 bytes (Singh et al. 2020)	10-5 (Singh et al. 2020)	<2 ms (Q. Zhang, Liu, and Zhao 2018)
URLLC		Haptic feedback (tactile)	<5.5 ms (Q. Zhang, Liu, and Zhao 2018)	0- 10 km/h	128-400 kbps (Guan et al. 2018)	64 and 128 bytes (Rahali, Sanner, and Rubino 2020)	10-5 (Singh et al. 2020)	<2 ms (Q. Zhang, Liu, and Zhao 2018)
URLLC		visual multimedia (remote surgery)	<150 ms (Q. Zhang, Liu, and Zhao 2018)	0-10 km/h	<10 Mbps (Guan et al. 2018)	1500 bytes, 1024 bytes, 512 bytes (Hollensen, Kotler, and Opresnik 2023)	10-7 to 10-3 (Nallappan et al. 2018)	3–30 ms (Q. Zhang, Liu, and Zhao 2018)

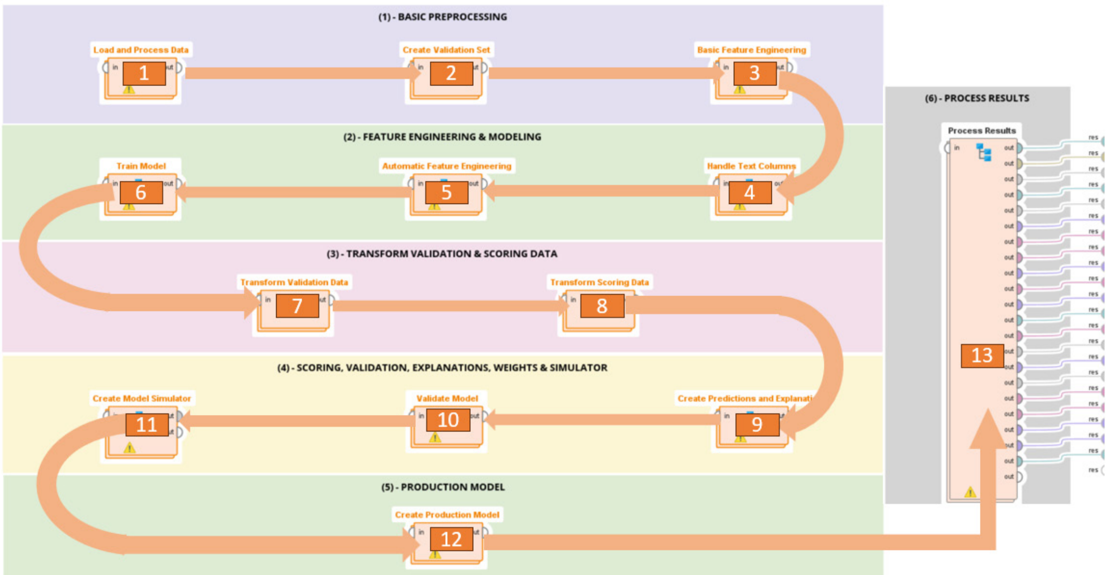


FIGURE 2. AutoModel Proccess in Rapid Miner

4. Stage 4: Scoring Validation Explanations Weights & Simulator: The data has now evolved to a stage where predictions and explanations become vital. 9) Create Predictions and Explanations applies the trained model to both the validation and scoring datasets, creating predictions and clarifying the reasoning behind them. Model-specific weights are also calculated. 10) Validate Model sets the stage for rigorous model validation through multiple hold-out sets, achieving performance estimations like cross-validation but with swifter runtimes. 11) Create Model Simulator concludes this stage by constructing a model simulator, enhancing the overall operational framework.
5. Stage 5: Production Model: With the validation process complete, the dataset proceeds to this phase. 12) Create Production Model crafts a final production model by training it with the same parameters on the combined training and validation datasets, ensuring efficient models for real-world application.
6. Stage 6: Process Results: The last process is 13) Process Results phase, where runtimes are meticulously documented, and annotations are added to the results.

These six parts, guided by their respective operators and functions, form an interconnected pipeline, transforming raw data into predictive models and finally, delivering results to classify the HoF dataset.

PHASE 2: DL ALGORITHM

Following model selection, the HoF dataset undergoes simulation within a DeepSlice (Thantharate et al. 2019) algorithm implemented in Python. Figure 3 illustrate the flowchart of the DL algorithm.

Neural networks are powerful tools in the field of machine learning that can be used for different tasks such as image recognition or text analysis. Python, being a hub for machine learning libraries, offers a seamless environment for constructing and training neural networks. The following script provides a detailed workflow for constructing, training and testing a neural network designed for classification using the Keras library.

Keras is a high-level DL framework known for its user-friendly API and modularity. It is widely adopted, versatile, and integrates well with popular backends like TensorFlow. We choose Keras for its ease of use, community support, and compatibility with various DL tasks and environments.

The first part of the code is about importing necessary libraries. These include Keras for the implementation of neural networks, NumPy for numerical computations, Pandas for data manipulation, Matplotlib for plotting, and scikit-learn for some basic data preprocessing functionalities. The code begins by assigning a random seed, this helps to ensure reproducibility in machine learning tasks, an element which is particularly vital when there is involvement of random processes. Setting the seed value equal to ‘7’ ensures that re-running the code will produce the same output.

Data lies at the core of DL, and the algorithm starts by loading the HoF datasets. Then, the algorithm splits the training data into two parts; the training set and the testing set. This step is important in evaluating the model performance while training the model. We split the data in the ratio of 70% for training and 30% for testing. This algorithm revolves around building a model for the neural network with the help of Keras. Described as 'Sequential,' this model originates from the layer-by-layer type.

After defining the model, it has to be compiled in order to make the model ready for training. In this phase, algorithm configures the model by setting the loss function, optimizer, and evaluation metric. The next step is to begin with the training phase given that the model and data are ready. The 'fit' method is called with the argument of 16 training epochs. The algorithm evaluates the model's performance with the test dataset after the training process. This accuracy is computed, clarifying the model's capability in the proper classification of data.

The algorithm also contains the model's architecture visualization. It creates a diagram which is the visual representation of the model architecture using Keras' plot model function. Finally, using Matplotlib, the algorithm plots the training and testing accuracy against the number of training epochs.

The next step consists of using the trained model to predict on a test case. The test case based on the 5G HoF requirements is created. This test case comprises 12 User Equipments (UEs) divided into 3 slices, each containing 4 UEs.

The previously trained model is loaded in the test case algorithm and predictions are generated on the test case. These predictions are then converted to class labels for further evaluation. To assess the performance of the model, the predicted labels are compared with the actual labels to calculate the accuracy.

The entire approach is comprehensive, involving training a DL model, utilizing it for predictions on the test case, and placing emphasis on preprocessing the data, model evaluation. The focus is on evaluating the DeepSlice algorithm's ability to dynamically classify the UEs according to the three network services.

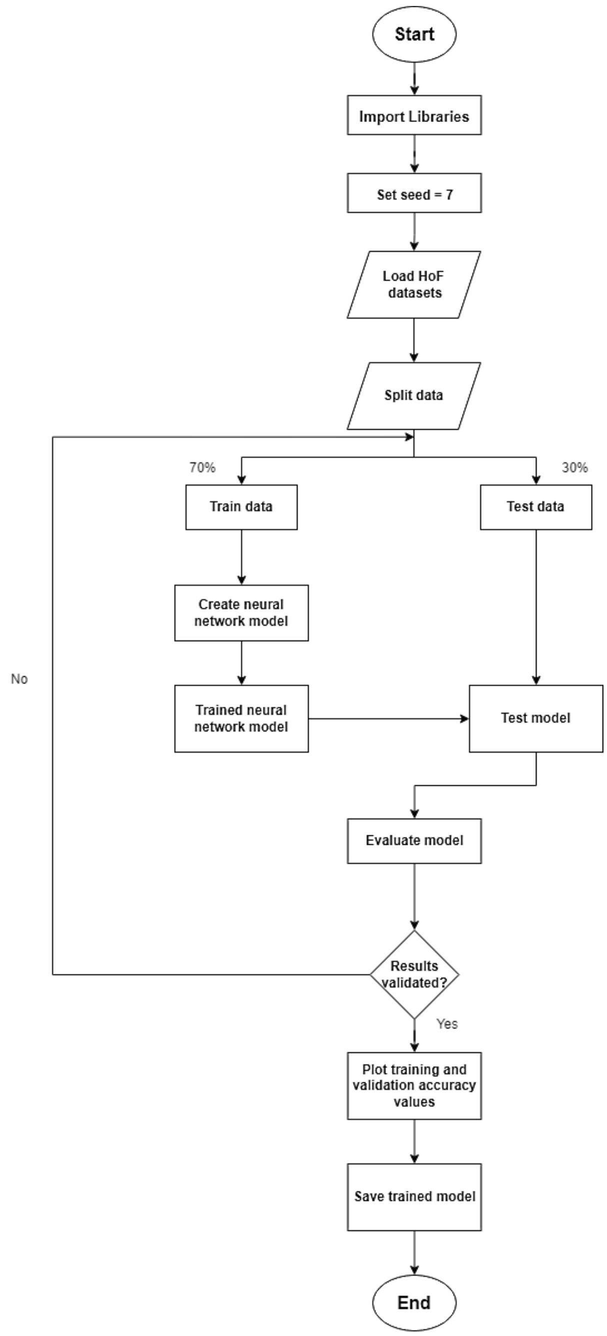


FIGURE 3. Flowchart of the proposed DL algorithm

PHASE 3: OMNET SIMULATION

Next, the proposed method is evaluated in the OMNeT network simulator due to its comprehensive network simulation capabilities, customization options, and open-source nature. To ensure realistic modeling and testing of 5G networks within the HoF context, we employ the Simu5G module within OMNeT. The previously created test case serves as input for the OMNeT simulation, with configuration parameters including mobility, packet size, and data rate being defined.

The evaluation is conducted across three distinct scenarios:

1. Network Slicing Using DL: In this scenario, we assess the performance of network slicing using DL, focusing on its ability to dynamically adapt to the specific requirements of HoF applications.
2. Network Slicing Without DL: This scenario provides a comparative perspective by evaluating network slicing without the incorporation of DL methodologies. It serves as a benchmark to measure the added value of DL in this context.

3. Unsliced Network: The third scenario involves the evaluation of an unsliced network within the HoF, providing insights into the baseline performance of 5G networks without any slicing mechanisms in place.

Configuration parameters for each scenario are defined as follows:

1. Network Slicing Using DL: Parameters specific to this scenario include the number of user equipment (UEs), resource block allocation, and DL-specific HoF requirements. Parameters specific to resource block allocation are adjusted based on service priority. For URLLC, which has the highest priority, 25 resource blocks are allocated. For eMBB, 15 resource blocks are allocated, and for mMTC, 10 resource blocks are allocated.
2. Network Slicing Without DL: Parameters for the unsliced network scenario include the number of UEs, resource block allocation with adjustments made to exclude DL-specific HoF requirements. The configuration for this scenario mirrors the DL scenario in terms of resource block allocation. This means that 25 resource blocks are allocated for the first slice, 15 for the second slice, and 10 for the last slice. Network slicing without DL utilizes a randomizer for slicing.
3. Unsliced Network: Parameters for the unsliced network scenario include the number of UEs, resource block allocation, and other relevant settings. A total of 50 resource blocks are allocated without any slicing mechanisms in place.

General simulation settings and mobility parameters are defined to create a realistic 5G simulation environment. This includes specifying transmission power, IPv4 configurator settings, and positioning of gNodeBs. Additionally, the position and mobility of UEs are set to ensure a representative network model.

Voice over Internet Protocol applications are configured for UEs and servers in the simulation. Local and destination ports, packet lengths, and start times are customized for each UE and server to simulate realistic application behaviour. Figure 4 shows the overview of OMNeT simulation.

SYSTEM ARCHITECTURE FOR IMPLEMENTING 5G NETWORK SLICING IN THE HOSPITAL OF THE FUTURE

Implementing 5G network slicing in the Hospital of the Future (HoF) environment necessitates a robust system architecture capable of accommodating diverse healthcare applications while ensuring high performance, and reliability. This section provides an explanation of the system model and architecture tailored to support the proposed framework.

FIGURE 4. Overview of OMNeT simulation

System Model Overview: The system model encompasses three key components: RapidMiner simulation, deep learning (DL) algorithm implementation, and OMNeT simulation. Each component plays a vital role in evaluating 5G network slicing within the HoF context.

Architecture Components:

1. Data Collection and Preparation:

   Relevant data pertaining to HoF applications and 5G service requirements are collected and preprocessed to create a comprehensive dataset. Performance metrics such as latency, mobility, data rate, packet size, bit error rate, and jitter are extracted and prepared for further analysis.

2. RapidMiner Simulation:

- Data Processing: The HoF dataset undergoes preprocessing within RapidMiner, including tasks such as data cleaning, transformation, and feature engineering.

- Model Training and Evaluation: Various machine learning models are trained and evaluated using RapidMiner's AutoModel feature. Models are compared based on performance metrics like accuracy, total time, and standard deviation.
- Result Analysis: Simulation outputs are analyzed to identify the most suitable model for classifying the HoF dataset.

3. Deep Learning (DL) Algorithm Implementation:

- Neural Network Construction: Python with the Keras library is employed to construct and train a DL neural network tailored for classification.
- Model Training and Evaluation: The DL model is trained on the HoF dataset, and its performance is assessed using test data. Metrics such as model loss and accuracy are calculated to gauge model effectiveness.
- Output Analysis: Simulation results are scrutinized to evaluate the DL algorithm's ability to classify the HoF dataset based on network slices.

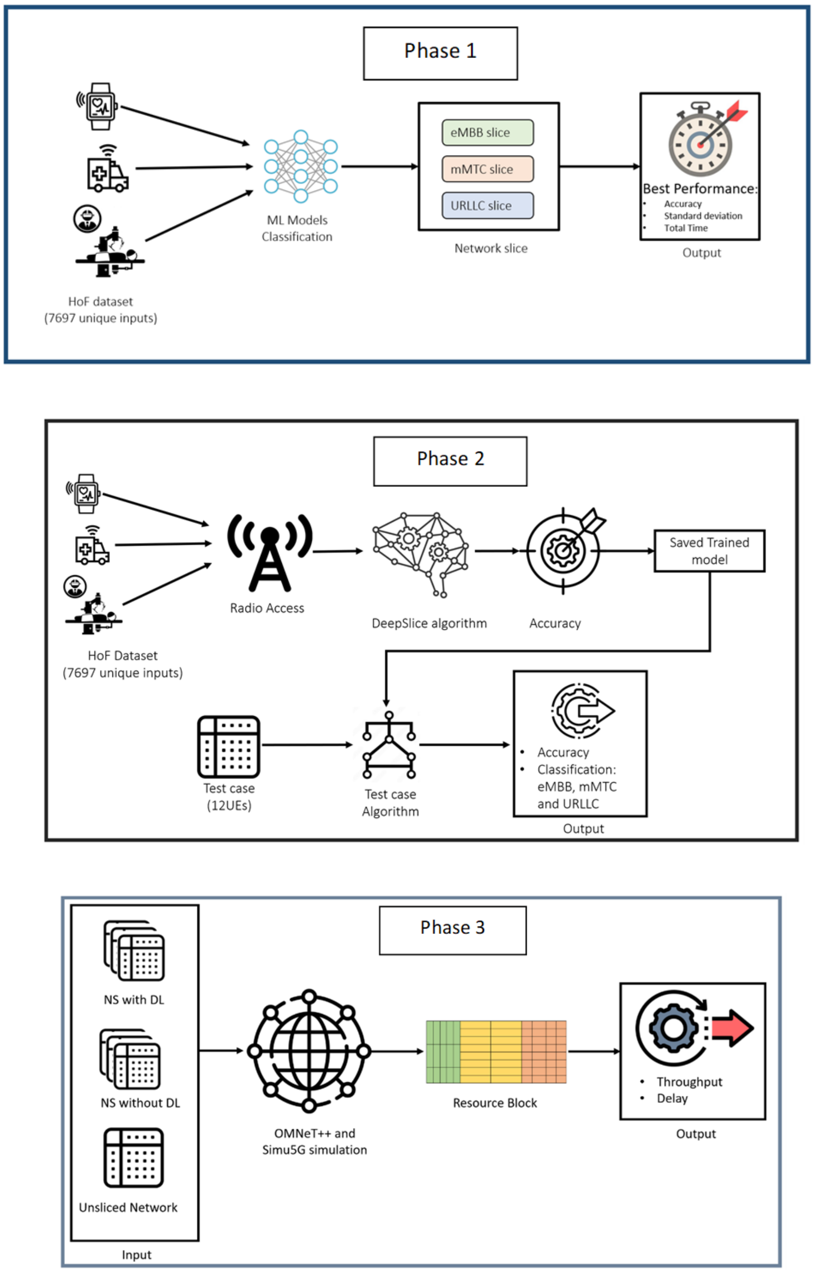


FIGURE 5. Proposed methodology of the study

4. OMNeT++ Simulation:

- Environment Setup: The OMNeT network simulator, coupled with the Simu5G module, is utilized to create a realistic 5G network simulation environment.
- Scenario Evaluation: Three scenarios are assessed: network slicing using DL, network slicing without DL, and unsliced network. Parameters such as UEs, resource block allocation, and HoF requirements are defined for each scenario.
- Performance Metrics Assessment: Key metrics such as throughput and delay are measured to evaluate network slicing performance in supporting healthcare applications.

The proposed system architecture offers a comprehensive framework for implementing and evaluating 5G network slicing in the HoF environment. By integrating RapidMiner simulation, DL algorithm implementation, and OMNeT simulation, the architecture enables thorough analysis of network slicing strategies to meet the demanding requirements of healthcare applications.

This architecture serves as a solid foundation for developing tailored solutions that enhance the efficiency, and reliability of healthcare services in the era of 5G connectivity.

RESULTS AND DISCUSSIONS

This section outlines the performance analysis of the proposed network slicing model.

RAPID MINER SIMULATION

The process started by using RapidMiner which is Machine Learning software to determine the suitability of several machine learning models on the HoF dataset consisting of 7697 unique inputs with distinct HoF criteria. These results have been previously presented and discussed in our conference paper (Zulkifli et al. 2023). The dataset was divided into two subsets: The training models were based on a 70-30 split, which formed the training set and test set. The confusion matrix became a key aspect of the evaluation process and provided insightful information on the performance of the model.

The prediction category was selected for its capability to address classification problems effectively. This category encompasses multiple algorithms, including Naive Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, DL, Decision Tree, Random Forest, and Gradient Boosted Trees. The simulation results from this prediction category demonstrated the comparative performance of various supervised machine learning models (Zulkifli et al. 2023).

Among all tested models, DL emerged as the most precise and accurate classifier across all three network slices: URLLC, mMTC, and eMBB. DL achieved accuracy of 100%. This is shown in standard deviation where DL model obtained 100% accuracy with zero standard deviation signifying an impressive consistency.

This summarizes key observations from the conference paper, with DL being considered to be the best solution for HoF slice prediction due to its high accuracy, and fast processing times.

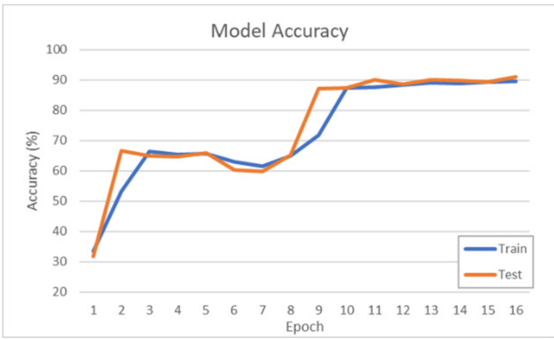
DEEPSLICE SIMULATION

Given this excellent performance, DL is selected as the preferred method for network slicing. Following this selection, the HoF dataset is then subjected to a simulation within a DL algorithm.

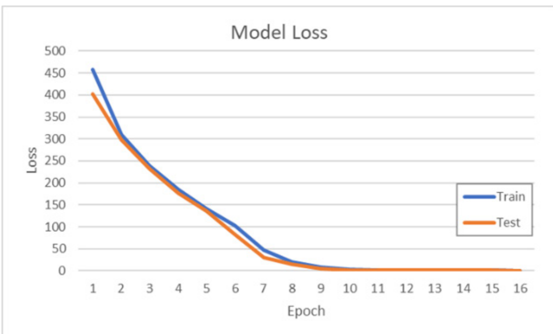
Our Python HoF simulation with DeepSlice algorithm result was remarkable with 85.17% accuracy. The obtained outcome demonstrates the algorithm's capability to identify complex patterns within healthcare data, indicating its potential to enhance network slicing in healthcare industries. To get more insight into the DeepSlice algorithm performance, we monitored model loss and model accuracy over the training and testing iterations as illustrated in Figure 6.

The x-axis in both graphs represents the number of training epochs. An epoch is a complete pass through the entire training dataset. In this context, it is a measure of how many times the DL model has been exposed to the training data during its training process. In the graph, there are 16 epochs, meaning that the model's performance is tracked over 16 rounds of training.

As shown in Figure 6a, the y-axis is the accuracy in percentage. It shows how accurate the DL model's predictions are when compared to the actual target values. Since there is no universal accuracy threshold linked to the number of epochs in DL, the optimal epoch is generally determined at the point where the validation accuracy stabilizes. The y-axis in Figure 6b represents the model loss. Model loss measures the error or the difference between the predicted values and the actual target values. It is a measure of how well the model is performing, with lower values indicating better performance. As the model learns, the loss should decrease over time, demonstrating that the model's predictions are getting closer to the actual values. Occasionally, a sudden drop in loss may occur, reflecting the model's discovery of a more efficient trajectory in the optimization landscape, often due to learning rate adjustments, optimizer momentum, batch composition, or the attainment of a critical learning milestone.



(a)



(b)

FIGURE 6. DL Algorithm: a) Model Accuracy and b) Model Loss versus Epoch

In both the model accuracy and model loss graphs, two lines are present, blue line representing the train set and orange line represents the test set. We expect that, over the course of training, the DeepSlice algorithm will show a significant increase in the model accuracy graph and a corresponding reduction in the model loss graph.

As we analyzed our model accuracy graph, we found that, initially, the algorithm rapidly adapts to healthcare data, showing a sharp accuracy increase. However, between epochs 3 and 7, there is a short drop in the precision related to the inherent complication of data for model which meets complex patterns on the training dataset. After epoch 7, the algorithm rebounds showing substantially high accuracy increases, an indication that the algorithm has learnt well how to deal with complex data patterns. A point of stagnation becomes evident, as the model near its peak performance. These fluctuations demonstrate the algorithm's ability to adapt, which is then followed by a period of consistent performance.

The downward trend of the model loss graph indicates the process in which the algorithm learns and refines its predictions. At the beginning of the process, the model starts with a relatively high loss, indicating that its predictions are far from the actual values. This is expected since the model is essentially untrained and makes inaccurate predictions. As the algorithm goes through training epochs, we see a consistent decrease in the loss.

Toward the latter epochs, the decreasing trend starts to level off, indicating that further significant reductions in loss become harder to achieve. The model is nearing its performance limits, and making substantial improvements in accuracy is increasingly challenging. The overall decreasing trend in the loss curve suggests that the algorithm is effective in minimizing the error between its predictions and the actual values.

The fluctuations in accuracy and the consistent reduction in model loss over time were in line with the anticipation that, while variations might occur, the overall trend would reflect improved performance. With an 85.17% accuracy on the HoF dataset and 100% accuracy on a test case, the incorporation of DL capabilities can be a key enabler for achieving the desired HoF vision and outcomes.

OMNET SIMULATION

In this section, we present the results of the OMNeT simulations. Specifically, we examine the received and generated throughput within all three deep learning slices, assess the throughput for all UEs of deep learning slices, and conduct a comprehensive comparison of throughput and delay across three different algorithms: the deep learning slices, slices without deep learning, and the unsliced network. These results provide valuable insights into the efficiency and effectiveness of deep learning slices and its impact on network performance when compared to conventional network configurations.

Figure 7 illustrates URLLC, eMBB, and mMTC received throughput and the generated throughput, for a single device, identified as UE1. The x-axis represents time in seconds, while the y-axis represents throughput in bits per second (bps).

In Figure 7a, 7b, and 7c, we observe the dynamic throughput behaviour over time. The blue line represents generated throughput, which starts at over 5,000 bps for URLLC, over 5,000 bps for eMBB, and 14,000 bps for mMTC. The yellow line indicates received throughput, which begins at approximately 3,300 bps for URLLC, around 3,300 bps for eMBB, and 8000 bps for mMTC. After this initial phase for all the three slices, the generated throughput consistently remains slightly higher than the received throughput for all three scenarios. While the difference is not significant, it is important to note that the generated throughput consistently exceeds the received throughput. This observation suggests that the network is efficiently managing data transfer. The observed decline in both generated and received throughput is in line with expected network adjustments. This decline typically occurs around the 50s mark, with throughput decreasing from approximately 3,334 bps to around 1,000 bps in the case of URLLC, around 3,339 bps to approximately 1,000 bps in the case of eMBB, and from an initial throughput level of approximately 8,356 bps to around 2,500 bps in the case of mMTC. Subsequently, in all three cases, the graphs stabilize at the lower throughput levels, indicating sustained and stable rates of data transfer or generation and reception within the specified network parameters.

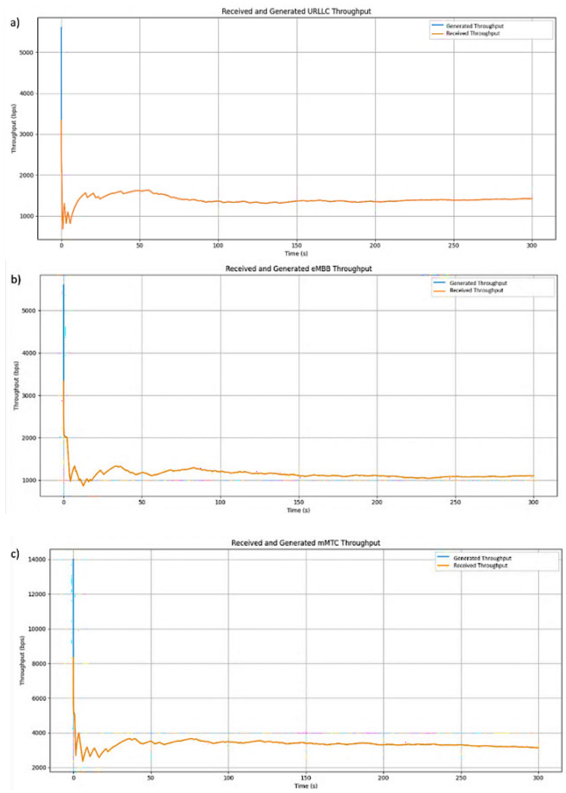


FIGURE 7. Comparison of received and generated throughput for different types: a) URLLC, b) eMBB and c) mMTC

It is evident that the generated throughput consistently exceeds the received throughput. The observation of a consistently higher generated throughput compared to the received throughput serves as an affirmation of the correct functionality of our simulation. It highlights the network's ability to ensure a continuous and reliable data transfer, aligning with the fundamental requirements for efficient and dependable communication in diverse applications. Additionally, in all three slices, the initial increase in throughput followed by a subsequent decline demonstrates network adaptations to varying conditions, with subsequent stabilization validating the effectiveness of network adjustments.

COMPARISON OF THROUGHPUT AMONG VARIOUS UES IN EMBB, MMTC AND URLLC

This section conducts a comparative analysis of throughput among four UEs in the contexts of eMBB, mMTC, URLLC. The investigation aims to provide insights into the performance variations of UEs across these communication scenarios. The y-axis represents throughput in bps, and the x-axis represents time in seconds over a 300s period.

Figure 8 illustrates the throughput performance of four different UE devices. Among these devices, UE3 demonstrates the highest throughput, ranging from 30,000 to 65,000 bps, whereas the other UEs display an average throughput of 1,000 to 3,000 bps. The packet length for UE3 is 200 KB, while the other UEs in URLLC have shorter packet lengths, with a packet size of 1024 bytes for UE3 and 100 bytes for the other UEs as shown in Table 3. The significantly higher throughput than the other UEs due to factors such as larger packet length and packet sizes in UE3. The observed higher throughput of UE3 in URLLC can be primarily attributed to the combination of a larger packet length and the unique packet size it utilizes in comparison to the other UEs. Specifically, the larger packet length in UE3 and the discrepancy in packet size contribute significantly to its superior performance. This significant difference in throughput highlights UE3's good data transmission performance compared to the other UEs.

In terms of eMBB throughput, Figure 9 illustrates that UE3 stands out with a significantly higher throughput compared to the other UEs. UE3 demonstrates throughput ranging from approximately 10,000 to 25,000 bps, whereas the remaining UEs maintain an average throughput in the range of 1,000 to 3,000 bps. The packet length for eMBB UE3 is 300 KB, while other UEs have significantly shorter packet lengths as depicted in Table 3. Additionally, UE3 uses a packet size of 512B, while the other UEs employ a packet size of 1024B. UE3 achieves significantly higher throughput in eMBB due to a combination of factors. The higher throughput achieved by UE3 is primarily the result of its utilization of a larger packet length and a distinct packet size compared to the other UEs. This combination allows UE3 to transmit more data efficiently, resulting in the observed higher throughput.

In the case of mMTC, it is evident from the data that all four UEs show similar throughput performance. The throughput for all UEs remains consistent, ranging from approximately 2,000 to 5,000 bps as shown in Figure 10.

The packet length ranges from approximately 2 KB to 700 B for all UEs, while the packet size remains consistent at 100 B. The similarity in throughput is likely due to the use of a consistent packet size of 100 bytes, which aligns with the requirements of mMTC applications, where numerous small data transmissions are common, while packet length variations have a minimal impact.

This analysis highlights significant performance variations among UEs in eMBB, mMTC, and URLLC scenarios. UE3 consistently excels in eMBB and URLLC due to its larger packet length and unique packet size, while all UEs perform similarly in mMTC, primarily because of the consistent packet size. These differences are influenced by configuration parameters and resource block allocation signifying the need for efficient resource management to maximize UE performance in different communication scenarios.

In eMBB and URLLC slices where UE3 exhibits higher throughput, resource blocks are allocated more efficiently to the slice. This allocation can involve assigning a larger share of available resource blocks to UE3, allowing it to transmit and receive data at higher rates. While, in the mMTC slice, where all UEs show consistent throughput, resource blocks are distributed more evenly among the UEs. This balanced allocation ensures that each UE gets a fair share of resources, preventing one UE from dominating the network while still maintaining a stable and consistent throughput for all devices.

TABLE 3. Parameters of URLLC, eMBB and mMTC devices

Slice	No. of user	Packet length (Bytes)	Packet size (Bytes)	Resource block
URLLC	UE[0]	2560	100	3
	UE[1]	6880	100	
	UE[2]	4240	100	
	UE[3]	200K	1500	
eMBB	UE[0]	1728	1024	2
	UE[1]	200	1024	
	UE[2]	200	1024	
	UE[3]	300K	512	
mMTC	UE[0]	760	100	1
	UE[1]	760	100	
	UE[2]	2000	100	
	UE[3]	1660	100	

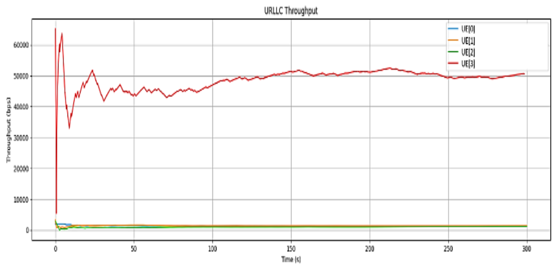


FIGURE 8. URLLC Throughput for all 4 UEs

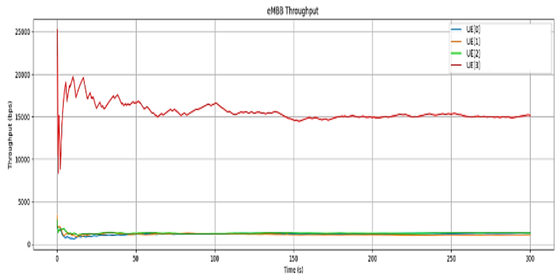


FIGURE 9. eMBB Throughput for all 4 UEs

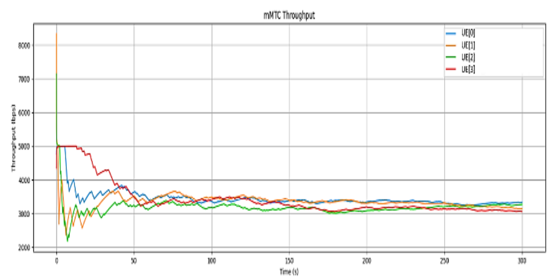


FIGURE 10. mMTC Throughput for all 4 UEs

THROUGHPUT COMPARISON BETWEEN NETWORK SLICING USING DL, NETWORK SLICING WITHOUT DL AND UNSLICED NETWORK

This section examines the performance of three network scenarios: the unsliced network shown in blue line, network slicing without DL represents by the yellow line, and the green line representing network slicing using DL. The x-axis on the graph represents time in seconds across a 300s interval, while the y-axis represents throughput measured in bps.

In Figure 11, we can observe the throughput performance UE3 across different scenarios, including URLLC network slicing without DL, and the unsliced network. UE3 demonstrates significantly higher throughput, reaching 50,000 bps at the 300s, while the other scenarios for UE3 maintain throughputs of less than 1,500 bps. The parameters for network slicing are randomized in the case of network slicing without DL, resulting in a mixed configuration of parameters. The unsliced network incorporates all 12 UEs, while network slicing divides the UEs into four UEs per slice, offering a more segmented approach to resource allocation. These findings underscore the importance of tailoring network configurations and resource allocation to meet the specific needs of different use cases and UEs. In the context of URLLC, the result translates into improved throughput, particularly benefiting UE3's communication performance.

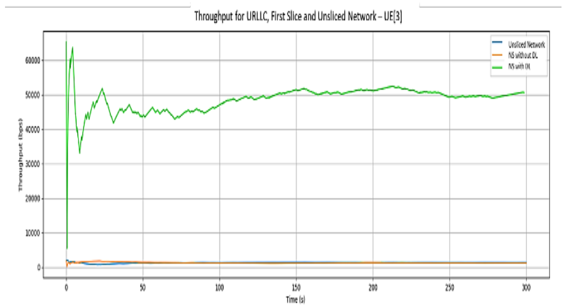


FIGURE 11. Throughput comparison of URLLC Network slicing with DL, Network slicing without DL, and an Unsliced Network.

Figure 12 illustrates the throughput performance for UE3 in various scenarios, including eMBB, network slicing without DL, and the unsliced network. The x-axis corresponds to a 300s time interval in seconds, while the y-axis represents throughput measured in bps. The UE experiences significantly higher throughput in the eMBB scenario, reaching 15,000 bps around the 300s mark as represented in Figure 12. In contrast, the UE in other scenarios maintain throughputs of less than 1,500 bps. This highlights the effectiveness of eMBB in providing good data transmission when compared to the other scenarios. In the context of network slicing without DL, parameters are randomly generated, resulting in a diversity of parameter configurations. On the other hand, the unsliced network consists of all the 12 UEs resulting in various parameter configurations. The parameters for network slicing without DL vary from 200 B to 4240 B. The packet size for one UE is 1024 B, while the other UEs use a packet size of 100 B.

These observations emphasize the importance of tailoring network configurations and resource allocation to specific use cases and UEs to achieve optimal throughput performance in diverse communication scenarios.

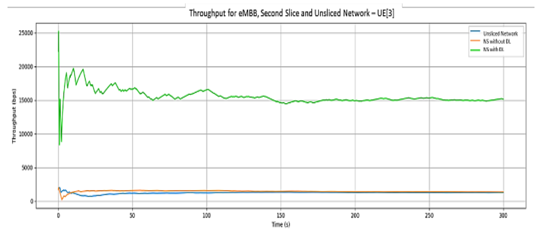


FIGURE 12. Throughput comparison of eMBB Network slicing with DL, network slicing without DL, and an Unsliced Network.

From the Figure 13, we can see that UE1 in the mMTC slice demonstrates approximately 53% higher throughput compared to UE1 in scenarios without DL and in the unsliced network. This observed pattern of significantly increased throughput holds true for the other three UEs in the mMTC slice as well. The UE1 DL begins with a throughput of more than 8000 bps, whereas the others start below 4000 bps. Subsequently, the UE1 DL experiences a sudden drop to below 3000 bps, followed by an increase. However, it remains nearly stagnant between 3000 to 4000 bps until the 300s mark. On the other hand, the UEs in other scenarios also encounter a sudden drop to below 1000 bps, then rise back, but they remain almost stagnant between 1100 to 2000 bps until 300 s. Remarkably, at the 300s mark, the DL scenario achieves the highest throughput. This significant throughput improvement highlights the effectiveness of network slicing, particularly when combined with DL techniques, in optimizing data transmission performance for UE1. The use of network slicing allows for the dedicated allocation of network resources and customized configurations to meet the specific requirements of UE1 within the mMTC slice.

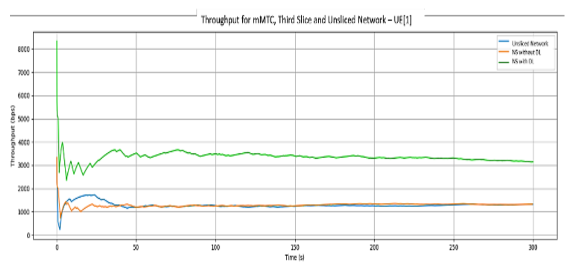


FIGURE 13. Throughput comparison of mMTC network slicing with DL, network slicing without DL, and an Unsliced Network.

DELAY COMPARISON BETWEEN NETWORK SLICING USING DL, NETWORK SLICING WITHOUT DL AND UNSLICED NETWORK

The delay performance of Network Slicing using DL, Network Slicing without DL and Unsliced Network is discussed in this section. The x-axis represents time in seconds with 300s intervals, while the y-axis measures delay in seconds.

Based on Figure 14, which illustrates the delay for URLLC Network Slicing using DL, Network Slicing without DL, and the Unsliced Network for UE1, several key observations can be made. Initially, all three scenarios display similar delay values, hovering between 0.0048 and 0.0049 seconds. Shortly after, a significant drop occurs in the delay for Network Slicing using DL, reaching 0.0042 seconds, while the delay for the Unsliced Network also decreases, dipping below 0.0042 seconds. As time progresses, the delay for all scenarios gradually increases, stabilizing after approximately 50 seconds. Throughout the duration of the simulation, the delay for Network Slicing using DL consistently remains lower than that of the other two scenarios. At the 300-second mark, Network Slicing using DL exhibits the lowest delay among the three, while the Unsliced Network experiences the highest delay. This suggests that the implementation of network slicing, particularly with a focus on Network Slicing using DL, leads to improved delay performance compared to both the Unsliced Network and Network Slicing without DL. Additionally, the stability and sustained lower delay of Network Slicing using DL indicate its suitability for applications requiring ultra-reliable and low-latency communication which is critical healthcare systems.

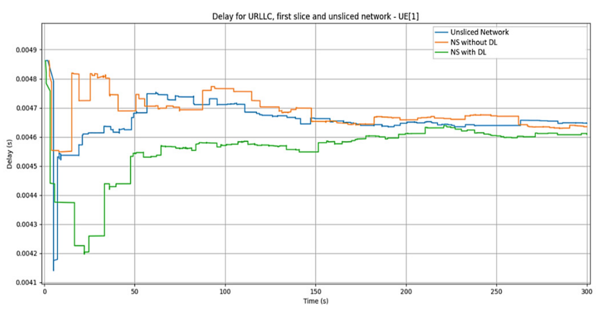


FIGURE 14. Delay comparison of URLLC Network slicing with DL, Network slicing without DL, and an Unsliced Network for UE1

The analysis of Figure 14 demonstrates the effectiveness of URLLC and network slicing in improving performance in wireless networks. URLLC outperforms other setups, highlighting its suitability for critical applications.

Figure 15 illustrates the delay across three scenarios: eMBB Network slicing using DL, Network slicing without DL, and an Unsliced Network for UE1. The x-axis represents time in seconds, up to 300s, while the y-axis represents the delay in seconds. Initially, all three scenarios exhibit delays above 0.0048s. However, the Unsliced Network experiences a rapid decline, dropping below 0.0042s. Similarly, both Network Slicing using DL and without DL exhibit delays dropping between 0.0046s and 0.0044s, showcasing a slightly slower but still notable improvement in delay reduction compared to the Unsliced Network. After the initial fluctuations, delays in all scenarios tend to stabilize after the 50-second mark. Network Slicing without DL marks the lowest delay among the three scenarios, followed by Network Slicing using DL, while the Unsliced Network demonstrates the highest delay.

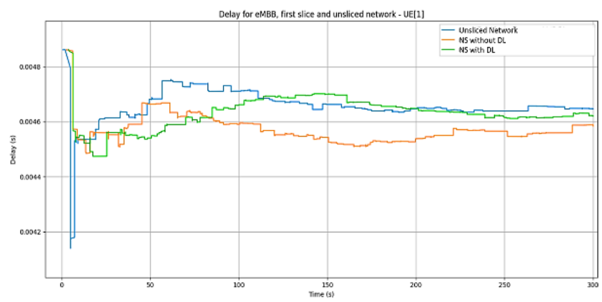


FIGURE 15. Delay comparison for eMBB Network slicing with DL, Network slicing without DL and an Unsliced Network -UE1

The analysis reveals that both Network Slicing with and without DL improve delay reduction compared to the Unsliced Network. However, Network Slicing without DL consistently outperforms Network Slicing using DL in maintaining lower delays. Despite this, Network Slicing using DL shows high throughput, indicating good performance for its applications. Delays for eMBB Network Slicing using DL applications remain within acceptable ranges throughout the observation period, indicating effective network management.

Figure 16 depicts the delay for mMTC Network Slicing using DL, Network Slicing without DL, and the Unsliced Network. The x-axis represents time, with a duration of 300s, while the y-axis represents the delay in seconds. The Network Slicing using DL delay reaches its peak at 0.0049s within the first 24s and then gradually decreases until it overlaps with the other scenarios, maintaining a consistent delay of approximately 0.0046s thereafter. It is important to note that after the 50s mark, Network Slicing using DL shows a decrease in delay, reaching its lowest point in the middle of the 50s to the

100s mark. Despite a slight increase during this period, the delay stabilizes and remains the lowest compared to the other two scenarios until the 150s mark. Network Slicing using DL exhibits the most favourable performance during this specified time period. In the comparison of delay performance, the Network Slicing using DL scenario and the Unsliced Network show an overlapping delay of approximately 0.00464s. The Network Slicing without DL configuration achieves the lowest delay, with a slight difference of 0.00462s, which is less than 1% compared to the other scenarios. This minimal difference in delay is outweighed by Network Slicing using DL's superior throughput capabilities. The mean delay for Network Slicing using DL peaks at 0.0049s, remaining within the acceptable range below 1s.

The analysis reveals that mMTC Network Slicing using DL has better performance when considering both throughput and delay. Its higher throughput performance compared to other scenarios indicates its efficiency in handling data transmission requirements.

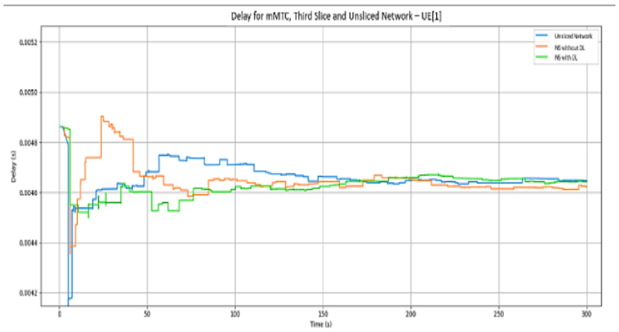


FIGURE 16. Delay comparison for mMTC Network slicing with DL, Network slicing without DL and an Unsliced Network

CONCLUSION

Given the healthcare challenges laid out in the piece including shortages of doctors, nurses, and other critical personnel during a global pandemic, it is clear that creative solutions are critically needed. Telemedicine emerges as an avenue to improve health care access and delivery. Although, the existing telemedicine structure lack the capacity to accommodate the rapidly increasing requirement for remote medical care. This issue is solved using the concept of HoF network slicing enabled by 5G technology. Network slicing enables the provision of dedicated network resources and configurations for various healthcare services: such as remote surgery, remote monitoring, and connected ambulances.

Moreover, the paper also brings attention the underexplored potential of DL techniques in network slicing for telemedicine applications. The employment of DL is proposed to improve the classification of network slices with regards to multiple performance metrics. In order to determine the suitability of machine learning models on HoF dataset developed, Rapid Miner software is used. DL is found to be top performer with high accuracy and precision.

The paper then discusses the implementation of a DeepSlice algorithm and its impressive simulation results in an 5G HoF setting. We evaluate three scenarios using OMNeT network simulation, namely network slicing using DL, network slicing without DL, and unsliced network in terms of throughput and delay. Throughput result for URLLC network slicing using DL shows approximately a 33.33 times improvement compared to other scenarios, while eMBB network slicing using DL exhibits approximately a 10 times improvement. Additionally, mMTC network slicing using DL demonstrates a 53% improvement. Regarding delay, URLLC network slicing using DL exhibits the lowest delay compared to other scenarios, while eMBB network slicing using DL shows the second lowest delay. In mMTC slice, network slicing using DL shows an overlapping performance with unsliced networks, and network slicing without DL exhibits the lowest delay. It's important to note that the differences in delay among eMBB, mMTC, and URLLC slices compared to other scenarios are minimal, approximately less than 1%. Delays for Network Slicing using DL applications remain within acceptable ranges throughout the observation period, indicating effective network management. The minimal difference in delay is outweighed by Network Slicing using DL's superior throughput capabilities.

ACKNOWLEDGEMENT

This work is supported by MOHE under Fundamental Research Grant scheme (FRGS/1/2020/TK0/UNITEN/02/14). The authors would like to acknowledge the contribution of the Institute of Power Engineering (IPE) and Innovation & Research Management Centre (IRMC), Universiti Tenaga Nasional.

DECLARATION OF COMPETING INTEREST

None.

REFERENCES

Abidi, Mustufa Haider, Hisham Alkhalefah, Khaja Moiduddin, Mamoun Alazab, Muneer Khan Mohammed, Wadea Ameen & Thippa Reddy Gadekallu. 2021. Optimal 5G network slicing using machine learning and deep learning concepts. Computer Standards & Interfaces 76: 103518. https://doi.org/10.1016/j.csi.2021.103518.

Alssaheli, Omran M. A., Z. Zainal Abidin, N. A. Zakaria & Z. Abal Abas. 2022. Software defined network based load balancing for network performance evaluation. International Journal of Advanced Computer Science and Applications 13(4). https://doi.org/10.14569/IJACSA.2022.0130414.

Bouroudi, Abdelmounaim, Abdelkader Outtagarts & Yassine Hadjadj-Aoul. 2023. Dynamic machine learning algorithm selection for network slicing in beyond 5G networks. Dlm. 2023 IEEE 9th International Conference on Network Softwarization (NetSoft), 314–316. https://doi.org/10.1109/NetSoft57336.2023.10175443.

Cai, G., Y. Fang, J. Wen, G. Han & X. Yang. 2019. QoS-aware buffer-aided relaying implant WBAN for healthcare IoT: Opportunities and challenges. IEEE Network 33(4): 96–103. https://doi.org/10.1109/MNET.2019.1800405

Chowdary, Oruganti S. S. Sumanth, Nithesh Naik, Vathsala Patil, Kinju Adhikari, B. M. Zeeshan Hameed, Bhavan Prasad Rai & Bhaskar K. Somani. 2023. 5G technology is the future of healthcare: Opening up a new horizon for digital transformation in healthcare landscape. ES General 2: 1010. https://doi.org/10.30919/esg1010.

Dhatchayeny, D. R., W. A. Cahyadi, S. R. Teli & Y.-H. Chung. 2017. A novel optical body area network for transmission of multiple patient vital signs. Dlm. 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), 542–544. https://doi.org/10.1109/ICUFN.2017.7993845.

Ding, Z., Y. Liu, J. Choi, Q. Sun, M. Elkashlan, I. Chih-Lin & H. V. Poor. 2017. Application of non-orthogonal multiple access in LTE and 5G networks. IEEE Communications Magazine 55(2): 185–191. https://doi.org/10.1109/MCOM.2017.1500657CM.

Doctors per 1,000 people - country rankings. t.th.

Ericsson & Arthur D. Little. 2021. Network slicing: A go-to-market guide to capture the high revenue potential.

Filali, Abderrahime, Zoubeir Mlika, Soumaya Cherkaoui & Abdellatif Kobbane. 2022. Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and EMBB services. IEEE Transactions on Network Science and Engineering 9(4): 2174–2187. https://doi.org/10.1109/TNSE.2022.3157274.

Fontana, Alessandra, Cristina Desogus & Maurizio Murroni. 2020. SMASH: A smart slicing heterogeneous 5G network selection algorithm. Dlm. 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 1–6. https://doi.org/10.1109/BMSB49480.2020.9379744.

Ghadi, Yazeed Yasin, Syed Faisal Abbas Shah, Tehseen Mazhar, Tariq Shahzad, Khmaies Ouahada & Habib Hamam. 2024. Enhancing patient healthcare with mobile edge computing and 5G: Challenges and solutions for secure online health tools. Journal of Cloud Computing 13(1): 93. https://doi.org/10.1186/s13677-024-00654-4.

González, Claudia Carballo, Ernesto Fontes Pupo, Luigi Atzori & Maurizio Murroni. 2022. Dynamic radio access selection and slice allocation for differentiated traffic management on future mobile networks. IEEE Transactions on Network and Service Management 19(3): 1965–1981. https://doi.org/10.1109/TNSM.2022.3150978.

Guan, Wanqing, Xiangming Wen, Luhan Wang, Zhaoming Lu & Yidi Shen. 2018. A service-oriented deployment policy of end-to-end network slicing based on complex network theory. IEEE Access 6: 1. https://doi.org/10.1109/ACCESS.2018.2822398.

Haleem, Abid, Mohd Javaid, Ravi Pratap Singh & Rajiv Suman. 2021. Telemedicine for healthcare: Capabilities, features, barriers, and applications. Sensors International 2: 100117. https://doi.org/10.1016/j.sintl.2021.100117.

Hollensen, Svend, Philip Kotler & Marc Oliver Opresnik. 2023. Metaverse – the new marketing universe. Journal of Business Strategy 44(3): 119–125. https://doi.org/10.1108/JBS-01-2022-0014.

Li, Xin, Mohammed Samaka, H. Anthony Chan, Deval Bhamare, Lav Gupta, Chengcheng Guo & Raj Jain. 2017. Network slicing for 5G: Challenges and opportunities. IEEE Internet Computing 21(5): 20–27. https://doi.org/10.1109/MIC.2017.3481355.

Nallappan, Kathirvel, Hichem Guerboukha, Chahé Nerguizian & Maksim Skorobogatiy. 2018. Live streaming of uncompressed HD and 4K videos using terahertz wireless links. IEEE Access 6: 58030–58042. https://doi.org/10.1109/ACCESS.2018.2873986.

NGMN Alliance. t.th. NGMN 5G white paper.

NGMN Alliance. 2015. 5G white paper.

Onyeaka, Helen, Christian K. Anumudu, Zainab T. Al-Sharify, Esther Egele-Godswill & Paul Mbaegbu. 2021. COVID-19 pandemic: A review of the global lockdown and its far-reaching effects. Science Progress 104(2): 00368504211019854. https://doi.org/10.1177/00368504211019854.

Osseiran, Afif, Jose F. Monserrat & Patrick Marsch. 2016. 5G Mobile and Wireless Communications Technology. Cambridge: Cambridge University Press.

Peralta-Ochoa, Angélica M., Pedro A. Chaca-Asmal, Luis F. Guerrero-Vásquez, Jorge O. Ordoñez-Ordoñez & Edwin J. Coronel-González. 2023. Smart healthcare applications over 5G networks: A systematic review. Applied Sciences 13(3). https://doi.org/10.3390/app13031469.

Rahali, Mohamed, Jean-Michel Sanner & Gerardo Rubino. 2020. TOM: A self-trained tomography solution for overlay networks monitoring. Dlm. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), 1–6. https://doi.org/10.1109/CCNC46108.2020.9045301.

Singh, Sushil Kumar, Mikail Mohammed Salim, Jeonghun Cha, Yi Pan & Jong Hyuk Park. 2020. Machine learning-based network sub-slicing framework in a sustainable 5G environment. Sustainability 12(15). https://doi.org/10.3390/su12156250.

Thantharate, Anurag, Rahul Paropkari, Vijay Walunj & Cory Beard. 2019. DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. Dlm. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 762–767. https://doi.org/10.1109/UEMCON47517.2019.8993066.

Tian, Chenjing, Haotong Cao, Sahil Garg, Georges Kaddoum, Mohammad Mehedi Hassan & Jun Xie. 2023. 5G in healthcare: Matching game-empowered intelligent medical network slicing. Alexandria Engineering Journal 77: 95–107. https://doi.org/10.1016/j.aej.2023.06.041.

WHO. t.th. Medical doctors (per 10,000).

WHO. 2023. WHO coronavirus (COVID-19) dashboard. World Health Organization.

Wint Yi Poe & Huawei. t.th. Design of network slicing and supporting systems V1.

Zhang, Qi, Jianhui Liu & Guodong Zhao. 2018. Towards 5G enabled tactile robotic telesurgery

Zhang, Shunliang. 2019. An overview of network slicing for 5G. IEEE Wireless Communications 26(3): 111–117. https://doi.org/10.1109/MWC.2019.1800234.

Zulkifli, Izzati Thaqifah, Nurul Asyikin Mohamed Radzi, Norazizah Mohd Aripin, Kaiyisah Hanis Mohd Azmi, Faris Syahmi Samidi & Nayli Adriana Azhar. 2023. Classification of hospital of the future applications using machine learning. Dlm. 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), 13–17. https://doi.org/10.1109/ISCAIE57739.2023.10165466.