الجامعة الإسلامية للتكنولوجيا
UNIVERSITE ISLAMIQUE DE TECHNOLOGIE
ISLAMIC UNIVERSITY OF TECHNOLOGY
DHAKA, BANGLADESH
ORGANISATION OF ISLAMIC COOPERATION

A COMPREHENSIVE ANALYSIS ON NETWORK SLICING FOR RESOURCE ALLOCATION OF 5G

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Electrical and Electronic Engineering

Academic Year: 2022-23

Presented to the Academic Faculty by

A. G. M. Fuad Hasan Akanda (190021209)
Rayan Hossain Khan (190021214)
Shadman Saqib (190021228)

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh

June 2024

A dissertation on

A COMPREHENSIVE ANALYSIS ON NETWORK SLICING FOR RESOURCE ALLOCATION OF 5G

Approved by

Prof. Dr. Md. Raqibul Islam
Head of the Department
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh

Supervised by

Dr. Mohammad Tawhid Kawser
Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur-1704, Bangladesh

Declaration of Authorship

This is to certify that the work presented in this thesis paper is the outcome of research carried out by the candidates under the supervision of Dr. Mohammad Tawhid Kawser, Professor, Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT). It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the reward of any degree or any judgement.

Authors

A. G. M. Fuad Hasan	**Rayan Hossain Khan**	**Shadman Saqib**
ID: 190021209	**ID: 190021214**	**ID: 190021228**

Dedicated to
our family and friends whose unwavering love and support throughout our academic lives made this work possible.

Acknowledgments

All praises belong to almighty Allah (SWT), the most gracious, and the most merciful, who bestowed upon us the ability to reason and use our intellect to educate ourselves with the knowledge of the world. It is He who grants us the strength and patience to complete our thesis successfully.

Our journey towards the bachelor's degree would not have been possible without the support of many people. It is our great pleasure to take this opportunity to thank them for their endless guidance, support, and advice throughout this journey.

First and foremost, we wish to express our deepest gratitude to our academic and research supervisor, Professor Dr. Mohammad Tawhid Kawser, for his continuous assistance, inspiration, and invaluable suggestions from time to time. His words inspire us to be a genuine scientist and sparks within us a curiosity for the unknown. We could not have imagined a better mentor for our thesis work.

Secondly, we would like to thank all the faculty members of the EEE department for their continuous support and encouragement. They have provided a friendly environment for us to learn and grow.

Finally, our deepest gratitude goes to our family who always listened toour hardships and enchanted us with their delightful words. Last but not the least, we would like to thank our friends who accompanied us through this journey.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3GPP	3rd Generation Partnership Project
AMPS	Advanced Mobile Phone Service
AR	Augmented Reality
B-GNB	Bagging Gaussian Naive Bayes
BER	Bit Error Rate
CNN	Convolution Neural Network
DRRA	Decomposition and Relaxation based Resource Allocation
DL	Deep Learning
DRL	Deep Reinforcement Learning
D2D	Device to Device
DCS	Digital Cellular Service
eMBB	Enhanced Mobile Broadband
FBMC	Filter Bank Multi-carrier
F-OFDM	Filtered Orthogonal Frequency Division Multiplexing
FDMA	Frequency Division Multiple Access
GFDM	Generalized Frequency Division Multiplexing
GSM	Global System for Mobile Communications
GS-DHOA	Glowworm Swarm and Deer Hunting Optimization Algorithms
IoT	Internet of Things
ISI	Inter-Symbol Interference
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IFFT	Inverse Fast Fourier Transform
LSTM	Long Short Term Memory
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
M2M	Machine to Machine
mMTC	massive Machine Type Communications
MIMO	Multiple Input Multiple Output
NOMA	Non-Orthogonal Multiple Access
ML	Machine Learning
OQAM	Offset Quadrature Amplitude Modulation
OWFE	Optimal Weighted Feature Extraction

OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PAPR	Peak to Average Power Ratio
PCS	Personal Communications Service
PDC	Personal Digital Cellular
PGACL	Policy Gradient-based Actor-Critic Learning
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RFID	Radio-Frequency Identification
RF	Random Forest
RB	Resource Block
SDN	Software Defined Networking
SC-FDMA	Single Carrier Frequency Division Multiple Access
URLLC	Ultra-Reliable Low-Latency Communication
UFMC	Universal Filtered Multi-Carrier
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VLAN	Virtual Local Area Networks
VNO	Virtual Network Operators
VR	Virtual Reality
WCDMA	Wideband Code Division Multiple Access
WSN	Wireless Sensor Networks

Abstract

Optimizing resource allocation in 5G networks involves reconciling the conflicting requirements of enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC), and Ultra-Reliable Low Latency Communications (URLLC). eMBB demands high data rates and substantial bandwidth to support applications such as high-definition video streaming and virtual reality. In contrast, mMTC requires the network to support a massive number of low-power, low-data-rate devices, essential for the Internet of Things (IoT). URLLC poses the additional challenge of requiring ultra-low latency and high reliability for critical applications such as autonomous driving and remote surgery.

Advanced machine learning techniques, specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, were utilized to develop prediction models. The CNN model achieved an impressive accuracy of 97% in predicting network slice allocations, while the LSTM model demonstrated a remarkable 97-98% accuracy in time series forecasting. Key achievements include enhanced model performance through meticulous hyperparameter tuning and data augmentation, which improved the model's robustness and generalization across diverse data scenarios.

Processing time was significantly reduced by implementing early stopping and batch normalization techniques, accelerating model convergence and deployment. Additionally, optimized load scheduling ensured balanced workload distribution across the network, enhancing overall system performance and reducing latency. This comprehensive approach addresses the diverse and stringent demands of 5G services, demonstrating a robust, efficient, and scalable framework for 5G network resource allocation. This research ensures improved network performance and reliability, effectively meeting the varied requirements of eMBB, mMTC, and URLLC, thereby contributing to the advancement of next-generation wireless communications.

Chapter 1

Introduction

A comprehensive analysis of network slicing for resource allocation in the context of 5G, focusing on the co-existence of eMBB, URLLC and mMTC with their distinct and sometimes contradictory requirements by this project. As 5G networks are envisioned to support diverse use cases, each with specific demands on data rates, latency, and connectivity, efficient resource allocation becomes paramount.

Given the complexity and dynamic nature of 5G services, automation emerges as a crucial component for accurate and timely slice prediction of resource allocation. This paper proposes leveraging deep learning and various machine learning algorithms to enhance the precision of resource allocation predictions. By harnessing the capabilities of these advanced technologies, we aim to address the challenges posed by the coexistence of eMBB, URLLC, and mMTC services in 5G; paving the way for optimized and adaptive network slices that cater to the diverse requirements of next-generation communication systems. This research contributes to the ongoing discourse on the practical implementation of network slicing as a key enabler for the efficient deployment and management of 5G networks.

1.1 Brief history of evolution of cellular networks

Although our project is mainly focused on the application of Network Slicing in 5G Resource Allocation, it is always helpful to have some preliminary knowledge on evolution of wireless networks throughout the decades of development. Besides one particular generation is not immediately eliminated right after implementation of the next generation. Otherwise, it will cause trouble to consumers. Though there is 5G under deployment right now, we still have many active base stations of 3G, 4G which are providing services. Even some places have active base stations of 2G. Therefore, this section is aimed to discuss briefly on the existing major technologies that are in development or have been developed in the field of cellular networks. [1]

1.1.1 First Generation (1G) Cellular Networks

The development of the 1G cellular networks commenced in the 1970s, utilizing frequencies within the 800-900 MHz band. These early systems employed FDMA and analogue frequency modulation. In FDMA, each carrier accommodated a single traffic channel, requiring two channels for a call— one for backward link or mobile to base station and another one for forward link or base station to mobile. To prevent interference, a duplexer was essential.

Throughout the 1980s, various countries launched their 1G systems. For instance, the Nippon Telephone and Telegraph (NTT) released the first analogue cellular system in 1979, while Ericsson Radio Systems AB introduced the Nordic Mobile Telephone (NMT) 900 system in 1981. AT&T joined the fray in 1983 with the AMPS. These milestones marked the diversification of 1G technologies, setting the stage for the evolution of mobile communication during this pivotal era.[2]

1.1.2 Second Generation (2G) Cellular Networks

The 2G of cellular systems emerged in the late 1980s to early 1990s, experiencing swift global adoption. Varied technologies were adopted by different countries, exemplifying regional diversity. European nations embraced GSM/DCS1800/PCS1900, while the USA opted for the IS-54/136 and IS-95 standards. In contrast, Japan chose the PDC standard. This era marked a significant advancement in mobile communication, offering enhanced capabilities and paving the way for further technological innovations in the telecommunications landscape.[2]

1.1.3 Third Generation (3G) Cellular Networks

The inception of 3G networks dates back to 1992, with collaborative efforts from two groups: International Telecommunications Union Radio Communications (ITU-R) and Telecommunications (ITU-T). Envisioned to operate in the 1885-2200 MHz band, initial expectations for 3G included no defined requirements for equipment or providers. However, it was anticipated that the peak data rate of a minimum downlink is 2 Megabit/s for pedestrians and 384 kbit/s for moving vehicles would be achieved. The two predominant standards in 3G are cdma2000 developed by 3GPP2 and WCDMA developed by 3GPP.[2]

1.1.4 Fourth Generation (4G) Cellular Network

4G networks, designed to meet demanding performance criteria, aimed for peak data rates of 1 Gbps in stationary or low-mobility scenarios and 100 Mbps in fast-moving scenarios. Essential requirements included seamless handoff across diverse networks, uninterrupted connectivity between multiple networks and high quality service for next generation multimedia. Backward compatibility with existing wireless standards was also a prerequisite.

4G technology featured flexible channel bandwidth ranging from 5 to 20 MHz and optionally extending up to 40 MHz Notable standardized and commercially deployed 4G systems include LTE standardized by 3GPP and IEEE 802.16e standardized by IEEE. OFDMA and SC-FDMA were commonly employed for resource allocation in 4G cellular networks. Subsequent versions such as LTE-A and IEEE 802-2012 further refined and expanded the capabilities of 4G, contributing to the evolution of high-speed, versatile wireless communication technologies.[2]

1.1.5 Fifth Generation (5G) Cellular Networks

5G networks are currently undergoing research as well as deployment worldwide. The goals and requirements of 5G span a broad spectrum, catering to diverse needs ranging from IoT connectivity to accommodating extremely high-capacity networks for the growing global population. Anticipated to link at least 100 billion devices, 5G aims to facilitate low data-rate and efficient machine-to-machine communication.

Key performance targets include latency of less than 1ms and peak data rates reaching 1 Gbps. These specifications are poised to support a myriad of applications, including but not limited to self-driving cars, virtual reality, industrial automation, and streaming services. The versatility and capabilities of 5G are expected to usher in a new era of connectivity, revolutionizing various industries and enhancing user experiences across a wide array of applications.[3]

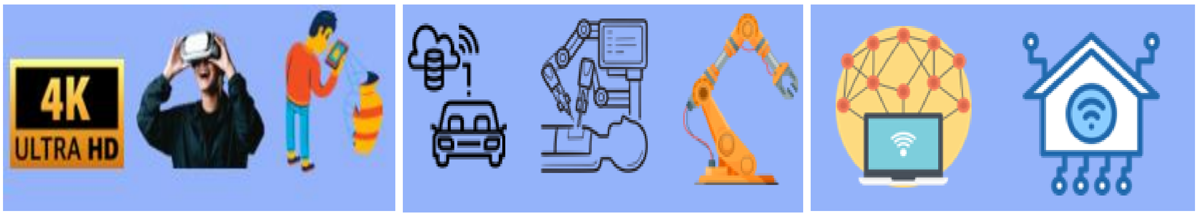
1.2 Different Aspects of 5G

1.2.1 Service Category

1. **eMBB:** eMBB is a pivotal service category within 5G, catering to increasing demand for higher data rates, enhanced capacity and broadband experiences. It is characterized by requirements such as ultra-fast data rates, improved efficiency, and the ability to support bandwidth-intensive applications. eMBB serves as the backbone for applications like 4K video streaming and immersive experiences in VR as well as AR, delivering a seamless and high-quality user experience.[4]
2. **URLLC:** URLLC is designed to meet the stern demands of applications requiring low latency, high reliability, and short packet sizes. It is tailored for critical uses such as V2X communication, remote surgery and industrial automation. URLLC ensures

minimal latency, enabling real-time and mission-critical communication. Its focus on reliability makes URLLC indispensable for scenarios where the consequences of communication delays can be crucial.[4]

3. **mMTC:** mMTC addresses the need for connecting a vast number of devices efficiently, prioritizing energy efficiency, scalability and massive device connectivity. Tailored for applications like the Internet of sensor networks and IoT. mMTC accommodates the diverse requirements of a multitude of connected devices, ranging from sensors and actuators to various IoT endpoints. Scalability and energy efficiency are paramount in supporting the extensive proliferation of connected devices in a seamless and sustainable manner.[4]



eMBB URLLC mMTC

Figure 1-1: Applications of 5G Services

1.2.2 Modulation Scheme

The following modulation schemes are proposed for 5G so far:

1. **OFDM:** In the context of multiple carrier transmission, OFDM divides channel bandwidth into parallel sub channels known as subcarriers. This method allows for multiplexing in both the time and frequency domains. The subcarriers are strategically spaced to ensure they are not frequency selective, exhibiting a uniform gain across frequency domain. This innovative approach in OFDM contributes to efficient data transmission, mitigating frequency-selective effects and optimizing spectral efficiency.[5]
2. **FBMC:** FBMC stands out from OFDM by forgoing cyclic prefixes and employing individual subcarrier filtering. Noteworthy features include spectrum-efficient tapered filters with extended time constants, achieving symbol overlap for enhanced efficiency. Orthogonality is maintained through an offset-QAM modulation scheme, eliminating the need for guard times and cyclic prefixes.[6] FBMC's use of PHYDYAS model filters optimizes performance, offering advantages such as asynchronous transmission adaptability and efficient mobility handling. Challenges include managing scattered

pilots, limited compatibility with MIMO schemes, and the need for a carrier guard in the uplink, making it less suitable for short bursts. [7]

3. **UFMC:** UFMC improves upon Cyclic Prefix OFDM by dividing the signal into sub bands before filtering, eliminating the cyclic prefix for enhanced ISI protection. Using Dolph-Chebyshev filters of length L ensures orthogonality without the need for OQAM modulation. N subcarriers are organized into fixed subbands, enabling efficient N-point IFFT computation. UFMC offers high spectral efficiency, low overhead, and is well suited for low latency modes and short burst transmissions. The main challenges include limitations at high data rates, increased delay spread, and the need for multi-tap equalizers. [8]
4. **GFDM:** GFDM represents an adaptive multiple carrier transmission method, distinguished by its non-maintained carrier orthogonality. This characteristic allows for improved management of out-of-band emissions and reduced PAPR. Individual filtering of each subcarrier spreads the available spectrum into multiple segments, making GFDM suitable for cognitive radio implementation. Circular filtering using Root Raised Cosine filters with tail biting facilitates flexibility in dividing data into sub-symbols, accommodating applications with diverse latency requirements. While adjacent sub-carriers overlap, enabling asynchronous data transmission. This results in a non-orthogonal waveform. It leads to challenges such as higher BER and the need for equalization. Despite its complexities, GFDM offers advantages such as low PAPR, efficient multiuser scheduling, spectrum hole clustering, and block-based transmission. Challenges include complex receivers, the use of matched filters for interference removal and limitations for MIMO due to OQAM. [7]
5. **F-OFDM:** F-OFDM optimizes spectrum utilization through distinctive sub-band filtering. An extension of classic OFDM, it introduces a short equiripple sub-band filter for enhanced out-of-band emission control. F-OFDM's flexibility in parameterization allows tailored adaptations for different users and applications, supporting asynchronous transmission.[7] Despite its advantages, challenges include implementation and structural complexity. The receiver mirrors the transmitter's model filter. F-OFDM offers efficient spectrum usage, out-of-band rejection and versatile coexistence of waveforms. [9]
6. **NOMA:** It is a proposed candidate for 5G but still it is not deployed that much. NOMA revolutionizes multiple access techniques by abandoning strict orthogonality among users. Unlike conventional methods, NOMA allows users to share same time and

frequency resource concurrently, optimizing spectrum usage. Its unique approach enhances spectral efficiency and reduces the PAPR.[10] NOMA's application extends to diverse scenarios, including cognitive radio and IoT; providing a flexible framework for accommodating various communication requirements. Challenges include the need for sophisticated signal processing techniques to separate overlapping signals at the receiver. Despite these challenges, NOMA stands out for its innovation in efficient multiple access strategies. [11]

1.2.3 Technological Aspects

The rapid evolution of wireless communication technologies has introduced several advanced methodologies to meet increasing demand for high speed as well as reliable connectivity. Among these advancements, Massive MIMO and millimeter wave technology are prominent, driving the future of wireless networks, particularly 5G. These technologies are integral in addressing the challenges of modern communication, such as capacity, latency, and efficiency, despite encountering specific implementation hurdles.

1. **Millimeter-Wave Technology:** It pertains to utilizing radio frequencies within the 30 to 300 gigahertz range, a crucial element in the evolution of 5G and advanced wireless networks. This technology facilitates high-capacity data transmission, although it presents challenges such as limited range and susceptibility to atmospheric absorption. Despite these hurdles, millimeter-wave technology remains instrumental in achieving low-latency, high-capacity communication. [12]
2. **Massive MIMO:** It stands as a pivotal advancement in wireless communication, characterized by base stations equipped with a substantial number of antennas for simultaneous communication with multiple users. This approach significantly enhances spectral efficiency, amplifies network capacity, and optimizes overall performance in wireless systems. Integral to 5G and upcoming standards, Massive MIMO offers improved spatial multiplexing and effectively addresses signal interference challenges.[13]

1.2.4 Cross-Cutting Concerns

As 5G networks evolve, several cross-cutting concerns are critical for their successful implementation and operation. These include network slicing, security and privacy, energy efficiency, and global standardization and collaboration. Addressing these areas ensures that

5G networks meet the ever growing diverse demands of users and applications while maintaining robust performance and reliability.

1. **Network Slicing:** It's a pivotal architectural concept in 5G networks, facilitating the invention of virtualized network instances. These instances are engineered to specific service requirements, allowing operators to optimize resource utilization and deliver customized network experiences to users based on their needs and preferences. It enables the allocation of dedicated resources as well as functionalities for various services like eMBB, URLLC and mMTC.[14]
2. **Security and Privacy:** Security as well as privacy are paramount concerns in 5G networks. With the proliferation of connected devices and the transmission of sensitive data, ensuring the overall protection of user data and network integrity is crucial. This involves implementing robust authentication protocols, advanced encryption algorithms and secure transmission mechanisms to safeguard against data breaches, unauthorized access and cyberattacks. [15]
3. **Energy Efficiency:** Energy efficiency is a significant focus area in 5G networks to minimize their environmental impact and operational costs. It encompasses various strategies such as smart power management techniques, dynamic resource allocation algorithms, and energy-aware protocols. By optimizing energy consumption across network components and infrastructure, operators can enhance sustainability while maintaining high-performance levels and service reliability.[16]
4. **Global Standardization and Collaboration:** The development and deployment of 5G technology rely heavily on global standardization efforts and collaborative frameworks. International organizations and industry consortia play a key role in establishing unified standards for 5G networks, ensuring interoperability, compatibility, and seamless integration across different network deployments and geographic regions. Collaborative initiatives foster innovation, knowledge sharing, and harmonized regulatory environments, driving the widespread adoption and deployment of 5G technologies worldwide.

1.2.5 Applications

The advent of advanced wireless communication technologies has enabled a multitude of innovative applications across various domains. These applications leverage the capabilities of modern networks to enhance connectivity, efficiency, and user experience. Key among these applications are the Vehicle to Everything (V2X) communication, Machine to Machine (M2M) communication, Device to Device (D2D) communication, Virtual Reality (VR), Augmented Reality (AR) and IoT. Additionally, Smart Cities, eHealth, and Industry 4.0 are emerging as significant areas where these technologies are being applied to revolutionize operations and interactions. Some of the groundbreaking applications are:

1. **IoT:** IoT refers to a network of interconnected objects and devices embedded with software, sensors and connectivity capabilities. These devices communicate and exchange data, enabling the collection, transmission, and analysis of information. IoT extends connectivity beyond traditional computing devices to everyday objects, creating a seamless ecosystem for data-driven insights.[17] Its technologies include RFID, WSN and other communication protocols, forming a diverse toolkit for building intelligent and interconnected systems. The architecture of IoT typically involves six layers: Device Layer, Communication Layer, Network Layer, Middleware Layer, Application Layer, and Business Layer.[18] This multi-layered structure facilitates the seamless integration of devices and data, creating intelligent, responsive systems across various industries and applications.
2. **V2X:** Currently, the integration of V2X communication is gaining significant attention for its role in advancing intelligent transportation systems or ITS.[19] In this type of application, effective collaboration between vehicles within the vehicular environment is crucial. However, this presents a substantial challenge to the underlying communication systems, demanding reliable information transmission and coordination in the presence of latency conditions. Vehicular networks serve a diverse range of applications, including but not limited to data downloading, information dissemination, mobility enhancement, accident alerting, internet access and mobile advertising. These emerging trends underscore the need for efficient as well as robust communication protocols to ensure seamless functioning of intelligent transportation systems. [20]
3. **M2M:** M2M communication, a cutting-edge technology, enables numerous "intelligent devices" to autonomously communicate and collaborate without human intervention, focusing on cost efficiency and time management. Key to M2M's growth is the widespread availability of low-cost, ubiquitous connectivity, driven by affordable, high-speed internet access and global deployment of LTE as well as 3G mobile networks. This surge in IP connected devices such as sensors has given rise to interconnected and interoperable services, transforming daily life. [21] To unlock the full potential of M2M, considerations must address diverse applications, device

functionalities, and other requirements. This includes developing a flexible M2M architecture, ensuring interoperability, preserving information confidentiality and privacy, and maintaining system reliability.[22] These challenges necessitate collaborative international efforts across industries, relying on agreement-based standards to foster continued growth in M2M technologies and markets.

IoT V2X V2V

Figure 1-2: IoT, V2X, V2V

M2M D2D

Figure 1-3: M2M, D2D

4. **D2D:** In the landscape of wireless communication, D2D communication emerges as a paradigmatic approach, allowing adjacent devices to communicate directly without necessitating a centralized network infrastructure. This decentralized mode of communication fosters efficient data exchange, reduced latency, and enhanced resource utilization. D2D communication operates in various modes, including direct communication between devices, device-assisted relaying, and communication facilitated through network assistance. It is particularly instrumental in scenarios demanding localized and swift data sharing, such as collaborative applications, content sharing, and proximity-based services. As a pivotal facet of emerging communication

paradigms, D2D holds promise for improving communication efficiency, reducing congestion, and enabling innovative applications in the era of interconnected devices.

AR VR

Figure 1-4: AR and VR

5. **VR:** VR is a transformative technology that immerses users in simulated environments through specialized headsets. Utilizing advanced graphics, motion tracking, and haptic feedback, VR creates realistic, interactive experiences.[23] From gaming to healthcare, VR applications span diverse fields, offering a glimpse into the future of immersive digital interactions.
6. **AR:** Augmented Reality is a technology that blends digital information such as computer generated data or images, with the real world environment. Unlike virtual reality which creates entirely immersive digital experiences, AR enhances real world experiences by overlaying digital content onto the user's physical surroundings. AR experiences can be accessed through AR-enabled devices such as smart glasses, smartphones, tablets as well as AR headsets.[24]

1.3 Background and Motivation

To summarize, the research presented in this dissertation aims at:

- The evolution from 1G to 5G represents a continuous effort to meet escalating connectivity needs. Here is a short comparison among them via visualization:

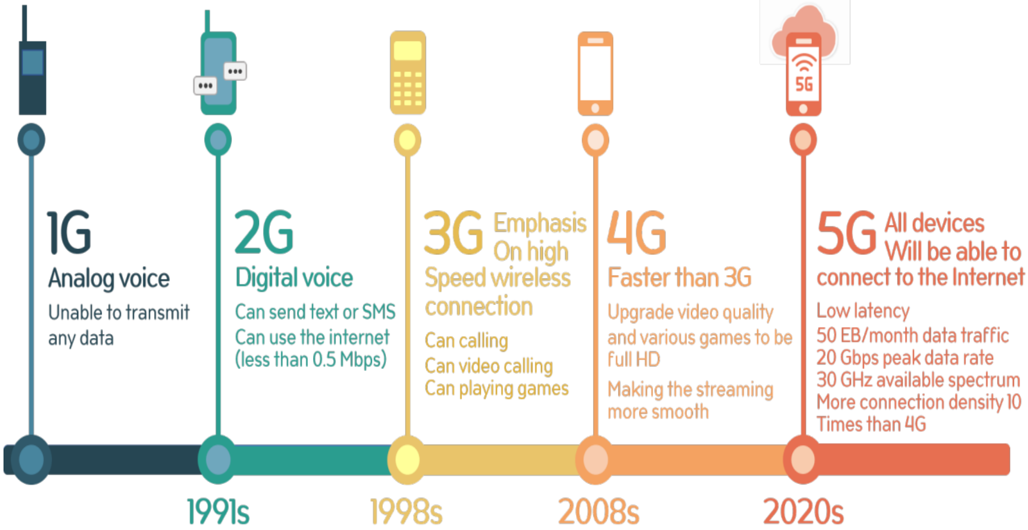


Figure 1-5: Evolution of Cellular Network

- 5G is motivated by the demand for low latency, ultra-fast speeds as well as support for diverse applications.
- Integration of emerging technologies like AI and VR underscores 5G's role as an innovation enabler.
- Industry-specific demands are addressed through features like network slicing, offering customized virtual networks.
- The overarching goal is to enhance the user experience in a technologically advanced and interconnected world.

Chapter 2

Overview of Network Slicing

2.1 Introduction

Network slicing is a transformative concept in the domain of 5G networks, where multiple virtual networks overlay a shared network infrastructure comprising shared network as well as computing resources. Unlike its predecessors, the 5G specification inherently embraces network slicing as fundamental capability, allowing for unprecedented flexibility and resource optimization.

Each network slice operates with its distinct security protocols, logical topology and performance characteristics; all within the constraints of the underlying physical networks. This granularity enables the dedication of slices to specific purposes, such as prioritizing access for particular applications, services, or isolating traffic for specific users or devices. By doing so, network operators can maximize resource utilization and enhance service flexibility.

The origins of network slicing date back to technologies like VLANs on Ethernet networks. However, its full realization has been achieved through the evolution of SDN and software-defined wide area networks (SD-WANs). SDN's separation of the control and data planes allows the definition of virtual networks and the enforcement of packet-handling rules across both physical and virtual devices.[25]

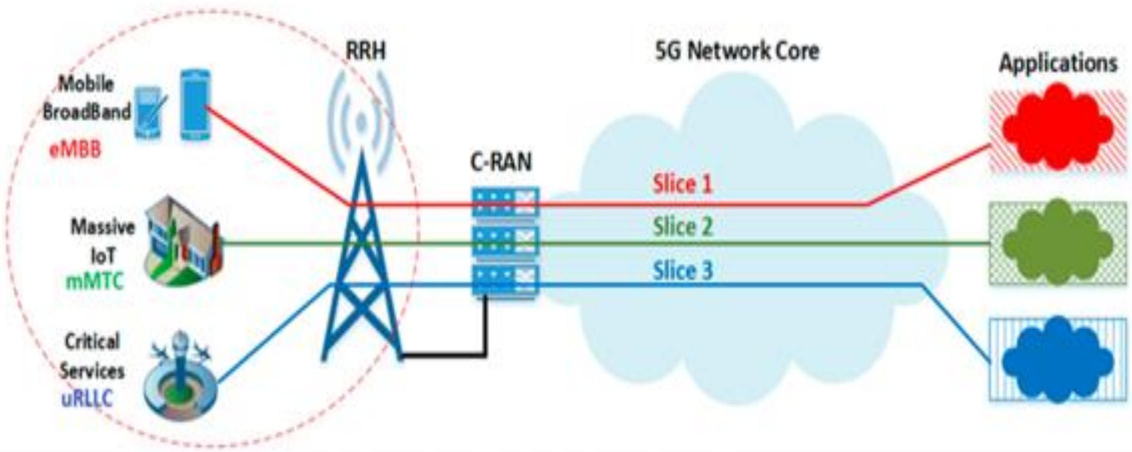


Figure 2-1: Network Slicing

In practical scenarios, network operators may choose to dedicate specific infrastructure, such as assigning a virtual firewall to a particular slice, to meet customer compliance or security requirements. This approach can also optimize network performance by tailoring services to specific needs, exemplified by deploying cost-effective firewalls.

Network slicing operates hand-in-hand with aggregation, allowing the pooling of physical connectivity resources to enhance overall capacity. This agile and efficient approach to network management aligns with the evolving landscape of 5G, marking a paradigm shift in the utilization of network resources.

2.1.1 RB allocation for 5G Downlink

In order to understand resource allocation, we need to visualize how the resource blocks are arranged in 5G Uplink and Downlink. This is done using MATLAB. Here are parameters and corresponding resource grid and channel view obtained via MATLAB for 5G Downlink.

Table 2-1: 5G Downlink parameters

Label	Carrier1
Frequency range	FR1 (410 MHz - 7.125 GHz)
Channel bandwidth (MHz)	60
Cell identity	1
Subframes	10
Initial subframe	0
Windowing source	Custom
Windowing(%)	0
Sample rate source	Auto

Table 2-2: Filtering Configuration for 5G Downlink

Filtering	None

Table 2-3: SCS Carriers for 5G Downlink

Subcarrier Spacing	15kHz	30kHz
Grid Size(RB)	270	132
Grid Start(RB)	3	3

Table 2-4: Bandwidth Parts for 5G Downlink

Subcarrier Spacing	15kHz	30kHz
Cyclic Prefix	Normal	Normal
BWP Size(RB)	270	132
BWP Start(RB)	3	3
Label	BWP1	BWP2

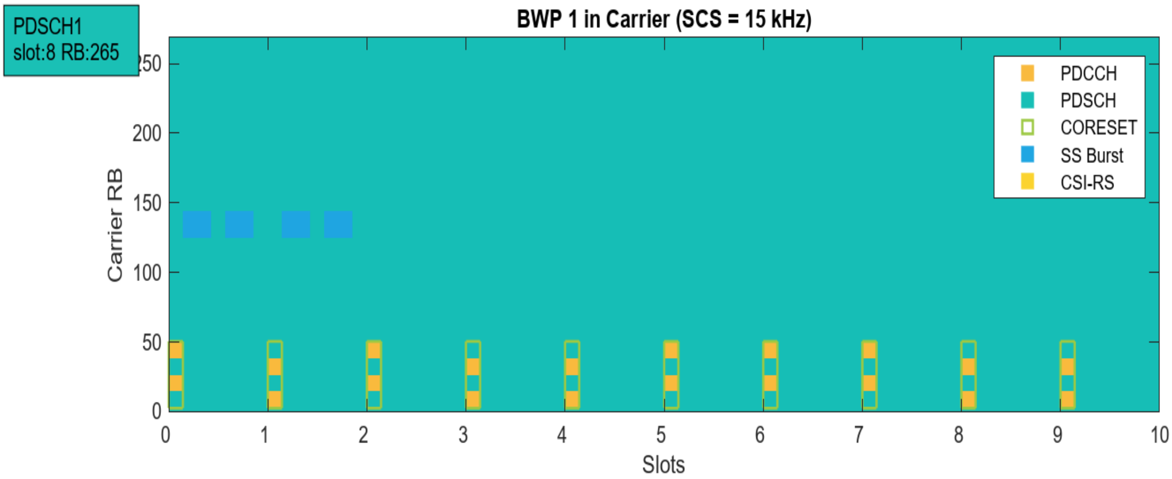


Figure 2-2: Resource Grid (BWP#1) for 5G Downlink

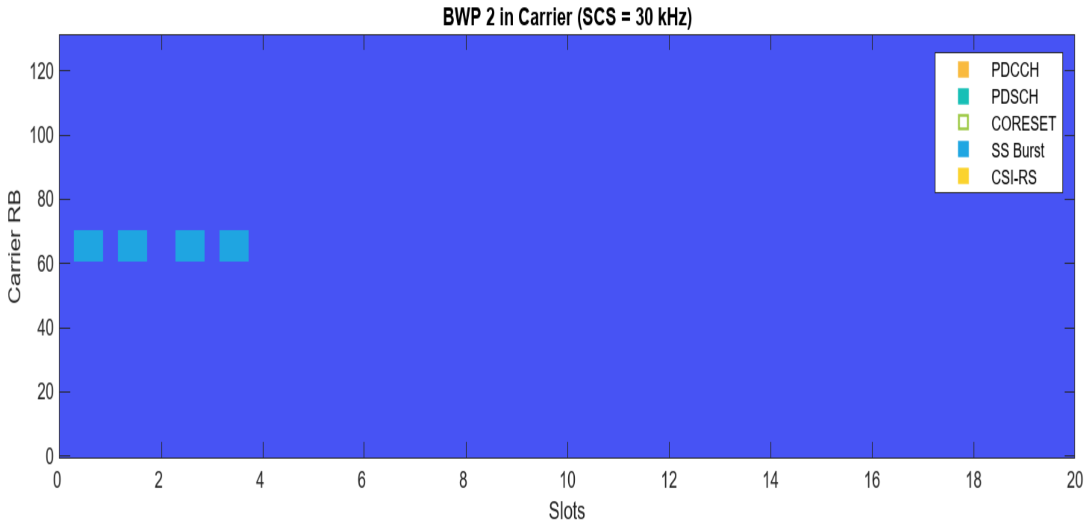


Figure 2-3: Resource Grid (BWP#2) for 5G Downlink

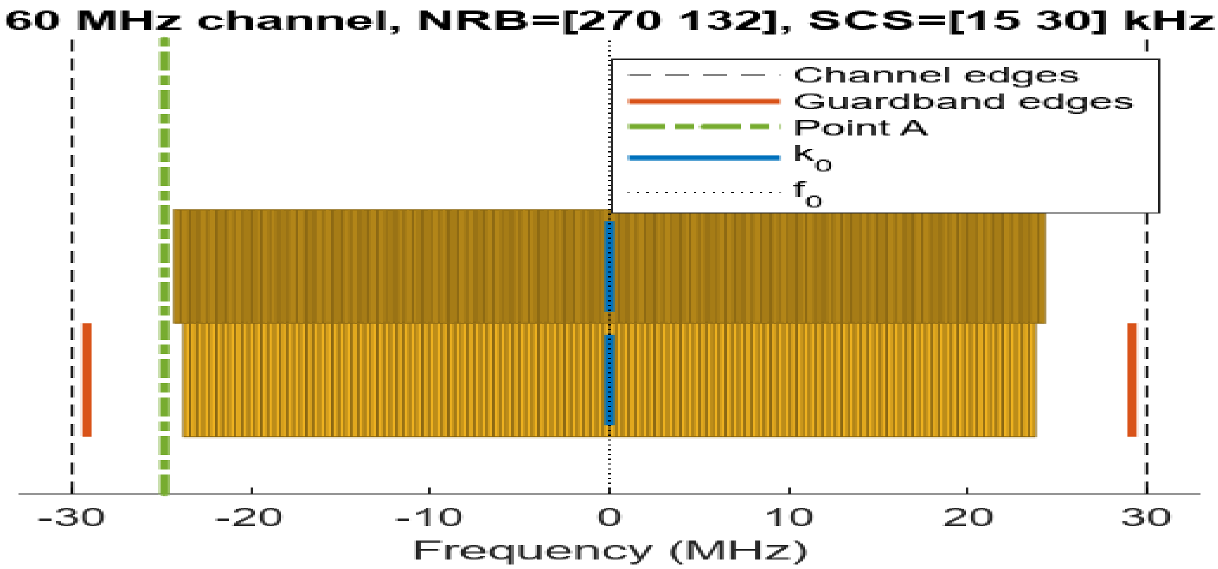


Figure 2-4: Channel View for 5G Downlink

2.1.2 RB allocation for 5G Uplink

Here are parameters and corresponding resource grid and channel view obtained via MATLAB for 5G Downlink.

Table 2-5: 5G Uplink parameters

Label	Carrier1
Frequency range	FR1 (410 MHz - 7.125 GHz)
Channel bandwidth (MHz)	60
Cell identity	1
Subframes	10
Initial subframe	0
Windowing source	Custom
Windowing(%)	0
Sample rate source	Auto

Table 2-6: Filtering Configuration for 5G Uplink

Filtering	None

Table 2-7: SCS Carriers for 5G Uplink

Subcarrier Spacing	15kHz	30kHz
Grid Size(RB)	270	132
Grid Start(RB)	3	3

Table 2-8: Bandwidth Parts for 5G Uplink

Subcarrier Spacing	15kHz	30kHz
Cyclic Prefix	Normal	Normal
BWP Size(RB)	270	132
BWP Start(RB)	3	3
Label	BWP1	BWP2

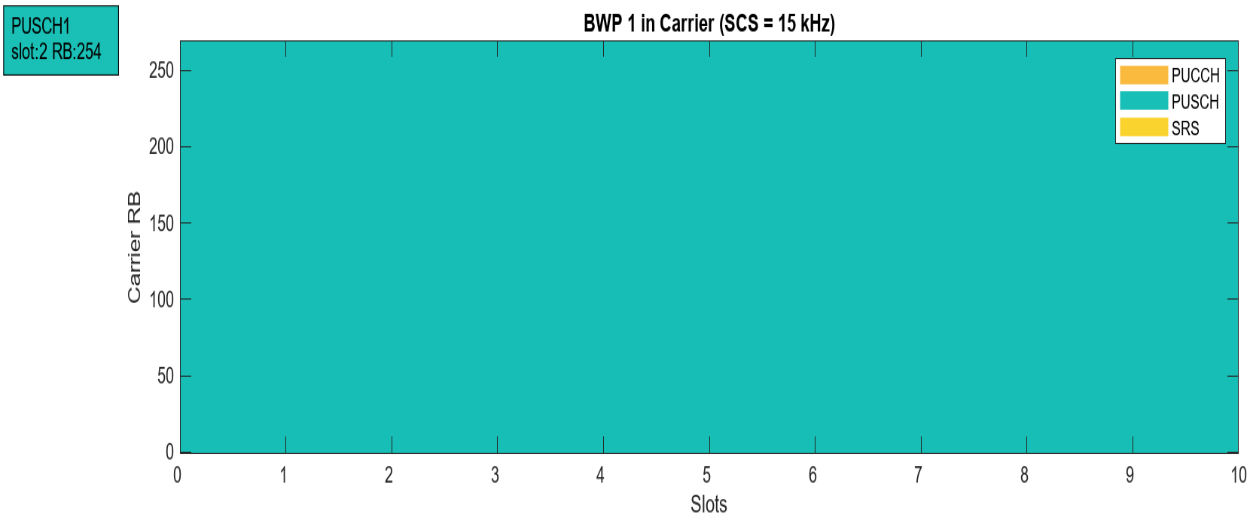


Figure 2-5: Resource Grid (BWP#1) for 5G Uplink

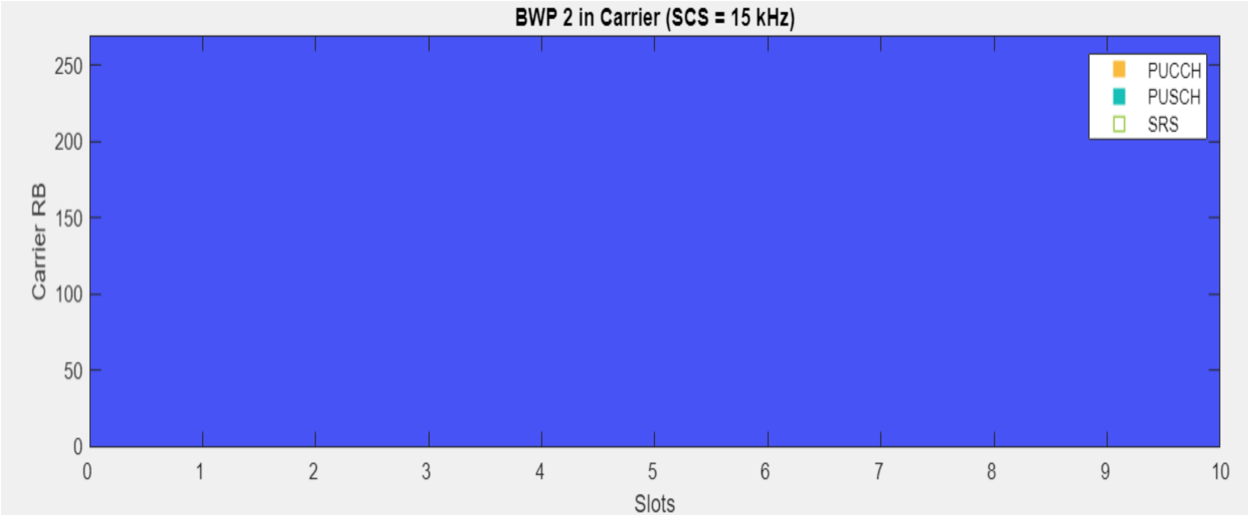


Figure 2-6: Resource Grid (BWP#2) for 5G Uplink

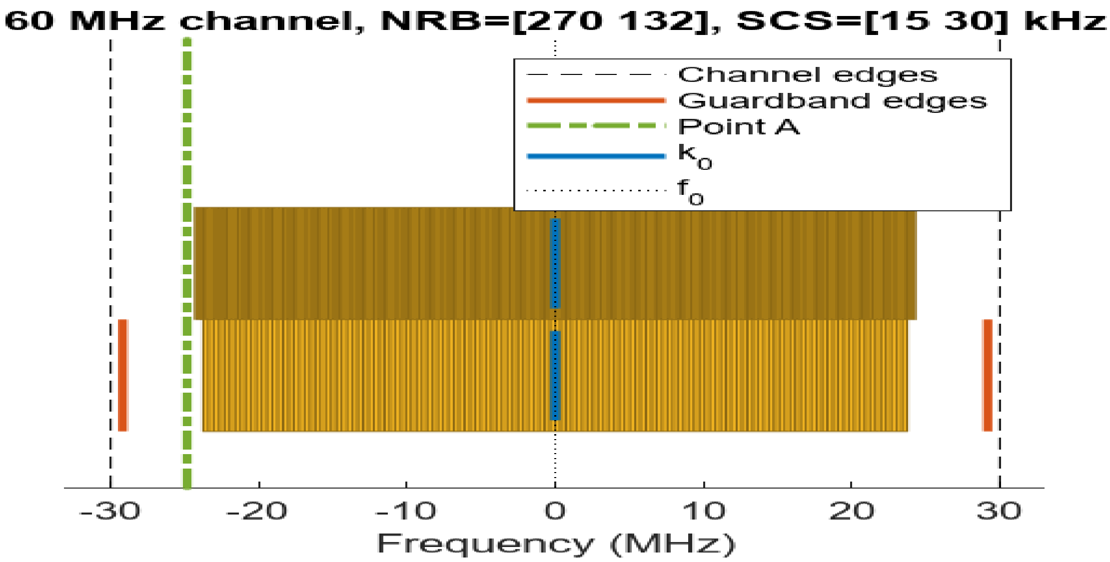


Figure 2-7: Channel View for 5G Uplink

2.2 Importance of Network Slicing

- **Increased number of devices:** The proliferation of connected devices, encompassing everything from smartphones and tablets to IoT sensors as well as smart appliances, has stretched the limits of IPv4 addressing. Each IPv4 address is represented as four sets of decimal numbers separated by dots (e.g., 192.168.155.113). The 32-bit address of IPv4 allows for approximately 4.3 billion unique addresses, which seemed abundant in the early days of the internet but is now insufficient to accommodate the ever growing number of devices coming online globally.

  IPv6 addresses are expressed as eight groups of four hexadecimal digits, separated by colons (e.g., 2001:0db8:85a3:0660:0900:8a2e:0370:7334). With its expansive 128-bit address space, IPv6 offers an incomprehensibly large pool of unique addresses, virtually eliminating the concerns about address exhaustion. Transitioning to IPv6 has become imperative to ensure that all devices, both current and future, can be uniquely identified and connected to the internet without the limitations posed by IPv4's address shortage.[26] In short, resource allocation has become crucial for efficiently allocating resources as the number of devices grows.
- **Conflict in Requirements:** 5G services, designed to cater to a diverse range of applications and use cases, present conflicting demands. For instance, applications such as eMBB require high data rates in order to support bandwidth-intensive activities like high definition video streaming, while Ultra-Reliable Low Latency Communications (URLLC) needs minimal latency to facilitate crucial and critical applications like remote surgery, autonomous vehicles and industrial automation.

  Network slicing emerges as a solution to reconcile these conflicting requirements by enabling the creation of virtualized, isolated network instances optimized for specific service applications or types. Each network slice operates independently, tailored to meet the performance, latency, and resource requirements of its associated service, thereby avoiding conflicts and ensuring optimal resource utilization across the network.[27]
- **Customization for Applications:** Network slicing enables service providers to customize network resources and parameters according to the unique needs of different applications or user groups. By creating dedicated network slices for specific use cases such as augmented reality (AR) or real-time gaming, providers can deliver tailored experiences optimized for each application category.

Customized slices can be fine-tuned in terms of bandwidth allocation, latency thresholds, security protocols, Quality of Service (QoS) parameters as well as other network attributes to accommodate the diverse requirements of applications running on the same physical infrastructure.

- **Resource Efficiency:** Dynamic resource allocation lies at the heart of network slicing, enabling the efficient utilization of network resources in order to changing application requirements, demand patterns and network conditions. Through automated resource management and orchestration mechanisms, network operators can dynamically allocate bandwidth, computing resources, and network capacity based on real-time demand and traffic fluctuations. [27]

  This dynamic provisioning and optimization of network slices help prevent congestion, alleviate network bottlenecks, and ensure equitable resource distribution across different services and applications, thereby enhancing overall network performance as well as user experience.

2.3 Challenges of Network Slicing

1. **Conflicting Requirements:** eMBB services demand high data rates to support bandwidth intensive applications such as ultra-high definition video streaming and virtual reality. On the other hand, URLLC applications require minimal latency and high reliability to support crucial and critical applications like remote surgery, autonomous vehicles and industrial automation. Balancing these conflicting requirements within the same network infrastructure presents a significant challenge, as optimizing for one service type may compromise the performance of the other. [27]
2. **Dynamic Adaptation:** Network slicing necessitates the ability to dynamically adapt to the varying demands of eMBB, mMTC as well as URLLC services in real-time. As network conditions change and new services are introduced, network slices must be capable of quickly adjusting their resource allocations and configurations to meet evolving requirements. Achieving dynamic adaptation requires sophisticated mechanisms for monitoring network performance, analyzing traffic patterns, and reallocating resources accordingly without disrupting ongoing services. [28]
3. **Resource Allocation:** Efficient allocation of network resources is critical to balance the conflicting requirements of eMBB and URLLC services while ensuring optimal

performance for both. Network operators must carefully manage bandwidth, computing resources, and other network resources to avoid under-provisioning or over-provisioning any particular slice. Dynamic resource allocation algorithms are needed to intelligently distribute resources based on real-time demand, traffic patterns, and service priorities, thereby maximizing resource utilization and network efficiency. [28]

4. **QoS Management:** Effective Quality of Service (QoS) management is essential to meet the rigorous latency and reliability needs of URLLC while maintaining satisfactory data rates for eMBB. URLLC applications, such as industrial automation and remote surgery, require ultra-high reliability as well as ultra-low latency to ensure critical operations. QoS mechanisms must be implemented to prioritize URLLC traffic and guarantee timely delivery with minimal packet loss or jitter, even under congested network conditions, without compromising the quality of experience for eMBB users. [29]
5. **Intelligent Algorithms:** Developing intelligent algorithms, potentially leveraging artificial intelligence (AI) as well as machine learning (ML) is crucial for predicting and adapting to changing service requirements in real-time. ML and AI techniques can analyze historical data, monitor network performance metrics, and forecast future demand patterns to proactively optimize network slicing configurations. These algorithms can enable predictive resource allocation, automatic network optimization, and closed-loop adaptation, enhancing network efficiency, resilience, and overall performance. [30]

2.4 Novelty

1. **Dynamic QoS Guarantees:** Network slices in 5G evolve into intelligent entities capable of adjusting resources and configurations in real-time to ensure consistent performance across a spectrum of applications. This adaptability is crucial in maintaining QoS guarantees for diverse 5G applications, where requirements for latency, throughput, and reliability vary significantly. By dynamically adapting resources based on changing network conditions and application demands, network slices can uphold stringent QoS requirements, providing users with a reliable and seamless experience across various services and use cases.

2. **Predictive Resource Allocation:** ML and DL algorithms play a vital role in predictive resource allocation, leveraging historical data as well as real-time analytics to anticipate traffic demands and preemptively allocate resources before congestion occurs. Through predictive resource allocation, 5G networks can maximize network efficiency, optimize resource utilization and most importantly, enhance user experience by ensuring that sufficient resources are available to meet demand spikes and mitigate performance degradation during peak usage periods.
3. **Self-Optimizing Network Management:** Machine learning models enable self-optimizing network management, minimizing the need for manual intervention and human oversight in configuring, managing, and optimizing network slices. By autonomously analyzing network performance metrics, ML models can dynamically adjust slice configurations, allocate resources and optimize network parameters in order to adapt to changing traffic patterns as well as environmental conditions, thereby reducing operational costs and minimizing the risk of human error.
4. **Closed Loop Adaptation:** In a closed loop adaptation framework, network data is continuously fed into ML models, enabling them to learn and adapt to evolving network conditions and user behaviors over time. By leveraging real-time feedback loops, ML models can iteratively refine their algorithms, optimize network performance, and enhance resilience against disruptions or anomalies, ultimately improving overall network reliability and service quality.
5. **Unlocking New Revenue Streams:** Advanced technologies like ML and DL facilitate the developement of customized network slices engineered to niche markets and demanding applications, unlocking new revenue streams and business opportunities for service providers. By offering specialized slices optimized for specific verticals or use cases like industrial IoT, augmented reality, or mission-critical communications, service providers can differentiate their offerings, attract new customers, and capitalize on the growing demand for innovative 5G-enabled services.

Chapter 3

Methodology

3.1 Flowchart

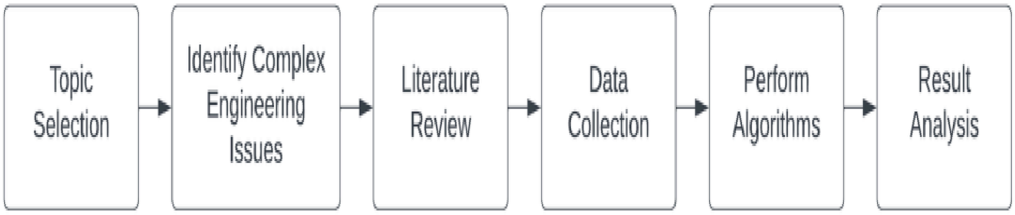


Figure 3-1: Workflow of Project

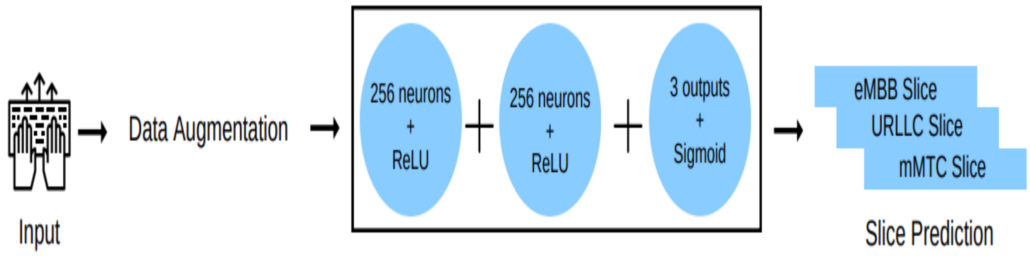


Figure 3-2: Flowchart of Prediction

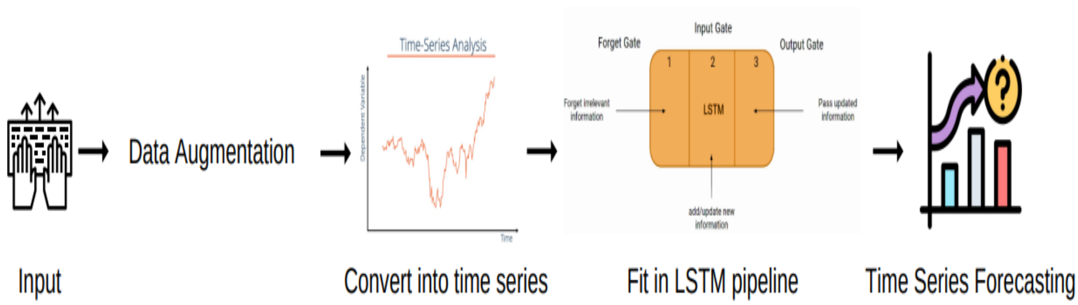


Figure 3-3: Flowchart of Load Scheduling using LSTM

3.2 Literature Review

Since the inception of the network slicing concept in 5G, various researchers tried to address various challenges and issues. Among them, one of the significant topic is automation in network slicing which includes slice predication, forecasting load, improving throughput etc. Kafle et al. proposed ML as suitable candidate for automation of 5G network slicing. The need for efficient resource management, information processing and adaptive network functionality is highlighted, especially in the context of network slicing. They delve into the necessity of automating network functions throughout the lifecycle of network slices, emphasizing the roles of various stakeholders such as infrastructure providers, VNOs and application service providers.[31] Thantharate et al. proposed a system called 'DeepSlice' which incorporated a statistical machine learning model based on a convolutional neural network (CNN) classifier as well as the Random Forest algorithm to manage network load, slice failure conditions, and device type identification and achieved 95% slice prediction accuracy from unknown device type scenario. They employed their model mainly in three scenarios including unknown device type prediction, network slice failure management as well as load balancing. For load balancing, the model redirected traffic to the master slice when individual slices exceeded predetermined thresholds. In the event of network slice failure, the model directed traffic to the master slice to prevent service disruption.[32]

Alsenwi et al. addressed the challenge of scheduling eMBB and URLLC services on the same radio resource in 5G networks. They aimed to improve resource allocation performance in the multiplexing scenario of eMBB-URLLC traffic. They proposed a puncturing scheduling approach to ensure eMBB data rate and reliability while meeting URLLC stringent requirements. They proposed a two stage framework: eMBB resource allocation and URLLC scheduling phases, utilizing DRL algorithms for dynamic scheduling. They also introduced a DRRA algorithm to solve optimization sub-problems effectively. Moreover, they proposed a PGACL algorithm to handle slow convergence in DRL. By leveraging the advantages of DRRA and PGACL algorithms, they achieve reliable and efficient resource allocation. They formulated an optimization aided DRL based framework and satisfied the stringent URLLC reliability while keeping the eMBB reliability higher than 90%. [33]

Haider et al. introduced an efficient network slicing approach for 5G networks using a hybrid learning algorithm. It comprised three phases: data collection, OWFE and slicing classification. Attributes from network devices are collected and OWFE optimizes attribute weighting using GS-DHOA. A hybrid classifier utilizing deep belief and neural networks classifies network slices such as mMTC, eMBB and URLLC for each device. GS-DHOA optimizes weight functions for both networks. Experiment results demonstrate the model's ability to accurately provision 5G network slicing. They achieved highest accuracy of 94.44% for GS-DHOA-NN+DBN algorithm.[34]

Vincenzi el al. proposes a policy based admission mechanism for exclusive intra-service slice allocation in 5G networks, addressing the complexity of resource allocation among multiple service providers. Optimal admission strategies are pre-computed offline for typical network conditions and used to train a neural network (NN). This NN provides near-optimal admission decisions at runtime under new conditions. A proof of concept demonstrates the approach's potential in terms of revenue and quality of service using real network data. The study confirms the feasibility of this approach, highlighting its efficiency, fairness, revenue benefits, and reduced complexity. [35]

Domeke and Cimoli et al. proposed synchronizing hierarchically distributed SDN controllers using RL for edge-enabled network slicing in heterogeneous networks. RL was chosen due to its ability to provide fast near-optimal solutions in scenarios with limited labeled data. Unlike existing methods, the proposed approach synchronized all controllers simultaneously to reduce runtime complexities and energy consumption. Data such as user and device counts, user mobility, link quality, CPU load and memory usage were used for RL implementation. The study aimed to address dependencies among slices and resources, ensure controller security, and handle edge cases like main controller failures. They showed that their model is able to improve the overall accuracy by 6%.[36]

Suh et al. introduced a DRL-based approach for optimizing network slicing in a downlink transmission scenario with eMBB, URLLC, and mMTC slices. They aimed to maximize system throughput while meeting QoS requirements. The approach used DRL to learn optimal resource allocation strategies and addressed the challenge of exploring vast action spaces through action elimination mechanisms. The model was implemented in a scenario with multiple base stations serving user equipment, aiming to dynamically allocate resources across slices to optimize throughput while ensuring QoS. They achieved around 27% and 19% improvements in the throughput performance over the equal allocation in 2022.[37]

Khan et al. proposed a hybrid deep learning model, combining CNN and LSTM networks, to tackle 5G network slicing complexities. CNN managed traffic detection and resource allocation, while LSTM handled load balancing and slice failures. Achieving high accuracy, the model addressed scenarios like unknown requests and resource imbalances, finding applications in IoT, internet security, and wireless localization. It optimized resource allocation in a Mesh UAV network, ensuring reliability and high throughput. Using TensorFlow, Keras, and NS2, a simulation model tested the performance in a Mesh UAV network, assessing request duration, packet loss, and delay. They achieved an overall accuracy of 95.17% in 2022.[38]

Vijayalakshmi *et al.* addressed the demands of modern communication networks in 2023, including low latency, high reliability, and increased capacity by using a machine learning based network slicing algorithm. They developed and evaluated a Bagging Gaussian Naive Bayes algorithm to classify devices into these slices. The B-GNB algorithm is shown to predict the best possible network slice even during network interruptions, with performance metrics like F-score, precision, sensitivity as well as accuracy analyzed. Comparative analysis shows B-GNB achieves an 86% classification accuracy. [39]

Venkatapathy et al. developed a fast and secure network slicing technique for 5G networks using PROMETHEE-II and SLE algorithms for node allocation and connection establishment. PROMETHEE-II ranks nodes based on characteristics like capacity and bandwidth, while SLE ensures optimal link configurations for network slice requests. Performance is evaluated using service revenue and acceptance ratio and simulations show the benefits of a small-world network structure for 5G.[40]

Table 3-1: Summary of Literature Review

Author	Concept	Algorithm	Objective	Highest result
Thantharate et al.	Network Slicing	RF+CNN	Slice prediction	95%
Alsenwi et al.		Optimization +DRL	Reliability	90%+
Haider et al.		ML+DL	Slice prediction	94.44%
Domeke et al.		ML	Slice prediction	6% increase
Suh et al.		DRL	Throughput increase	27%
Khan et al.		CNN+LSTM	Slice prediction	95.17%
Vijayalakshmi		B-GNB	Slice prediction	86%

3.3 Experimental Setup

3.3.1 Dataset and Tools Used

The dataset was collected from *IEEE dataport* which contains 4,66,739 samples and is used for resource allocation of a 5G network. It includes nine attributes that capture various aspects of network usage and conditions. These attributes are: LTE/5G UE Category, Use Case Type, Technology Supported, Day, Time, QCI (Quality of Service Class Identifier), Packet Loss Rate (Reliability), Packet Delay Budget (Latency) and Slice Type. The last attribute, Slice Type is the target variable. The dataset provides detailed information on the type of use case (e.g., Smartphone), the technology in use (LTE/5G), the time and day of data capture, as well as critical performance indicators like packet loss rate, QCI as well as latency requirements. The target variable, Slice Type, indicates the specific slice of the network (e.g., eMBB, mMTC) allocated under given conditions. This rich dataset can be utilized for predictive analytics and optimizing resource allocation strategies in 5G networks. We will use python and Google colab for buiding our models

3.3.2 Importing Libraries & Read CSV File

The following libraries are used in our code and the csv File was read:

1. **Pandas**: Pandas is a python library which is used for data analysis as well as manipulation. It is a very powerful and flexible tool and is broadly used in machine learning tasks as well as artificial intelligence. Some of its functions used in our task are: [41]
   a) Reading CSV files by using 'pd.read_csv()' function
   b) Data cleaning and pre-processing by using 'df.dropna()' and 'df.fillna()' functions to drop and fill missing values respectively.
   c) Getting descriptive statistics of the total dataset by using 'df.describe()' and 'df.info()' functions
   d) Feature engineering and selection can be done effectively using the column and row manipulation techniques of the pandas library.
2. **Numpy**: Numpy is one of the most fundamental and extensively used python libraries used in artificial intelligence and machine learning tasks. Some of its functions used in our task are: [42]
   a) Creating multi-dimensional arrays to store and manipulate data from the main dataset.
   b) Indexing and slicing for efficient data selection and manipulation.

c) Many mathematical operations can be done such as mean, median, mode and it also allows element wise operations.

3. **Seaborn**: Seaborn is a python visualization library. It is a powerful tool which uses high level interface for drawing statistical graphics which is presented attractively with a lot of informative applications. Some of its functions used in our task are: [43]

   a) Visualization of data distributions with functions like ‘sns.histplot()’ and ‘sns.boxplot()’. These plots help in visualizing and understanding the distribution of the dataset and also identifying outliers.

   b) We can create correlation matrices with the ‘sns.heatmap()’ function. This helps us understanding relationship between different features.

   c) Seaborn is very important in the feature engineering task. Both categorical data and multi-plot grids can be presented using the ‘sns.barplot()’ and ‘sns.facetgrid()’ functions respectively.

4. **Matplotlib**: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for 2D graphics and has a variety of plotting functions to create various types of charts and plots. Some of its functions used in our task are: [44]

   a) It can be used for plotting basic graphs with functions like ‘plt.plot()’, ‘plt.scatter()’, ‘plt.bar()’, and ‘plt.hist()’. These were used to create basic scatter plots, line plots, bar charts and histograms respectively.

   b) We can customize plots using various functions to set titles with functions like ‘plt.title()’, labels with ‘plt.xlabel()’, ‘plt.ylabel()’ and legends with functions like ‘plt.legend()’. Apart from these, various other attributes like colors, line styles, and markers can be introduced in the code.

   c) The plt.show() function can be used to display different types of plots.

   d) Although primarily created using Seaborn (which is built on top of Matplotlib), we can use Matplotlib functions to further customize heatmaps, such as adding color bars and adjusting figure size.

5. **Sklearn**: Scikit-learn is a powerful and efficient tool for data mining and data analysis. It is built on NumPy, Matplotlib, SciPy and provides a variety of unsupervised as well as supervised learning algorithms. Some of its functions used in our task are: [45]

   a) Sklearn can be used for preprocessing the data which includes functions for standardization, ‘StandardScaler’, encoding categorical variables like ‘LabelEncoder’ and ‘OneHotEncoder’ and handling missing values ‘SimpleImputer’.

b) Model Selection can be done with the library as it provides tools for splitting data with 'train_test_split', cross-validation with 'cross_val_score', and parameter tuning with 'GridSearchCV'.

c) Classification and regression models can be used as he library offers various algorithms like 'RandomForestClassifier', 'DecisionTreeClassifier', 'KNeighborsClassifier', and more for building models.

d) Models can be easily evaluated as it includes metrics for evaluating model performance, such as 'accuracy_score', 'classification_report', 'confusion_matrix', and others.

6. **Xgboost**: XGBoost (Extreme Gradient Boosting) is an efficient as well as scalable implementation of gradient boosting algorithms, designed for speed and performance. It is widely used in competitive machine learning because of its high performance and flexibility. Some of its functions used in our task are: [46]

a) Model training can be done with the 'XGBClassifier' and 'XGBRegressor' classes, which are used for classification and regression tasks, respectively.

b) Hyperparameter tuning can be done as XGBoost provides a range of hyperparameters that can be tuned to optimize model performance.

c) Model evaluation can be done similar to other machine learning libraries, because XGBoost integrates well with Scikit-learn's functions for model evaluation and cross-validation.

7. **Tensorflow**: TensorFlow is an open source library for numerical computation and large scale ML. It is particularly well-suited for building and training deep learning models. 'Keras', which is included with 'TensorFlow', provides a high-level API for building and training neural networks. Some of its functions used in our task are: [47]

a) Neural networks can be built using the sequential model and functional API in 'Keras', which allows for easy construction of neural networks.

b) 'TensorFlow' provides various types of layers such as 'Dense', 'Conv2D', 'LSTM', etc., to build different kinds of neural networks.

c) The compile method configures the model with loss functions, optimizers, and metrics required for various functions according to the needs.

d) The fit method trains the model on the dataset.

e) The evaluation method assesses the model's performance on a test dataset.

f) The library also includes useful callbacks like 'EarlyStopping' and 'LearningRateScheduler' to enhance training processes.

3.3.3 Data Augmentation

Data augmentation is a method by which various techniques are applied to artificially increase the size of the dataset. In our code, random missing values are introduced into the dataset to simulate real-world scenarios where data might be incomplete or missing. [48] This is done by creating a random mask and setting a percentage of data points to NaN (Not a Number) based on this mask. This process helps in training models that are robust to missing data. For this purpose, we used the 'np.random()' function to introduce random values in the dataset. We introduced 7% data in our case and this helps in the following cases:

- **Model Robustness**: Training models that can handle missing data effectively. This is important for real-world applications where perfect data is rarely available.
- **Data Imputation Techniques**: Evaluating different imputation techniques to fill in missing values, which is crucial for maintaining data integrity and improving model performance. [49]
- **Better Generalization**: Ensuring that the models do not overfit to complete data as well as can generalize well to unseen data that may have missing values.[50]

3.3.4 Pre-processing

Pre-processing steps are essential to prepare and remove noise the data for analysis. This includes renaming columns for better readability, handling missing values and normalizing or scaling data if necessary. [51] In our code, columns are renamed to simpler names. After adding missing values, the code checks the unique values in categorical columns to understand the data better.

3.3.5 Encoding

Encoding involves converting categorical variables into numerical formats that can be used by machine learning algorithms. [52] In this code, the function 'pd.get_dummies()' is used to perform one-hot encoding on categorical columns. This process converts categorical values into binary vectors, making it easier for the machine learning models to interpret the data.

3.3.6 Performance metrics of Network slicing

Here are some important performance matrices that are often used in Network slicing

1. **Resource Allocation:**

$R_i = \frac{B_i}{N_i}$

where R_i is the resource allocated to slice i, B_i is the available bandwidth for slice i, and N_i indicates the number of users in slice i.

2. **Quality of Service (QoS):**

$$QoS_i = \alpha \cdot Data\ Rate_i - \beta \cdot Latency_i + \gamma \cdot Reliability_i$$

where α, β and γ are weight factors for data rate, latency, and reliability respectively.

3. **QoS Constraints:**

Quality of Service for a network slice can be defined in terms of latency, bandwidth as well as reliability. These constraints can be expressed as:

$$Latency_i \leq L_i$$

$$Bandwidth_i \geq B_i$$

$$Reliability_i \geq R_i$$

Where,

L_i is the maximum allowable latency for slice i,

B_i is the minimum required bandwidth for slice i,

R_i is the minimum required reliability for slice i.

4. **Network Slicing Efficiency:**

$$Efficiency = \frac{Total\ Throughput}{Total\ Resources}$$

5. **Erlang B Formula**

For calculating the blocking probability in a system with limited resources:

$$B(E, m) = \frac{\frac{E^m}{m!}}{\sum_{k=0}^{m} \frac{E^k}{k!}}$$

Where, E is the offered traffic in Erlangs and m is the number of servers. [53]

3.3.7 Parameter Selection

Choosing the right parameters for a model is pivotal for achieving optimal performance. Each parameter influences the model's generalization ability, learning process as well as computational efficiency. Below, a detailed justification for each parameter setting used in the proposed framework, ensuring they align with the best practices and the specific needs of the task is provided:

1. **Missing Percentage:** Introducing a 7% missing data rate helps the model to learn how to handle incomplete data during training, which is common in real-world scenarios. This percentage is low enough not to overly disrupt the data, yet sufficient to teach the model to deal with such situations. Optimal Range is 5% to 15%.[48] The exact value can be adjusted based on the amount of missing data expected in the actual use case.
2. **Test Size:** Reserving 20% of the data for testing ensures that there is enough data to reliably evaluate the model's performance while keeping 80% for training to learn effectively. Optimal Range is 15% to 30%. Larger datasets might allow for a smaller test size, while smaller datasets might need a larger test size to ensure the test set is representative.
3. **Random State:** The random state ensures reproducibility of the results by fixing the seed for random number generation. This specific value of 77 is arbitrary but important for consistent results. Optimal Range is any fixed integer value (e.g., 42, 77, 123). The actual number chosen is less important than the fact that it is fixed.
4. **Activation:** ReLU activation helps the model learn complex patterns by introducing non-linearity, and it avoids the vanishing gradient problem, making training faster and more effective. Alternatives like 'tanh' or 'leaky relu' might be explored. 'tanh' can be useful for data that is normalized between -1 and 1, while 'leaky relu' can help in cases where ReLU might lead to dead neurons.[54]
5. **Initial Learning Rate:** An initial learning rate of 0.001 is a standard starting point for the Adam optimizer, balancing convergence speed and stability. Optimal Range is 0.0001 to 0.01. Lower values might lead to slower but more stable convergence while higher values can speed up training but risk overshooting minima.
6. **Learning Rate Decay Factor:** Gradually decreasing the learning rate by a factor of 0.9 helps in fine-tuning the weights and improves the chances of finding a better minimum by reducing the step size over time. Optimal Range is 0.8 to 0.99. This factor can be adjusted based on the observed training dynamics.

7. **Optimizer:** The Adam optimizer combines the merits of both RMSprop and SGD with momentum. It adapts the learning rate for each parameter, making it highly effective and widely used. While Adam is typically a good default, alternatives like RMSprop, SGD with momentum or even AdamW (a variant of Adam) might be considered depending on specific needs.[55]
8. **Loss Function:** Mean Squared Error is ideal for regression tasks because it penalizes larger errors more heavily, leading to better overall performance in predicting continuous values. For regression, MSE is optimal. For other tasks such as binary or multi class classification, loss functions like binary cross entropy or categorical cross entropy would be more appropriate.
9. **Call backs:** Early stopping rigorously monitors validation performance and stops training when improvement ceases; preventing overfitting and saving computational resources. [56] Early stopping is generally beneficial, though the specific parameters such patience and min delta should be tuned based on the dataset and training behavior.
10. **Patience:** Allowing 20 epochs without improvement before stopping gives the model a chance to overcome temporary plateaus in performance. Optimal Range is 10 to 50. Smaller patience values might stop training prematurely, while larger values might lead to unnecessary overfitting.
11. **Min Delta:** A minimum change of 0.0001 in validation loss is considered an improvement, allowing the model to stop if no significant improvement is observed. Optimal Range is 0.0001 to 0.01. This should be small enough to detect meaningful improvements but not too small to be affected by noise.
12. **Validation Split:** Using 20% of the training data for validation helps in fine-tuning hyperparameters and monitoring model performance without heavily impacting the training set size. Optimal Range is 10% to 30%. This depends on the total dataset size, where larger datasets can afford a smaller validation split.
13. **Batch Size:** A batch size of 512 is a good balance between computational efficiency and stable gradient estimates. It allows for faster training on modern GPUs. Optimal Range is 32 to 1024. Smaller batches provide more frequent updates but can be noisy, while larger batches are more stable but require more memory and computational power. [57]
14. **Epochs:** 200 epochs provide enough opportunity for the model to converge. Early stopping will prevent unnecessary training if convergence is achieved earlier. Optimal Range is 50 to 500. The actual number depends on how quickly the model converges, which can be monitored using early stopping.

15. **Scaler:** MinMaxScaler scales features to a fixed range, usually [0, 1], which is ideal for neural networks as it speeds up convergence and prevents dominance of certain features. MinMaxScaler is often a good default. StandardScaler might be used if the data follows a Gaussian distribution, scaling to unit variance and zero mean.
16. **Sequence Length:** A sequence length of 10 captures short-term dependencies in data, making it suitable for time series or sequence prediction tasks. [58] Optimal Range is 5 to 50. Depending on the temporal dependencies in the data, longer sequences might be necessary to capture more extended patterns, but at the cost of increased computational complexity.
17. **LSTM Units:** 50 LSTM units offer a balance between model complexity and computational cost, allowing the model to capture temporal dependencies without overfitting. Optimal Range is 20 to 200. More units increase the model's capacity to learn complex patterns but also the risk of overfitting and the computational cost.[59]

Table 3-2: List of Parameters

Name	Value
Missing Percentage (Data Augmentation)	0.07
Test size	0.2
Random state	77
Activation	'relu'
Initial learning rate	0.001
Learning rate decay factor	0.9
Optimizer	'adam'
Loss function	'mse'
Call backs	Early stopping
Patience (Early stopping)	20
Min delta (Early stopping)	0.0001
Validation split	0.2
Batch size	512
Epochs	200
Scaler	MinMaxScaler()
Sequence length	10
LSTM Units	50

3.4 Part-1: Slice Prediction

3.4.1 Procedures

The following steps are performed for slice prediction:

1. **Test-Train Split**: Our dataset is split into testing and training sets to evaluate the performance of the model. The 'train_test_split' function from 'sklearn.model_selection' is used for this purpose. This function splits the data into two parts: one part is used to train the model, and the other is used to test the model's performance on unseen data. This helps in assessing the generalization capability of the model.
2. **Model Pipeline Building**: Building a model pipeline involves defining the sequence of steps that will be executed to train and evaluate the model. This includes data preprocessing, model training as well as evaluation. Various machine learning models such as 'RandomForestClassifier', 'DecisionTreeClassifier', 'KNeighborsClassifier' and 'XGBoost' are imported for building different models. For deep learning, 'tensorflow' and 'keras' are used to create neural networks.
3. **Visualization**: Visualization is crucial for understanding data and model performance. Libraries like matplotlib and seaborn are used to create various plots. In our code, a correlation matrix is computed and visualized using a heatmap to understand the relationships between different features. Visualization helps in identifying patterns, trends, and potential issues in the data. The following visualization is used in our work:
    - **Heatmap**: A heatmap is a graphical representation of data where values are depicted by color. In our code, a heatmap is used to visualize the correlation matrix, showing the correlation coefficients between different features in the dataset. This helps in identifying highly correlated features which might be redundant for the model.
    - **LR**: Learning rate is a hyperparameter that controls how much the model's weights are adjusted with respect to the loss gradient. It's crucial in training neural networks as it affects the convergence of the model. The 'LearningRateScheduler' from 'keras.callbacks' is used to adjust the learning rate during training.
    - **ACC**: Accuracy is a metric used to evaluate the performance of a classification model. It is the ratio of correctly predicted instances to the total instances. The 'accuracy_score' function from 'sklearn.metrics' is used to calculate the accuracy of the model.

- **Classification Reports**: A classification report provides detailed metrics about the performance of a classification model, including precision, recall, F1-score, and support for each class. The 'classification_report' function from 'sklearn.metrics' is used to generate this report, which helps in understanding how well the model performs for each class in the target variable.

3.4.2 CNN Model Evaluation

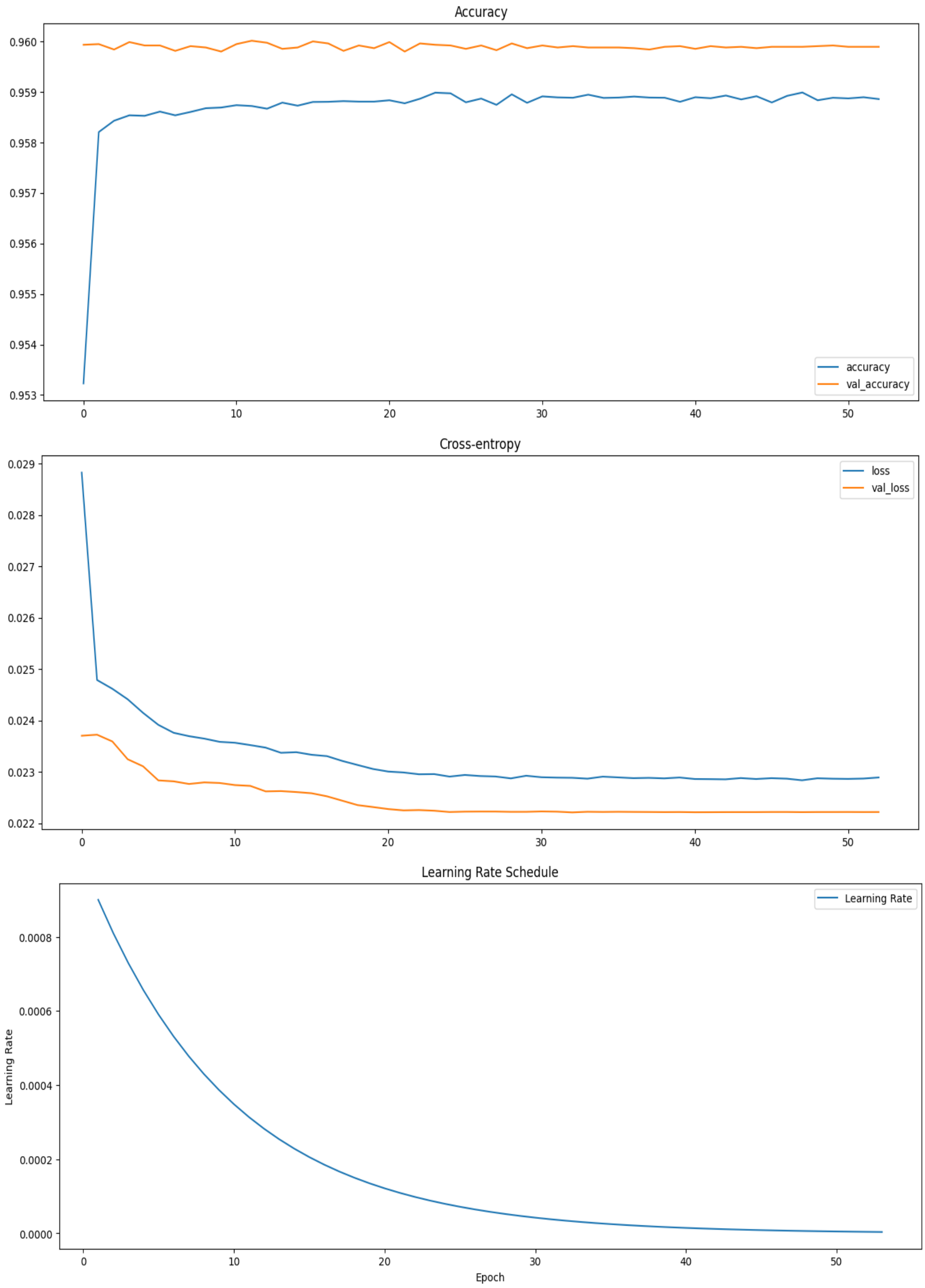


Figure 3-4: CNN Model Evaluation

3.4.3 Confusion matrix and outcomes

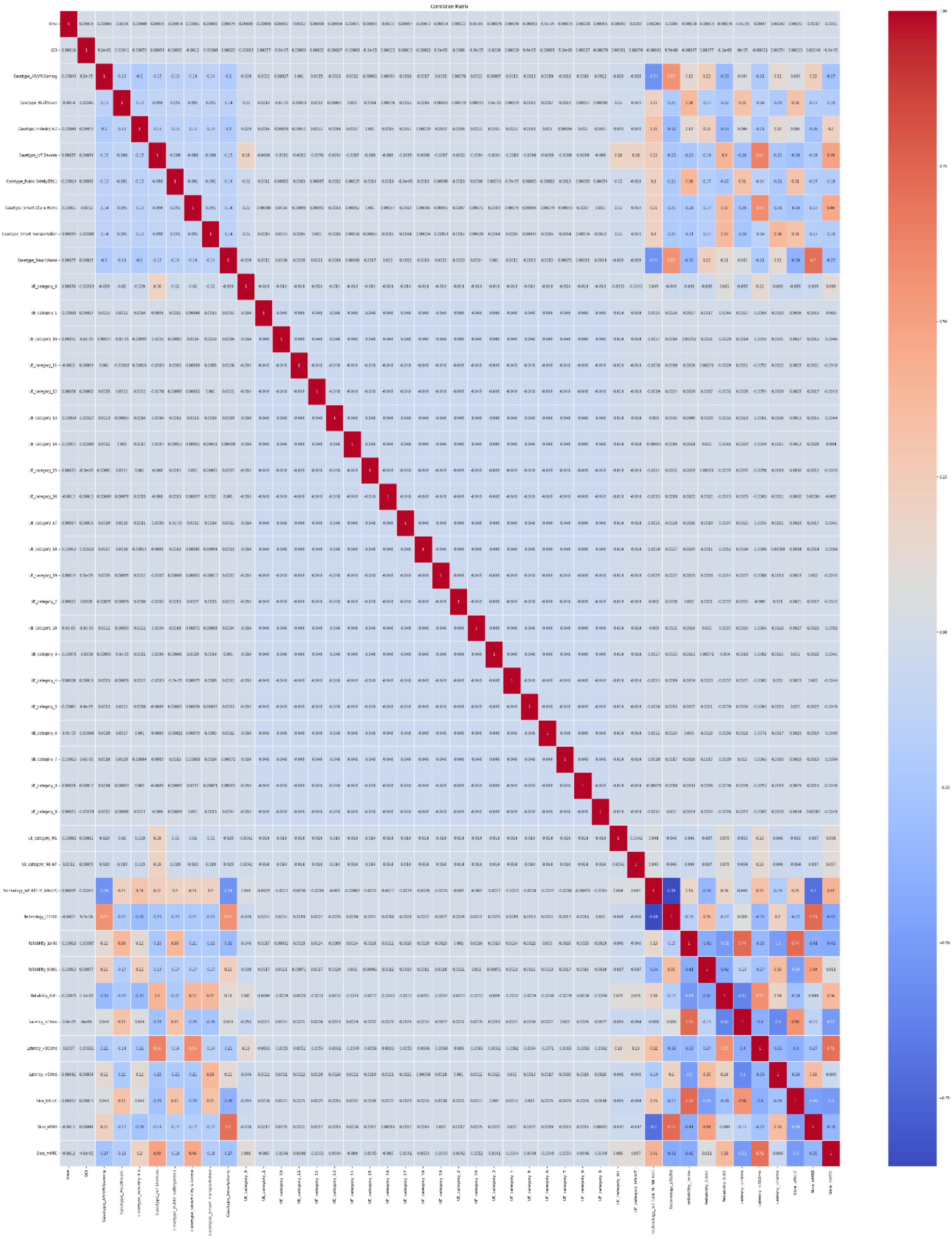
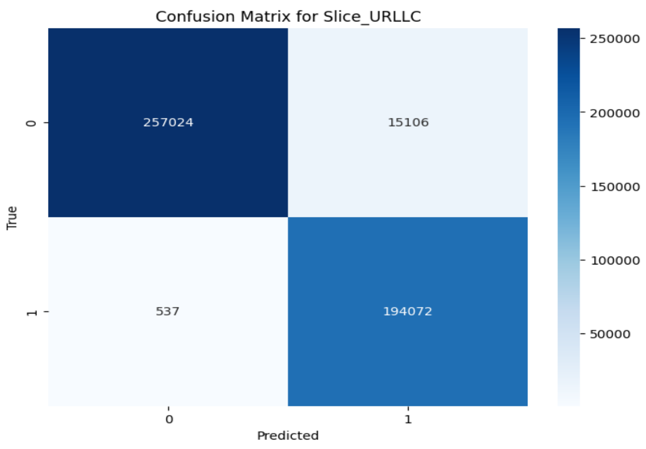
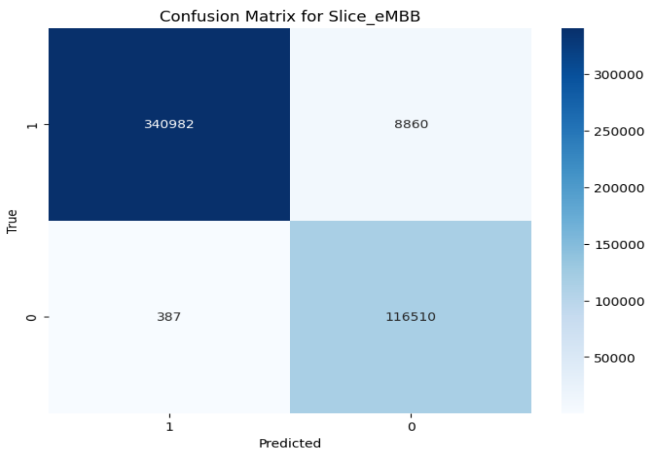


Figure 3-5: Correlation Matrix



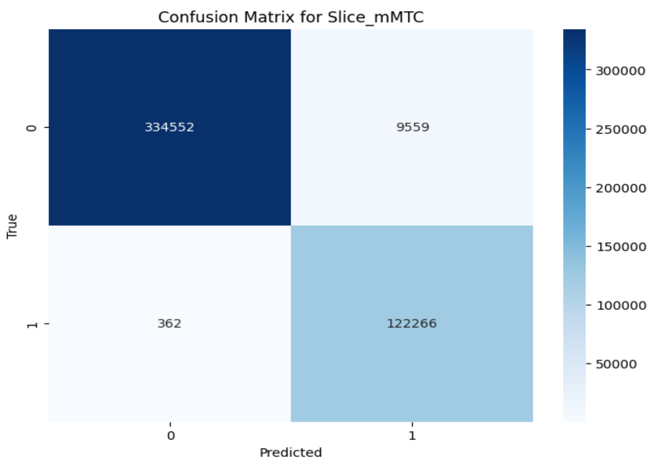
Classification Report for Slice_URLLC:

	precision	recall	f1-score	support
0	1.00	0.94	0.97	272130
1	0.93	1.00	0.96	194609
accuracy			0.97	466739
macro avg	0.96	0.97	0.97	466739
weighted avg	0.97	0.97	0.97	466739



Classification Report for Slice_eMBB:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	349842
1	0.93	1.00	0.96	116897
accuracy			0.98	466739
macro avg	0.96	0.99	0.97	466739
weighted avg	0.98	0.98	0.98	466739



Classification Report for Slice_mMTC:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	344111
1	0.93	1.00	0.96	122628
accuracy			0.98	466739
macro avg	0.96	0.98	0.97	466739
weighted avg	0.98	0.98	0.98	466739

Figure 3-6: Confusion Matrix and Classification Report for each cases

3.4.4 Result Analysis

Since we have used data augmentation, the model has become more robust and therefore the results are more realistic in the classification report.

The accuracy of the prediction of our model is = 97%

3.5 Part-2: Load Forecasting

3.5.1 Procedures

LSTM networks are a sophisticated type of RNN designed to address the challenges of time series forecasting. Traditional RNNs struggle with long-term dependencies due to the vanishing gradient problem, limiting their ability to retain information over extended periods. LSTMs overcome this with a unique cell architecture that includes forget, input, and output gates. These gates control information flow, enabling the network to maintain and update memory states efficiently, capturing long-term dependencies crucial for accurate forecasts.

LSTMs effectively handle non-stationary data, which typically exhibit dynamic, evolving patterns. Their adaptive learning capabilities allow them to adjust to shifts in data patterns, making them effective for applications where data properties change over time, such as financial markets or climate analysis.[59] This adaptability ensures robust and accurate forecasts even with complex, evolving data.

LSTMs also demonstrate remarkable robustness to noise, a common issue in time series data. The gating mechanisms in LSTM cells filter out irrelevant noise, allowing the network to focus on significant signals, enhancing prediction accuracy and reliability.[60] This makes LSTMs ideal for applications with inconsistent or noisy data, such as sensor data or economic indicators.

Furthermore, LSTMs excel in modeling complex temporal patterns, including seasonality, trends and cyclic behaviors.[61] Their recurrent architecture and ability to learn hierarchical temporal dependencies enable them to capture intricate patterns effectively. Additionally, LSTMs can incorporate exogenous variables—external factors influencing the time series—further enhancing their predictive power for comprehensive, contextually aware forecasting models.[62]

The following steps are performed for the time series forecasting for URLLC, mMTC, eMBB Slice respectively:

1. **Reset the Index and Define the Target Variable:** Resetting the index of a DataFrame is an essential preprocessing step. When working with pandas, operations like merging, filtering, or data augmentation can result in disordered indices. By resetting the index using df_encoded.reset_index(), we ensure that the DataFrame has a clean and sequential index. This step is particularly important before any operations that rely on the index, such as iterative processes or time series analysis, to maintain data integrity and avoid indexing errors. Here the parameter is df_encoded.

In the context of supervised learning, defining the target variable is crucial. The target variable, Slice_eMBB in this case, represents the output that the model is trained to predict. Identifying and isolating this variable allows us to construct the dependent and independent variables required for training. This step sets the stage for subsequent preprocessing, scaling, and model training, as the model needs to know what to predict based on the input features.

2. **Normalize the Data Using Min-Max Scaling:** Normalization is a key preprocessing step, especially in machine learning algorithms that are sensitive to the scale of input data, such as neural networks. Min-Max scaling transforms features to lie within a specified range, typically between 0 and 1. This is achieved by subtracting the minimum value and dividing by the range (max - min). The code uses MinMaxScaler to normalize the target variable. This transformation ensures that all input features contribute equally to the model training, preventing features with larger ranges from dominating those with smaller ranges. Normalized data often leads to faster convergence and improved model performance.
3. **Convert the Data to a Time Series Sequence:** Time series analysis requires data to be structured in sequences that reflect temporal dependencies. In this step, the normalized data is converted into overlapping sequences of a specified length. In our work we used a sequence length of 10, each sequence contains 10 consecutive data points, and the model uses these sequences to predict the next data point. This sliding window approach generates multiple input-output pairs from the original data. The code appends these sequences to lists 'X' and 'y', which are then converted to numpy arrays. This transformation prepares the data for training models like LSTMs, which are designed to capture temporal patterns in sequential data.
4. **Split the Data into Training and Testing Sets:** Splitting the dataset into training and testing sets is a fundamental step in machine learning. The training set is used to train the model, while the testing set evaluates its performance on unseen data. This ensures that the model's performance metrics reflect its generalization ability, not just its ability to memorize the training data. The code uses 'train_test_split' with a test size of 20%, meaning 80% of the data is used for training, and 20% for testing. The 'shuffle=False' parameter is critical for time series data, preserving the temporal order and ensuring that the model learns from past data to predict future values. This split provides a robust framework for model validation and performance assessment.

5. **Building the LSTM Pipeline:** LSTM networks are a type of RNN particularly effective for sequence prediction problems. In building the LSTM pipeline, the code constructs a sequential model using Keras. The model comprises an LSTM layer with 50 units and ReLU activation, followed by a Dense layer with a single unit. The LSTM layer captures long-term dependencies in the data, making it suitable for time series forecasting. The Dense layer maps the LSTM outputs to the desired prediction. This architecture is designed to learn from past sequences and make accurate predictions, leveraging the LSTM's ability to remember information over extended periods.
6. **Define a Simple Learning Rate Schedule:** A learning rate schedule dynamically adjusts the learning rate during training, which can enhance the model's convergence and performance. Here in our case, `lr_schedule` function defined in the code decreases the learning rate by a factor of 0.9 every epoch, starting from an initial rate of 0.001. This gradual reduction helps in fine-tuning the model, allowing it to make smaller adjustments as training progresses. This strategy can prevent overshooting the optimal solution and help the model settle into a better minimum of the loss function. A well-designed learning rate schedule can significantly improve the training process and final model accuracy.
7. **Create a LearningRateScheduler Callback:** The LearningRateScheduler callback in Keras implements the learning rate schedule during model training. By passing the 'lr_schedule' function to the 'LearningRateScheduler', the learning rate is adjusted at the start of each epoch according to the predefined schedule. This callback ensures that the learning rate decreases progressively, which can help in achieving a more stable and effective training process. Using a learning rate scheduler can lead to better convergence rates and improved model performance by fine-tuning the learning rate over time.
8. **Model Evaluation & Visualization:** Evaluating and visualizing the model's performance is crucial for understanding its behavior and effectiveness. After training, the code converts the training history into a DataFrame and plots the loss and accuracy metrics for both the training and validation sets. These plots provide insights into how well the model is learning and generalizing. For instance, if the training loss decreases while the validation loss increases, it may indicate overfitting. Conversely, if both losses decrease smoothly, it suggests good learning. Visualization helps in diagnosing issues and guiding further model tuning, ensuring that the model performs well on unseen data.

3.5.2 Daily Forecasting

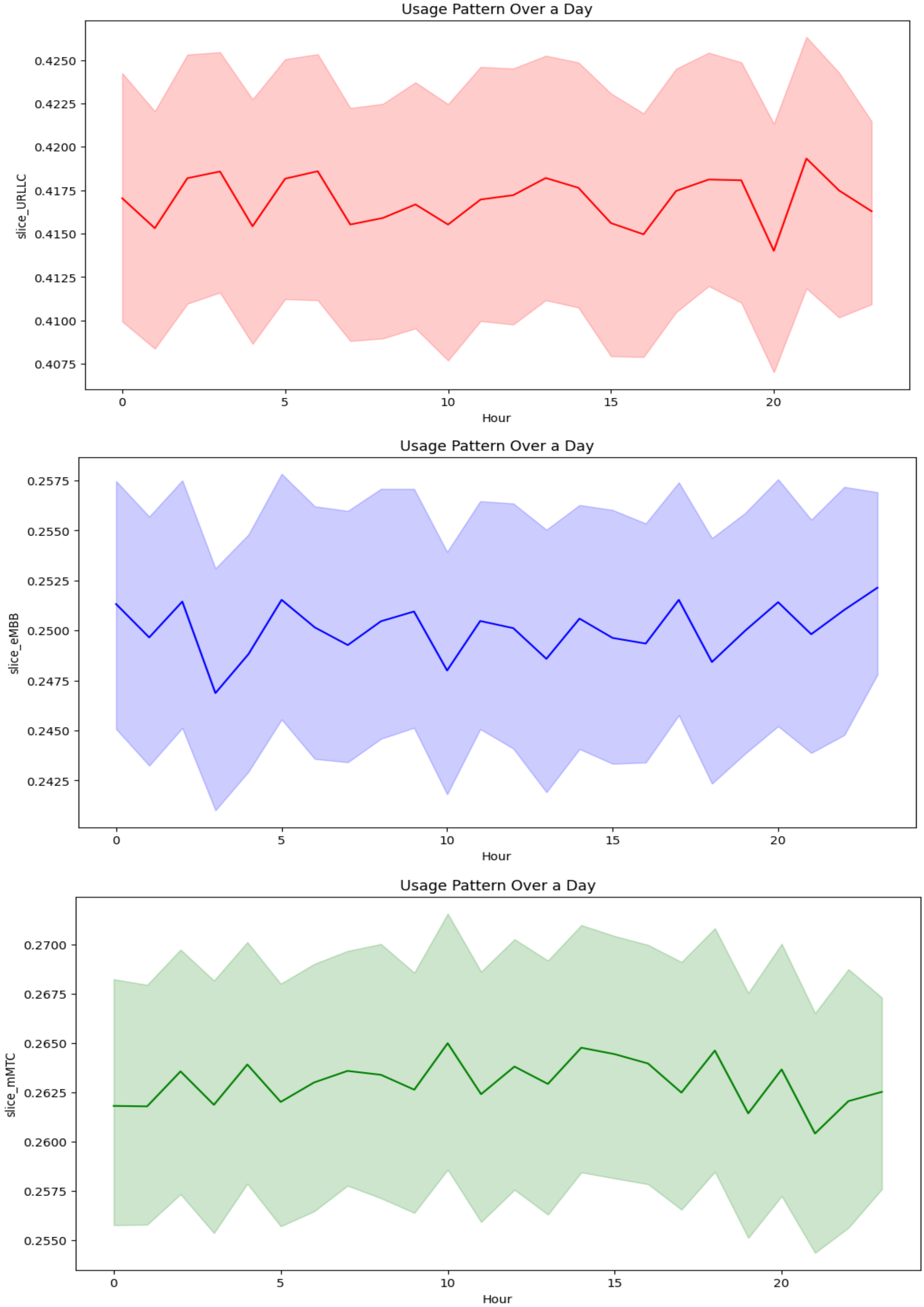


Figure 3-7: Daily Forecasting

3.5.3 Weekly Forecasting

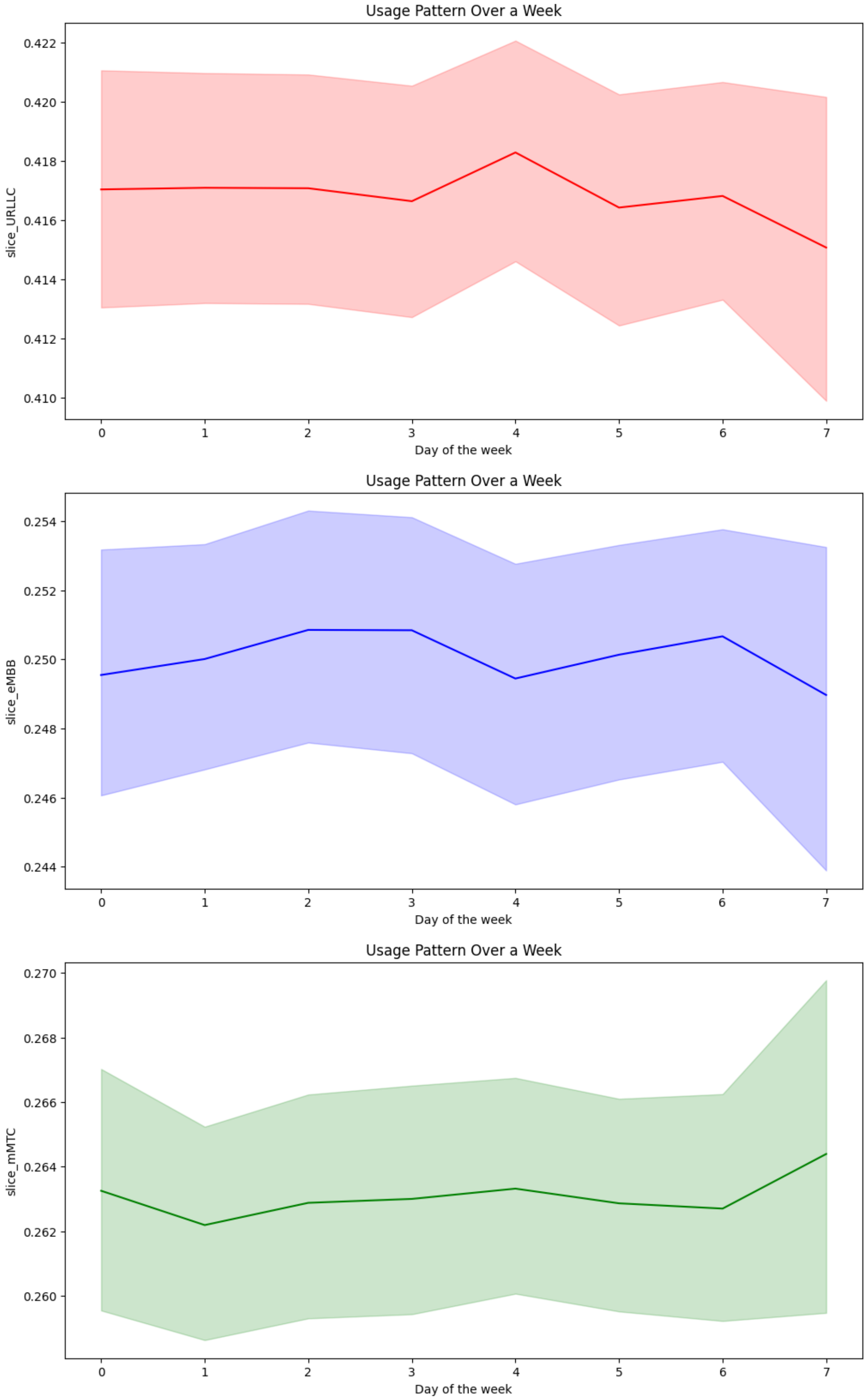


Figure 3-8: Weekly Forecasting

3.5.4 Comparison

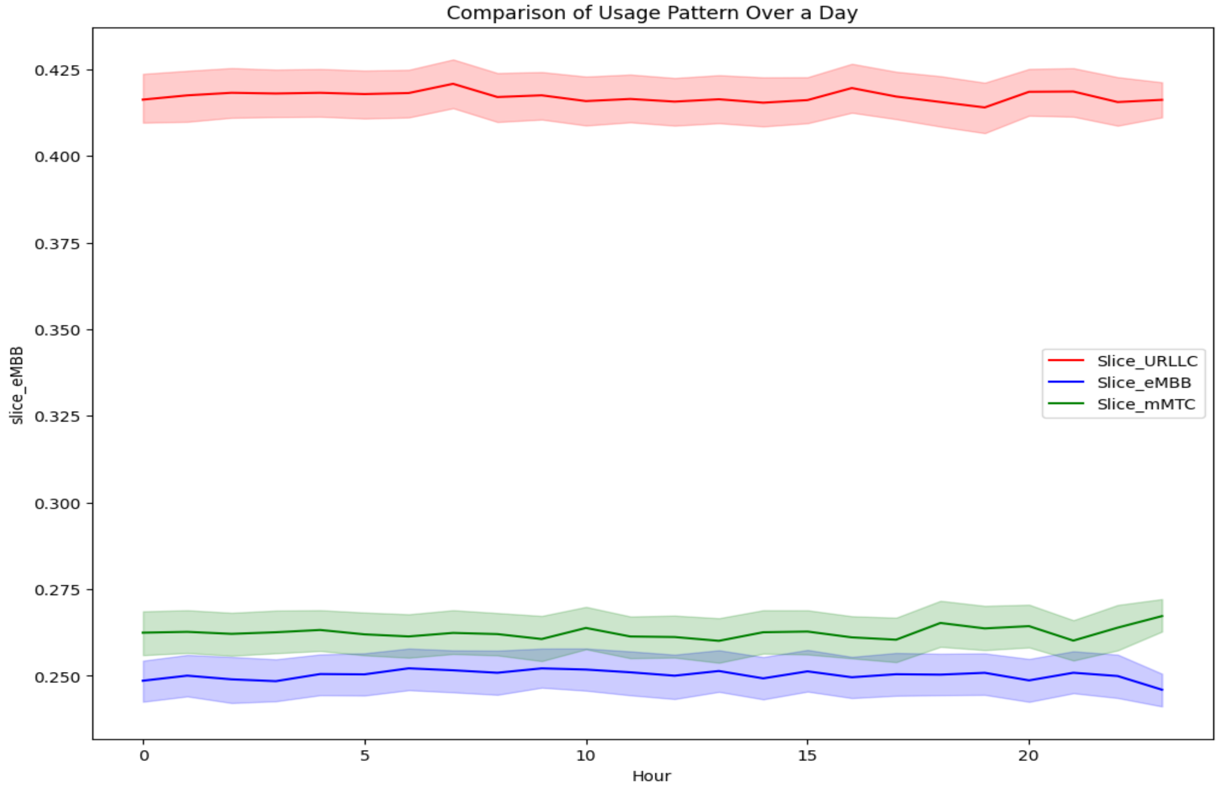


Figure 3-9: Comparison of Daily Forecasting

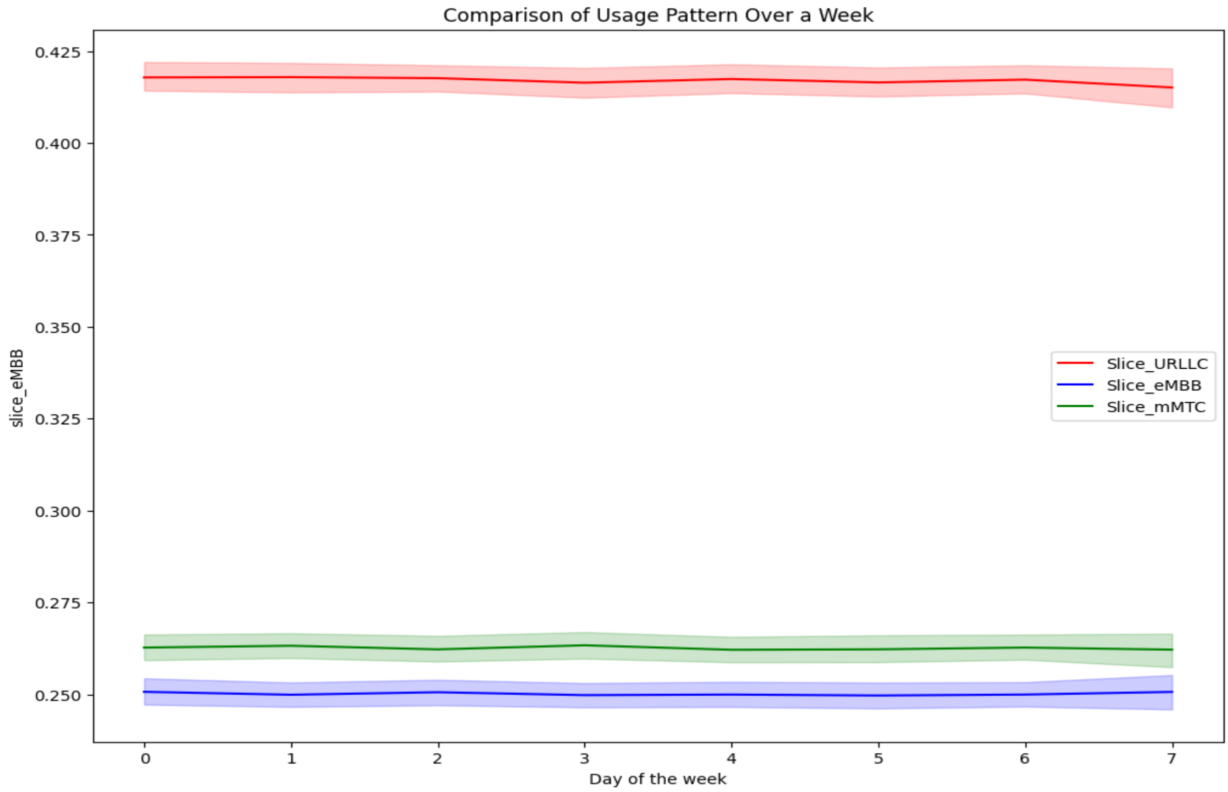


Figure 3-10: Comparison of Weekly Forecasting

3.5.5 Result Analysis

Here the x-axis represents time (hour/ week) and y-axis represents the percentage of the total resources. For instance 0.25 means 25% of the total available resources to the category.

Table 3-3: Error in Forecasting

Type	Accuracy	Error
URLLC	97%	3%
eMBB	98.22%	1.78%
mMTC	97.93%	2.07%

3.5.6 *Uses of load scheduling*

Load forecasting is essential when managing network resources to ensure optimal performance and avoid overutilization. For instance, if the number of connections in a slice exceeds a certain threshold—90% usage in our case—immediate action is required to prevent congestion. In situations of complete slice failure, such as the eMBB slice failure depicted in Fig. 3-11, DeepSlice reroutes all new eMBB-related traffic to the master slice, ensuring continuity in network transmission. However, ongoing communications on the failed slice are impacted, leading to loss of existing connections. The system logs these incidents, including date and time, to refine strategies and minimize future disruptions.

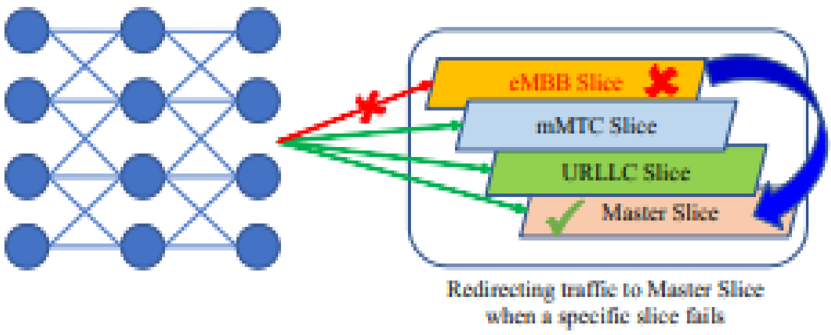


Figure 3-11: Network Failure Scenario[32]

Furthermore, our simulations, as shown in Fig. 3-12, indicate that failures on the mMTC slice from 3hr to 5hr and on the eMBB slice from 16hr to 18hr were successfully mitigated by the master slice, which redirected traffic during these periods. By reserving substantial resources in terms of capacity and processing speed within the master slice, we ensure that each network slice can rely on these backup resources during high load periods or failures, maintaining network stability and performance.[32]

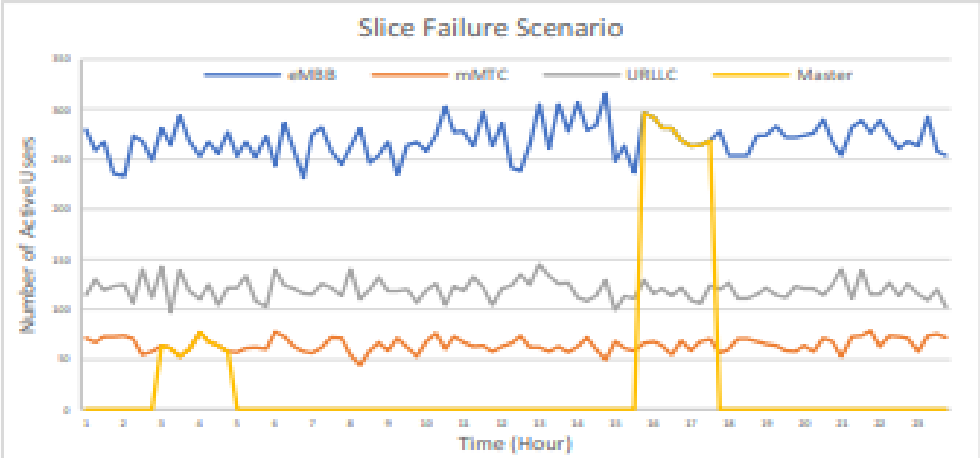


Figure 3-12: Failure Scenario Graph[32]

3.6 Achievements

The implementation of these models resulted in several notable achievements, enhancing both the efficiency and accuracy of our predictions. Below, these achievements are elaborated:

1. **Higher Prediction Model Performance:** Our CNN model demonstrated superior performance with an impressive accuracy rate of 97%. This high level of accuracy is indicative of the model's capability to effectively learn and generalize from the complex data associated with 5G network slicing. By fine-tuning the hyperparameters and utilizing a large dataset, we were able to significantly improve the prediction accuracy, ensuring reliable and precise resource allocation recommendations.
2. **More Robust Model:** To enhance the robustness of our model, data augmentation techniques were employed. This approach involved generating synthetic variations of the existing data, which helped the model generalize better by being exposed to a wider range of scenarios. As a result, the model became more resilient to overfitting and demonstrated consistent performance across diverse and unseen data. This robustness is crucial for maintaining high prediction accuracy in real-world applications where network conditions can vary significantly.
3. **Less Processing Time:** The implementation of early stopping and batch normalization significantly reduced the processing time required for training our models. Early stopping allowed us to halt the training process once the model performance ceased to

improve on the validation set, thereby avoiding unnecessary computations. Batch normalization, on the other hand, helped in stabilizing the learning process and accelerating convergence. Together, these techniques not only minimized the computational resources required but also shortened the model development cycle, enabling faster deployment of our solution.

4. **Load Scheduling:** Effective load scheduling was integral to our approach in managing computational resources efficiently. By optimizing the allocation of tasks across various processing units, we were able to ensure a balanced distribution of workload. This optimization led to improved system performance and reduced latency in the prediction and allocation processes. Consequently, our load scheduling mechanism played a critical role in enhancing the overall efficiency and scalability of our 5G resource allocation framework.

3.7 Limitations

Every research and project has some limitations. Ours is not an exception from that as well. Here are the main limitations of our project that we have faced so far:

1. **Restricted or nonexistent access to real-time data:** The effective prediction of network slicing mostly depends on whether the algorithm is trained with real time data. Since we have no collaboration with industrial entity, we have to train our model with static data. However, we are hopeful that the model will also work on real time data as well because the model that we used is effective in all cases.
2. **Limited computational power:** We have used our personal computers which does not have the same computational power and speeds as those that are set in the base station for network slicing. Therefore, the computational speed might be way too faster for that case.

Chapter 4

Demonstration of Outcome Based Education (OBE)

4.1 Introduction

The industry witnessed a transformative shift with the advent of 5G technology. "A Comprehensive Analysis on 5G Network Slicing," the thesis, explores how this innovative approach deals with virtual networks for specific 5G needs. Aligned with Outcome Based Education principles, the research offers practical insights for industry, policymakers, and academia.

4.2 Course Outcomes (COs) Addressed

Table 4-1: Course Outcomes (COs) addressed

COs	CO Statement	POs	Put Tick (√)
CO1	Identify a contemporary real life problem related to electrical and electronic engineering by reviewing and analyzing existing research works.	PO2	√
CO2	Determine functional requirements of the problem considering feasibility and efficiency through analysis and synthesis of information.	PO4	√
CO3	Select a suitable solution and determine its method considering professional ethics, codes and standards.	PO8	
CO4	Adopt modern engineering resources and tools for the solution of the problem.	PO5	√
CO5	Prepare management plan and budgetary implications for the solution of the problem.	PO11	√
CO6	Analyze the impact of the proposed solution on health, safety, culture and society.	PO6	
CO7	Analyze the impact of the proposed solution on environment and sustainability.	PO7	√
CO8	Develop a viable solution considering health, safety, cultural, societal and environmental aspects.	PO3	√
CO9	Work effectively as an individual and as a team member for the accomplishment of the solution.	PO9	√
CO10	Prepare various technical reports, design documentation, and deliver effective presentations for demonstration of the solution.	PO10	√
CO11	Recognize the need for continuing education and participation in professional societies and meetings.	PO12	√

4.3 Aspects of Program Outcomes (POs) Addressed

Table 4-2: Program Outcomes (POs) Addressed

	Statement	Different Aspects	Put Tick (√)
PO3	**Design/development of solutions**: Design solutions for complex electrical and electronic engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	Public health	√
		Safety	√
		Cultural	√
		Societal	√
		Environmental	√
PO4	**Investigation:** Conduct investigations of complex electrical and electronic engineering problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.	Design of experiments	√
		Analysis and interpretation of data	√
		Synthesis of information	√
PO6	**The engineer and society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solutions to complex electrical and electronic engineering problems.	Societal	
		Health	
		Safety	
		Legal	
		Cultural	
PO7	**Environment and sustainability:** Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex electrical and electronic engineering problems in societal and environmental contexts.	Societal	√
		Environmental	√
PO8	**Ethics:** Apply ethical principles embedded with religious values, professional ethics and responsibilities, and norms of electrical and electronic engineering practice.	Religious values	
		Professional ethics and responsibilities	
		Norms	
PO9	**Individual work and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.	Individual	√
		Teamwork	√
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	Comprehend and write effective reports	√
		Design documentation	√
		Make effective presentations	√
		Give and receive clear instructions	√

PO11	**Project management and finance:** Demonstrate knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	Engineering management principles	√
		Economic decision-making	√
		Manage projects	√
		Multidisciplinary environments	√

The following table explains or justifies how the COs and corresponding POs have been addressed in Project and Thesis.

Table 4-3: Justification of CO's & PO's

COs	POs	Explanation/Justification
CO1	PO2	Our project focuses on real-life problems because it helps managing and allocating limited network resources efficiently to meet the diverse requirements of these applications. We also reviewed and analyzed existing research works to enhance network performance by reducing prediction time and increasing the accuracy of the model.
CO2	PO4	In our project, we determined the functional requirements of the problem by investigating the existing literature on 5G architecture and resource allocation techniques. We compared various network slicing approaches to identify the most efficient and effective methods for resource allocation through analysis and synthesis of information.
CO3	PO8	N/A
CO4	PO5	Our project adopts modern engineering resources and tools for the solution of the problem by implementing ML algorithms in prediction of network slicing as well as developing and implementing sophisticated algorithms for dynamic resource allocation such as Neural Networks and LSTM.
CO5	PO11	Due to load profiling and efficient resource allocation, we can get to know about the load beforehand and manage it accordingly. Also this will allow to accommodate large number of users which will increase the profit.
CO6	PO6	N/A
CO7	PO7	Because of less energy loss and efficient use of network slices, our system consumes less energy and becomes more environment friendly. Thus, it causes less harm on the environment.
CO8	PO3	Our project efficiently manages limited network resources meet diverse application requirements. Network slicing can be very important in healthcare applications (URLLC) and safety-critical systems. Environmental considerations are also taken into account by allowing optimized resource allocation and energy efficiency.
CO9	PO9	Our project required working effectively as an individual and as a team member for the accomplishment of the solution because conducting a thorough review of existing research on network slicing and 5G technology is a critical individual task that forms the foundation of the project. Also, the team members provided continuous feedback to each other, enabling

		iterative improvements and refinements to the algorithms and the solutions being developed.
CO10	PO10	For the completion of our project, we have written comprehensive reports on methodology, experimentation and results related to network slicing in 5G. We tried to present our works with graphs, charts and diagrams to illustrate key points and data. We also worked effectively within our project group, following clear instructions for experimental setups, data collection and analysis methods.
CO11	PO12	Our project recognizes the need for continuing education and participation in professional societies and meetings because the field of 5G technology and network slicing is rapidly evolving, with frequent advancements and new research findings. Also, keeping updated with the latest developments, standards and best practices improves collaboration, innovation and practical application of research findings.

4.4 Knowledge Profiles (K3 – K8) Addressed

Table 4-4: Knowledge Profiles (K3 – K8) Addressed

K	Knowledge Profile (Attribute)	Put Tick (√)
K3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline	√
K4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline	√
K5	Knowledge that supports engineering design in a practice area	√
K6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline	√
K7	Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability	
K8	Engagement with selected knowledge in the research literature of the discipline	√

The following table explains or justifies how the Knowledge Profiles (K3 – K8) have been addressed in Project and Thesis.

Table 4-5: Justification of K's

K	Explanation/Justification
K3	Here we tried to utilize the theoretical knowledge of 5G network slicing concepts including its main three classifications and their corresponding requirements. Based on that we created a model that will predict and give the forecasting of slices. In this way we utilized the concept of communication.
K4	We integrated specialist knowledge on 5G network slicing, machine learning predictions, and CNN-based time series forecasting. We also addressed the requirement conflicts in eMBB, URLLC and mMTC, providing theoretical frameworks for each. This ensures our research is grounded in advanced, cutting-edge practices within the engineering discipline.
K5	By detailing network slicing using machine learning predictions and time series forecasting, we offer practical frameworks for addressing requirement conflicts. This ensures our research directly aids in the design and optimization of 5G networks.
K6	We demonstrated knowledge of engineering practice by exploring key technologies such as 5G network slicing and wireless communication. We addressed practical challenges such as trade-off between requirement conflicts in eMBB, URLLC and mMTC, showcasing how these technologies are applied within the communication system.
K7	N/A
K8	We engaged with selected knowledge from the research literature by incorporating findings on 5G network slicing, machine learning predictions and CNN based time series forecasting. We reviewed and synthesized relevant studies to address requirement conflicts in eMBB, URLLC, and mMTC, ensuring our research is informed by and contributes to current academic and industry discussions.

4.5 Use of Complex Engineering Problems

1. **Demand for Flexibility and Adaptiveness (P1, P2, A1):** The dynamic landscape of 5G resource allocation necessitates a profound understanding of engineering principles, reaching the level of a fundamentals-based, first-principles analytical approach (P1). This depth of knowledge is essential for addressing the conflicting technical issues and navigating the abstract complexities inherent in ensuring adaptability and flexibility (P2). To successfully navigate this terrain, diverse resources, encompassing people, money, equipment, materials, information, and technologies, must be effectively employed (A1).

2. **Dynamic Connectivity Challenges (P2, A2):** As 5G allows devices to connect or disconnect dynamically, addressing the associated challenges demands a strategic approach to conflicting technical issues and abstract problem-solving (P2). The resolution of problems stemming from interactions between stakeholders with varying needs is imperative (A2).
3. **Sophistication in Service and Demand Management (P2, P3, A2, A3):** The sophisticated nature of 5G services and demand management requires tackling conflicting technical issues, employing abstract thinking, and resolving problems arising from diverse stakeholder interactions (P2, P3, A2). Creative utilization of engineering principles and extending beyond conventional experiences are integral for effective management (A3).
5. **Prediction and Forecasting for Peak Demands (P2, P3, A2):** Addressing the challenges of managing peak demands in 5G involves navigating conflicting technical issues, utilizing abstract thinking, and resolving significant problems arising from interactions. These challenges are high-level, demanding strategic solutions.
6. **Real-time Resource Adaptation (P1, P2, P3, A2):** The imperative for real-time adaptation in 5G resource allocation underscores the necessity for in-depth engineering knowledge, addressing conflicting technical issues, utilizing abstract thinking, and resolving problems arising from diverse stakeholder interactions. These challenges are high-level and demand a holistic approach.
7. **Security Challenges in Dynamic Connectivity (P2, P3, A2):** Ensuring security in dynamic connectivity requires a strategic approach to conflicting technical issues, abstract thinking, and resolution of significant problems arising from interactions. Security challenges, though infrequent, demand a comprehensive response.
8. **Adaptive Network Load Balancing (P1, P3, A4):** The dynamic balancing of loads in 5G networks demands profound engineering knowledge, addressing conflicting technical issues, and utilizing abstract thinking (P1, P3). These challenges are high-level, necessitating the resolution of significant problems from interactions, with potential consequences for society and the environment (A4).

4.6 Socio-Cultural, Environmental, And Ethical Impact

1. **Energy Efficiency in 5G Networks (PO7):** Optimizing resource allocation in 5G networks contributes to energy efficiency, aligning with the assessment of societal and environmental contexts as per PO7 (Environment and Sustainability).
2. **Considerations for E-Waste Management (PO7):** The implementation of network slicing in 5G raises considerations for e-waste management, reflecting on the potential environmental impact. This aligns with PO7, evaluating the sustainability and impact of professional engineering work in societal and environmental contexts.
3. **Privacy and Security Measures (PO8):** Implementing robust privacy and security measures in 5G resource allocation aligns with ethical principles embedded in professional ethics and responsibilities (PO8). Engineers must uphold these principles to ensure the protection of user data and network integrity.
4. **Public Awareness and Education Initiatives (PO10):** With the deployment of 5G, effective communication becomes crucial. Engaging in public awareness campaigns and educational initiatives aligns with PO10, ensuring clear communication with the engineering community and society at large.

4.7 Attributes of Ranges of Complex Engineering Problem Solving (P1 – P7) Addressed

Table 4-6: Attributes of Ranges of Complex Engineering Problem Solving (P1 – P7) Addressed

P	**Range of Complex Engineering Problem Solving**	**Put Tick (√)**
Attribute	Complex Engineering Problems have characteristic P1 and some or all of P2 to P7:	
Depth of knowledge required	**P1:** Cannot be resolved without in-depth engineering knowledge at the level of one or more of K3, K4, K5, K6 or K8 which allows a fundamentals-based, first principles analytical approach	√
Range of conflicting requirements	**P2:** Involve wide-ranging or conflicting technical, engineering and other issues	√
Depth of analysis required	**P3:** Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models	√
Familiarity of issues	**P4:** Involve infrequently encountered issues	√
Extent of applicable codes	**P5:** Are outside problems encompassed by standards and codes of practice for professional engineering	
Extent of stakeholder	**P6:** Involve diverse groups of stakeholders with widely varying needs	

involvement and conflicting requirements		
Interdependence	**P7:** Are high level problems including many component parts or sub-problems	√

The following table explains or justifies how the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) have been addressed in Project and Thesis.

Table 4-7: Justification of P's

P	Explanation/Justification
P1	Our work requires a clear and in-depth understanding of wireless communication and 5G network architecture. Without this foundational knowledge, solving the problem solely with expertise in ML or CNN is insufficient.
P2	eMBB requires high data rates while URLLC focuses on low latency and high reliability. This leads to conflicting requirements which is considered when the pipeline or model was created.
P3	Since the requirements of each category (eMBB, mMTC, URLLC) are contradictory and conflicting, there are no obvious solutions for predicting slices. Several researchers proposed several solutions by compromising data rate, latency or reliability because it is not possible to maintain everything at the same time.
P4	The variable demand of network resources along with their conflicting requirements causes infrequent change of demands. In order to counter that infrequent changes adaptive algorithms such as ML, CNN are needed. Also we tried to maximize the efficiency by applying various algorithms.
P5	N/A
P6	N/A
P7	In our work there are mainly three components. First identifying the requirements of all type of network slices which is done using the principals and basics of wireless communication. Secondly making predictions and thirdly forecasting the demand. These two steps are done using Ml and CNN respectively. So there interconnection between each steps in order to articulate the work.

4.8 Attributes of Ranges of Complex Engineering Activities (A1 – A5) Addressed

Table 4-8: Attributes of Ranges of Complex Engineering Activities (A1 – A5) Addressed

A	Range of Complex Engineering Activities	Put Tick (√)
Attribute	Complex activities means (engineering) activities or projects that have some or all of the following characteristics:	
Range of resources	**A1:** Involve the use of diverse resources (and for this purpose resources include people, money, equipment, materials, information and technologies)	√
Level of interaction	**A2:** Require resolution of significant problems arising from interactions between wide-ranging or conflicting technical, engineering or other issues	√
Innovation	**A3:** Involve creative use of engineering principles and research-based knowledge in novel ways	√
Consequences for society and the environment	**A4:** Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation	
Familiarity	**A5:** Can extend beyond previous experiences by applying principles-based approaches	√

The following table explains or justifies how the attributes of ranges of Complex Engineering Activities (A1 – A5) have been addressed in EEE 4700/4800 (Project and Thesis).

Table 4-9: Justification of A's

A	Explanation/Justification
A1	In order to solve the problem, we took help from various resources and concepts. We tried to solve an issue of wireless communication using ML and CNN architectures. Also we collected data for our work from IEEE data port.
A2	The conflicting requirement of eMBB, URLLC, mMTC leads us to a solution where we can only satisfy either one of the conditions. So there is a trade-off while considering the solutions.
A3	We used engineering principles and research-based knowledge to explore novel approaches to network slicing in 5G. By combining traditional concepts like OFDM and Massive MIMO with innovative methodologies for resource allocation, we offer new perspectives on optimizing network performance.
A4	N/A
A5	Various researchers have tried to solve it with various algorithms of Machine learning, Neural networks etc. Since the field of algorithms and models are continuously emerging. The solution of this experiment can also be extended further.

References

[1] "From 1G to 5G: The Evolution of Mobile Communications - Mpirical." [Online]. Available: https://www.mpirical.com/blog/the-evolution-of-mobile-communication. [Accessed: 19-Jun-2024].

[2] "Evolution of wireless technologies 1G to 5G in mobile communication - RF Page." [Online]. Available: https://www.rfpage.com/evolution-of-wireless-technologies-1g-to-5g-in-mobile-communication/. [Accessed: 19-Jun-2024].

[3] V. D. Nguyen, T. Q. Duong, and Q. T. Vien, "Editorial: Emerging Techniques and Applications for 5G Networks and Beyond," *Mob. Networks Appl.*, vol. 25, no. 5, pp. 1984–1986, 2020.

[4] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[5] Z. Xu and A. Petropulu, "A Bandwidth Efficient Dual-Function Radar Communication System Based on a MIMO Radar Using OFDM Waveforms," *IEEE Trans. Signal Process.*, vol. 71, pp. 401–416, 2023.

[6] V. S. Kataksham and P. Siddaiah, "A Low Complexity PAPR Reduction of FMBC-OQAM Systems with Hybrid SLM Technique," in *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2023, pp. 1148–1153.

[7] S. Nagul, "A review on 5G modulation schemes and their comparisons for future wireless communications," *2018 Conf. Signal Process. Commun. Eng. Syst. SPACES 2018*, vol. 2018-Janua, pp. 72–76, 2018.

[8] J. Wen, R. Zhou, J. Hua, B. Sheng, and A. Wang, "Design of nonlinear phase prototype filter based on coefficients symmetry for UFMC communication system," *Digit. Signal Process.*, vol. 138, p. 104053, 2023.

[9] M. H. Mahmud, M. D. M. Hossain, A. A. Khan, S. Ahmed, M. A. Mahmud, and M. H. Islam, "Performance Analysis of OFDM, W-OFDM and F-OFDM Under Rayleigh Fading Channel for 5G Wireless Communication," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020, pp. 1172–1177.

[10] A. Kumar, "A low complex PTS-SLM-Companding technique for PAPR reduction in

5G NOMA waveform," *Multimed. Tools Appl.*, vol. 83, no. 15, pp. 45141–45162, 2024.

[11] B. Makki, K. Chitti, A. Behravan, and M. S. Alouini, "A survey of noma: Current status and open research challenges," *IEEE Open J. Commun. Soc.*, vol. 1, no. February, pp. 179–189, 2020.

[12] W. Hong *et al.*, "The Role of Millimeter-Wave Technologies in 5G/6G Wireless Communications," *IEEE J. Microwaves*, vol. 1, no. 1, pp. 101–122, 2021.

[13] F. A. Pereira De Figueiredo, "An Overview of Massive MIMO for 5G and 6G," *IEEE Lat. Am. Trans.*, vol. 20, no. 6, pp. 931–940, 2022.

[14] S. Zhang, "An Overview of Network Slicing for 5G," *IEEE Wirel. Commun.*, vol. 26, no. 3, pp. 111–117, 2019.

[15] Wojciech Mazurczyk, Pascal Bisson, Roger Piqueras Jover, Koji Nakao, and Krzysztof Cabaj, "Challenges_and_Novel_Solutions_for_5G_Network_Security_Privacy_and_Trust," *IEEE Wirel. Commun.*, no. August, pp. 6–7, 2020.

[16] O. Elgarhy, L. Reggiani, M. M. Alam, A. Zoha, R. Ahmad, and A. Kuusik, "Energy Efficiency and Latency Optimization for IoT URLLC and mMTC Use Cases," *IEEE Access*, vol. 12, pp. 23132–23148, 2024.

[17] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 957–994, 2021.

[18] A. R. Nimodiya and S. S. Ajankar, "A Review on Internet of Things," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 113, no. 1, pp. 135–144, 2022.

[19] F. Abbas, P. Fan, and Z. Khan, "A Novel Low-Latency V2V Resource Allocation Scheme Based on Cellular V2X Communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2185–2197, 2019.

[20] J. Clancy *et al.*, "Wireless Access for V2X Communications: Research, Challenges and Opportunities," *IEEE Commun. Surv. Tutorials*, p. 1, 2024.

[21] P. K. Verma *et al.*, "Machine-to-Machine (M2M) communications: A survey," *J. Netw. Comput. Appl.*, vol. 66, pp. 83–105, 2016.

[22] P. K. Baheti and A. Khunteta, "Priority-Based Resource Scheduling for Smart City

M2M Communication in 5G Networks," in *2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, 2023, pp. 1–6.

[23] B. Krogfoss, J. Duran, P. Perez, and J. Bouwen, "Quantifying the Value of 5G and Edge Cloud on QoE for AR/VR," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–4.

[24] Y. Zhuang, T. Qu, J. Wong, W. Wan, M. tom Dieck, and T. Jung, "Using 5G Mobile to Enable the Growing Slate of VR and AR Applications," in *Augmented Reality and Virtual Reality: Changing Realities in a Dynamic World*, T. Jung, M. C. tom Dieck, and P. A. Rauschnabel, Eds. Cham: Springer International Publishing, 2020, pp. 185–194.

[25] 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 - Conf. Proc.*, no. October 2019, pp. 20–25, 2018.

[26] A. B. Pawar, M. A. Jawale, P. William, and B. S. Sonawane, "Efficacy of TCP/IP Over ATM Architecture Using Network Slicing in 5G Environment," in *Smart Data Intelligence*, 2022, pp. 79–93.

[27] A. Y. Ding and M. Janssen, "5G Applications: Requirements, Challenges, and Outlook," 2018.

[28] K. Abbas, M. Afaq, T. A. Khan, A. Mehmood, and W. C. Song, "IBNSlicing: Intent-based network slicing framework for 5G networks using deep learning," *APNOMS 2020 - 2020 21st Asia-Pacific Netw. Oper. Manag. Symp. Towar. Serv. Netw. Intell. Humanit.*, no. March 2021, pp. 19–24, 2020.

[29] H. Chergui and C. Verikoukis, "Big data for 5G intelligent network slicing management," *IEEE Netw.*, vol. 34, no. 4, pp. 56–61, 2020.

[30] M. Malkoc and H. A. Kholidy, "5G Network Slicing: Analysis of Multiple Machine Learning Classifiers including the logistic regression model, linear discriminant model, k-nearest neighbor's model, decision tree model, random forest model, SVC BernoulliNB model, and GaussianNB model, to."

[31] V. P. Kafle, Y. Fukushima, P. Martinez-julia, and T. Miyazawa, "Consideration On Automation of 5G Network Slicing with Machine Learning CONSIDERATION ON AUTOMATION OF 5G NETWORK SLICING WITH MACHINE LEARNING," no.

November, 2018.

[32] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, “DeepSlice : A Deep Learning Approach towards an Efficient and Reliable Network Slicing in 5G Networks,” pp. 762–767, 2019.

[33] M. Alsenwi, N. H. Tran, S. Member, M. Bennis, and S. R. Pandey, “Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond : A Deep Reinforcement Learning Based Approach,” 2021.

[34] M. Haider *et al.*, “Computer Standards & Interfaces Optimal 5G network slicing using machine learning and deep learning concepts,” *Comput. Stand. Interfaces*, vol. 76, no. May 2020, p. 103518, 2021.

[35] M. Vincenzi, E. Lopez-Aguilera, and E. Garcia-Villegas, “Timely Admission Control for Network Slicing in 5G with Machine Learning,” *IEEE Access*, vol. 9, pp. 127595–127610, 2021.

[36] A. Domeke and B. Cimoli, “applied sciences Integration of Network Slicing and Machine Learning into Edge Networks for Low-Latency Services in 5G and beyond Systems,” 2022.

[37] K. Suh, S. Kim, Y. Ahn, S. Kim, and G. S. Member, “Deep Reinforcement Learning-Based Network Slicing for Beyond 5G,” *IEEE Access*, vol. 10, pp. 7384–7395, 2022.

[38] 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,” *J. Netw. Syst. Manag.*, vol. 30, no. 2, pp. 1–22, 2022.

[39] A. Vijayalakshmi, E. Abishek B, A. G, S. N, M. Absar M, and A. S. C, “5G Network Slicing Algorithm Development using Bagging based-Gaussian Naive Bayes,” in *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 2023, pp. 1–5.

[40] S. Venkatapathy, T. Srinivasan, H. G. Jo, and I. H. Ra, “Optimal Resource Allocation for 5G Network Slice Requests Based on Combined PROMETHEE-II and SLE Strategy,” *Sensors*, vol. 23, no. 3, pp. 1–19, 2023.

[41] “User Guide — pandas 2.2.2 documentation.” [Online]. Available: https://pandas.pydata.org/docs/user_guide/index.html#user-guide. [Accessed: 20-Jun-2024].

[42] "NumPy user guide — NumPy v2.0 Manual." [Online]. Available: https://numpy.org/doc/stable/user/index.html#user. [Accessed: 20-Jun-2024].

[43] "User guide and tutorial — seaborn 0.13.2 documentation." [Online]. Available: https://seaborn.pydata.org/tutorial.html. [Accessed: 20-Jun-2024].

[44] "Using Matplotlib — Matplotlib 3.9.0 documentation." [Online]. Available: https://matplotlib.org/stable/users/index.html. [Accessed: 20-Jun-2024].

[45] "User Guide — scikit-learn 1.5.0 documentation." [Online]. Available: https://scikit-learn.org/stable/user_guide.html. [Accessed: 20-Jun-2024].

[46] "XGBoost Documentation — xgboost 2.0.3 documentation." [Online]. Available: https://xgboost.readthedocs.io/en/stable/. [Accessed: 20-Jun-2024].

[47] "Module: tf | TensorFlow v2.16.1." [Online]. Available: https://www.tensorflow.org/api_docs/python/tf. [Accessed: 20-Jun-2024].

[48] L. Taylor and G. Nitschke, "Improving Deep Learning with Generic Data Augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1542–1547.

[49] Q. Yang, P. Li, X. Xu, Z. Ding, W. Zhou, and Y. Nian, "A Comparative Study on Enhancing Prediction in Social Network Advertisement through Data Augmentation," pp. 9–11, 2024.

[50] A. O. Victor and M. I. Ali, "Enhancing Time Series Data Predictions: A Survey of Augmentation Techniques and Model Performances," in *Proceedings of the 2024 Australasian Computer Science Week*, 2024, pp. 1–13.

[51] C. V. Gonzalez Zelaya, "Towards Explaining the Effects of Data Preprocessing on Machine Learning," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019, pp. 2086–2090.

[52] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021.

[53] O. Adamuz-Hinojosa, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, "Computationally Efficient UE Blocking Probability Model for GBR Services in Beyond 5G RAN," *IEEE Access*, vol. 12, pp. 39270–39284, 2024.

[54] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A Review of Activation Function for Artificial Neural Network," in *2020 IEEE 18th World Symposium on Applied Machine*

Intelligence and Informatics (SAMI), 2020, pp. 281–286.

[55] Z. Zhang, “Improved Adam Optimizer for Deep Neural Networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2.

[56] L. Prechelt, “Early Stopping - But When?,” in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 55–69.

[57] P. M. Radiuk, “Impact of training set batch size on the performance of convolutional neural networks for diverse datasets,” *Inf. Technol. Manag. Sci.*, vol. 20, no. 1, pp. 20–24, 2017.

[58] S. M. Al-Selwi *et al.*, “RNN-LSTM: From applications to modeling techniques and beyond—Systematic review,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 5, p. 102068, 2024.

[59] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[60] H. Zhang, H. Sun, L. Kang, Y. Zhang, L. Wang, and K. Wang, “Prediction of Health Level of Multiform Lithium Sulfur Batteries Based on Incremental Capacity Analysis and an Improved LSTM,” *Prot. Control Mod. Power Syst.*, vol. 9, no. 2, pp. 21–31, 2024.

[61] U. B. Mahadevaswamy and P. Swathi, “Sentiment Analysis using Bidirectional LSTM Network,” *Procedia Comput. Sci.*, vol. 218, pp. 45–56, 2023.

[62] M. Ehteram, M. Afshari Nia, F. Panahi, and A. Farrokhi, “Read-First LSTM model: A new variant of long short term memory neural network for predicting solar radiation data,” *Energy Convers. Manag.*, vol. 305, p. 118267, 2024.