Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

An Improved 5G Slicing Classification Based on Machine Learning Approaches

A Dissertation
Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology/ Software

By:

Hawraa Shareef Hamza Hussein

Supervised by:

Asst. Prof. Dr. Mehdi Ebady Manaa

2023 A.C. **1445 A.H.**

بسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ

﴿ وَلَوْ أَنَّمَا فِي الْأَرْضِ مِن شَجَرَةٍ أَقْلَامٌ وَالْبَحْرُ يَمُدُّهُ مِن بَعْدِهِ سَبْعَةُ أَبْحُرٍ مَّا نَفِدَتْ كَلِمَاتُ اللَّهِ ۗ إِنَّ اللَّهَ عَزِيزٌ حَكِيمٌ ﴾

صدق اللّهُ العلي العظيم

سورة لقمان- الآية ٢٧

Supervisor Certification

I certify that this dissertation was prepared under my supervision at the Department of Software / College of Information Technology / Babylon University, by **Hawraa Shareef Hamza Hussein** as a partial fulfillment of the requirements for the degree of **Ph.D. in Information Technology**.

Signature:
Name: **Dr. Mehdi Ebady Manaa**
Title: **Assistant Professor**
Date: / / 2024

The Head of the Department Certification

In view of the available recommendation, we forward this dissertation for debate by the examining committee.

Signature:
Name: **Dr. Sura Zaki Alrashid**
Title: **Assistant Professor**
Date: / / 2024

Declaration

I hereby declare that this dissertation entitled **"An Improved 5G Slicing Classification Based on Machine Learning Approaches"** submitted to University of Babylon in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for reports and summaries whose sources are appropriately cited in the references.

Signature:

Name: **Hawraa Shareef Hamza**

Date: / / 2024

Dedication

To

The Messenger of Allah

Muhammad, (Allah's praise and peace upon him and his purity family)

In loving memory of my dear brother, **Hatem**.

Throughout my academic journey, you have been a constant source of love, support, and encouragement. In times of doubt, you were always there to remind me of my strengths and to push me beyond my limits. Your guidance and wisdom have shaped my thoughts and guided me in the right direction. Your presence may be absent physically, but your spirit and influence continue to resonate within me

Acknowledgment

The greatest thanks be to ALLAH Almighty, Who Has been my helper in all my moments and has given me strength and authority to complete this work. I also thank His Prophet and beloved Muhammad and his pure family, especially Imam Al-Kadhim, the owner of generosity.

I would like to express my sincere gratitude and appreciation to **Dr. Mehdi Ebady Manaa** for his invaluable guidance and mentorship throughout this journey. His expertise, patience, and constant encouragement were helpful in shaping this work and pushing me to success

I extend my heartfelt appreciation to my friends and colleagues who provided me with their support, feedback, and valuable insights. Their constructive criticism and collaborative spirit have significantly enriched this work.

Finally, I would like to express my sincere gratitude and appreciation to all those who have contributed to the completion of this work. Without their support, guidance, and assistance, this achievement would not have been possible.

Abstract

The rapid spread of Fifth Generation (5G) technology has piloted in an era of extraordinary connectivity, accommodating a vast array of users and transmitting immense data volumes. Among this growth, the concept of network slicing within the 5G has emerged. Network slicing enables the creation of isolated logical networks tailored to specific service requirements. However, this advancement has own challenges; the limited availability of appropriate datasets for constructing a model capable of classifying network slices that fulfill user requirements. Furthermore, there is a necessity to choose network slices in a manner that guarantees optimal network performance, encompassing low latency and high reliability for each slice.

This dissertation proposes an approach to deal with the 5G network slicing classification. There are two proposed cases. Case 1 include two models: Model 1, a hybrid model combining decision tree, random forest, TabNet, and a hybrid PSO-ANN classifier, is designed for classifying network slicing by handling unbalancing data sources. The main contribution of this model is using a hybrid model for decreasing the training time. This model outperforms existing benchmarks on all performance metrics. Model 2, an enhanced model, combining decision tree, random forest, ANN, and an enhanced TabNet classifier, is designed for classifying network slicing by handling tabular data sources. This model gives a good results comparing to an existing benchmarks on all performance metrics. Both models offer potential reductions in latency and increased reliability for real-time 5G applications.

While Case 2 include a model that deal with the situation of data scarcity; the model based on principle of transfer learning approach by utilizing a TabNet and a decision tree classifier. TabNet, trained on a vast source domain dataset extracts features and patterns while the fine-tuning process adapts the model to the specific target domain, enhancing its ability to determine domain-specific patterns and improve classification accuracy.

The first model of case 1 achieved an accuracy (97%), compared to the benchmark model (96.6). While the second model exhibited comparable performance with (98%) compared to the benchmark model (96.6%). Furthermore, case 2 Model (Transfer Learning with TabNet) demonstrated significant accuracy gains, conduct extensive experiments using a three cases of limited dataset (100, 200 and 300 instances) representative of 5G network scenarios. The experimental results demonstrate that the transfer learning approach achieves good classification accuracy (75%,77% and 78%) respectively for three data set cases compared to situations before using transfer learning where accuracy is (70%,75% and 73%). Finally, the high accuracy the proposed models suggest a potential reduction in latency and improve reliability, critical for real-time applications and services.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
1G	First Generation
2G	Second Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5GE2ES	5G End-to-End Slicing
5QI	5G Quality-of-service Identifier
AR	Augmented Reality
B5G	Beyond Fifth Generation
BBU	Base Band Unit
BeC3	Behavior Crowd Centric Composition
BN	Batch Normalization
BS	Base Stations
CAPEX	Capital Expenditure
CM	Confusion Matrix
CNN	Convolution Neural Network
C-RAN	Cloud Radio Access Network
CSI	Channel State Information
CU	Control Unit
DL	Deep Learning
DNN	Deep Neural Networks
DRL	Deep Reinforcement Learning
DTL	Deep Transfer Learning
DU	Data Unit
ELU	Exponential Linear Unit
eMBB	Enhanced Mobile Broadband
ERM	Episodic Relaxation Method
FC	Fully Connected
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
GLU	Gated Linear Unit
IoT	Internet of Things
ITU	International Telecommunication Union
KPIs	Key Performance Indicators
KQIs	Key Quality Indicators

MADRL	Multi-Agent Deep Reinforcement Learning
MDP	Markov Decision Process
MIMO	Multiple Input and Multiple Output
MINLP	Mixed Integer Nonlinear Programming
MIP	Mixed Integer Programming
ML	Machine Learning
mMTC	Massive Machine Type Communications
NFV	Network Functions Virtualization
NN	Neural Network
OAI	Open Air Interface
OPEX	Operational Expenditure
PSO	Particle Swarm Optimization
QoS	Quality of Service
RAN	Radio Access Network
ReLU	Like Rectified Linear Units
RF	Random Forest
RRH	Radio Remote Head
RRU	Remote Radio Unit
SDN	Software Defined Networking
SDR	Software-Defined Radio
TL	Transfer Learning
TN	True Negative
TP	True Positive
uRLLC	Ultra-Reliable and Low Latency Communications
VBSs	Virtual Base Stations
VR	Virtual Reality

List of Dissertation Related Publications

(First Paper)

- Name of Journal: IEEE Xplore.
- Scopus.
- Name of Conference: 1st. Babylon International Conference on Information Technology and Science 2021 (BICITS 2021)- Babil- IRAQ.
- Paper Title: The Importance of Network Slicing and Optimization in Virtualized Cloud Radio Access Network.
- State of paper: published.
- Authors:

➢ Hawraa S. Hamza
➢ Asst. Prof. Dr. Mehdi E. Manaa.

Software Department, Information Technology College, Babylon University. Information Networks Department, Information Technology College, Babylon University.

Email: hawraa@itnet.uobabylon.edu.iq

(Second Paper)

- Name of Journal: Journal of Physics: Conference Series.
- Scopus. Cite score 1.0.
- Name of Conference: 12th Int'l Conference on Mathematical Modeling in the Physical Sciences, August 28-31, 2023, Belgrade, Serbia.
- Paper Title: A developed Optimal Network Slicing Model for 5G Network using Multi-Layer Deep Learning.
- State of paper: Accepted.
- Authors:

➢ Hawraa S. Hamza
➢ Asst. Prof. Dr. Mehdi E. Manaa.

Software Department, Information Technology College, Babylon University. Department of Intelligent Medical Systems, College of Health and Medical Techniques, Al-Mustaqbal University, Babylon, Iraq; University of Babylon.

Email: hawraa.shareef@uobabylon.edu.iq.

Chapter One

General Introduction

Chapter One

General Introduction

1.1 Introduction

The utilization of telecommunication networks has evolved considerably in the last two decades, including the number of users connected and the volume of data transmitted. The service-oriented design of the Fifth Generation (5G) architecture enables a multi-service network to handle a variety of communication scenarios with a wide range of performance and service needs, and to supply all types of services to all types of user requirements [1, 2].

The deployment of new technologies is significantly increasing network costs, including **Cap**ital **Ex**penditure (CAPEX) and **Op**erational **Ex**penditure (OPEX) because of the rapid growth in the number of connected terminals and mobile devices (smartphones, tablets, IoT devices, and so on). So, to meet the various service requirements, the **C**loud **R**adio **A**ccess **N**etwork (C-RAN) architecture has been suggested as a possible 5G design. The primary idea behind C-RAN is to separate **B**ase **B**and **U**nit (BBU) from antennas and combine processing power into centralized data centers, or BBU pools. To increase resource usage and reduce network costs, several **B**ase **S**tations (BS) will share the BBU computation pools [3].

An essential component of the 5G network is network slicing, which enables the creation of numerous isolated logical networks over the same physical infrastructure. A slice is created with the goal of enabling various use cases with various service needs. Due to the vast array of new networking services, 5G mobile networks have diverse service needs. The network slicing model has therefore evolved since the "one size fits all" networking notion is not appropriate for 5G and beyond. In order for each logical network to be tailored to provide certain network capabilities and features for a particular use case, the physical network is divided into a number of logical networks (referred to as "network slices") [1, 4, 5].

To be more precise, a network slice is a tenant's own private, isolated subnetwork, complete with set of topology, virtual resources, provisioning rules , and traffic flows. Logical networks are constructed and deployed to various services to satisfy the diverse communication requirements of users by monitoring the demands of users and administrators. Because network slicing enables dynamic, flexible, and scalable networks to adapt quickly to changing business requirements, network slice is often recognized as a key enabler of 5G systems [6].

Smart decisions must be made regarding the construction, design, operation, deployment, management, and administration of a network slice; In order to effectively meet the **Q**uality of **S**ervice (QoS) requirements of the service intended to be delivered through a network slice. Analyzing a massive amount of data in a short amount of time is a challeng for a human to manually design and run network slices. Therefore, automation of these tasks is required. Network slicing operations can be automated thanks to **M**achine **L**earning (ML) and **D**eep **L**earning (DL) [5].

In addition to these, ML will be required for 5G networks and future use cases. In general, ML algorithms can process massive amounts of data, spot outliers, anticipate outcomes, and adjust rapidly to new circumstances. ML is able to enhance and automate network management thanks to the presence of such features. Actually, ML can be helpful in a variety of scenarios, including QoS prediction, power savings, operation, fault management, maintenance, network setup, power control, coverage, and throughput [7].

DL is a subset of ML and consists of three learning methods: unsupervised, semi-supervised, and supervised. DL has received more attention from academics and companies in recent years than standard techniques to ML. DL has been the most widely used computational approach achieving remarkable outcomes on a wide range of complex cognitive tasks, matching, or even outperforming human performance. One of the advantages of deep learning is the ability to learn from massive amounts of data. DL has surpassed well-known ML algorithms in a variety of domains, including networking cybersecurity, natural language processing, bioinformatics, robotics and control, and medical information processing [8].

In networking for instance, DL can cause network automation to update the resources that are accessible and make changes immediately. In addition to processing and supplying information, DL will be tasked with adapting network resource use without human involvement. The greatest judgments will be made by combining a range of criteria, maybe too numerous for a person to evaluate all at once or to be able to process quickly. For every particular slice, DL will do real-time analysis to assess network performance, establish a prospective performance baseline, be proactive in predicting issues, examine various network components, and look for any anomalies [9].

Accurate service classification in 5G networks enables intelligent resource allocation, quality-of-service optimization, and network traffic management. However, building effective classification models relies heavily on the availability of labeled data for training. In practice, generating a large-scale labeled dataset is often challenging. This scarcity of labeled data poses a significant hurdle to achieving accurate service classification in 5G networks [10].

Also, a tabular data makes another challenge because of many issues, including lack of locality, mixed feature types (numeric, ordinal, categorical), data sparsity, and a lack of previous understanding of the dataset structure (unlike with text or images) [11]. Moreover, unbalancing data can be more challenging to deal with, which means that some classes have a significantly higher number of examples in the training set than other classes [12].

Service classification facing challenges due to a scarcity of data from the networks that is used by a small number of users. Therefore, to address above challenges, **T**ransfer **L**earning (TL), combined with DL models, has emerged as a promising approach. Transfer learning leverages pre-trained models that are trained on large-scale datasets from related domains and adapts them to the target classification task with limited labeled data. By exploiting the knowledge learned from these pre-trained models, transfer learning enables effective classification even when faced with scarce datasets [13, 14].

1.2 Problem Statement

The classification of 5G optimal network slice that meet the users demands is becoming increasingly significant for effective resource utilization and network management as 5G networks evolve. Within the domain of 5G network slicing, significant challenges, arise due to the scarcity of suitable datasets for building a model capable of classifying network slices that meet user's requirements, Additionally, there's a need to select optimal network slices in a way that ensure optimal network performance, such as low latency and high reliability of each slice. Based on the above discussion, the nature of data is a tabular data and unbalanced data so the complexities of handling such data requires specialized tabular DL model. Furthermore. The lack of enough 5G network services dataset and lack of computing capabilities, which limits the capability of ML/DL classification process: requires service classification model based on TL principles. By leveraging the knowledge learned from the pre-training tabular model that can be beneficial for the target task.

1.3 Dissertation Aim

This dissertation aims to explore these challenges and develop solutions to enhance the efficiency and accuracy of 5G network slicing and classification processes. Top of Form.

1.4 Dissertation Challenges

Working in the field of 5G network slicing requires the effort and work of large companies and institutions, so there was a great challenge in obtaining information related to the topic. 5G network slicing also one of the modern topics that greatly attracts the attention of researchers. So, the most challenging aspect of this dissertation was obtaining data. All available datasets were searched, but satisfactory data could not be found. The search proved to be time-consuming.

1.5 Dissertation Contributions

The main contributions of this dissertation are:

1) Due to the lack of dataset for 5G network slicing: we generated a set of n features (such as Latency, Jitter, Bit Rate .. etc.) with m of classes (such as Ultra-High-Definition Video Streaming, Immersive Experience, ... etc.) that represent 5G network scenarios by examining parameters derived from **I**nternational **T**elecommunication **U**nion (ITU) standards papers as well as other European studies, instances for each class could be generated consistently every time. As well as making the number of instances different for each class and not convergent.
2) To address the challenges of dealing with tabular data and selecting the optimal network slice that satisfies the requirements of the user of the 5G network, two models were proposed:
   - Case 1: A DL model with two levels. The first level is an Attentive Interpretable Tabular Learning; TabNet that is interlocked with ML models. The second level fine-tunes the hyper-parameters using a proposed metaheuristic approach that hybridizes Deep Neural Networks (DNN) with Particle Swarm Optimization (PSO). The model achieved good results for selecting an optimal network slice compared to the benchmark and compared with each stand alone model.
   - Case 2: A DL model with two levels. The first level is a DNN model that is interlocked with ML models. The second level is an enhanced

TabNet model. The model also achieved good results for selecting an optimal network slice compared to the benchmark.

3) The scarcity of enough 5G network services dataset and lack of computing capabilities, which limits the capability of DL classification process: a service classification model was proposed based on TL principles. By leveraging the knowledge learned from the pre-training phase, the pre-trained models can capture useful patterns, feature representations, and relationships that can be beneficial for the target task. TabNet model was used as a pre-trained model; Where this model has not been previously used as a TL model.

1.6 Dissertation Questions

The main questions of this dissertation are:

1) What challenges arise due to the lack of a sufficient dataset for 5G network slicing, and how was a new dataset generated to address these challenges?
2) How does the proposed DL model with two levels address the task of optimal network slice selection in 5G networks, and how is the hybrid metaheuristic approach utilized for fine-tuning?
3) How did the proposed models achieve efficient results for selecting an optimal network slice compared to the benchmark?
4) How does dealing with tabular data pose unique challenges for DL, and how do specialize tabular DL models address these challenges effectively?
5) How can TL be used to improve the performance of 5G network service classification models?

1.7 Dissertation Objectives

The main objectives of this dissertation are:

1) Carrying out the process of generating dataset for the 5G Network slicing based on information gathered from Institutions and other.
2) Dealing with tabular data, which is considered as a barrier to DL, unlike other data types such as images and text; by using a special tabular DL Models.
3) Dealing with unbalanced data that can be more challenging to work with! Because unbalanced data can be biased towered majority class making less accurate in predicting the minority class.
4) Selecting an optimal network slice to satisfy user demand by using a model that based on tabular DL and ML models followed by hybrid optimization classifier.
5) Selecting an optimal network slice to satisfy user demand by using a model that based on DNN and ML models followed enhanced TabNet model.
6) Dealing with scarce data by applying the TL principle through the use of a TabNet model.

1.8 Related Works

Many research projects deal with techniques to work with 5G Network. It can be summarized as follows:

A. Constructing and developing approaches based on network slicing,
B. Constructing and developing approaches based on TL techniques.

1.8.1 Related works in 5G network based on network slicing

In 2018, Costanzo et al. [15] proposed a prototype of Flex **R**adio **A**ccess **N**etwork (RAN) with **S**oftware **D**efined **N**etworking (SDN) controller and **O**pen-Air **I**nterface (OAI) platform for network slicing in a C-RAN with the objective of sharing spectrum between different slices in an efficient way with considering their requirements. The result showed that the proposed prototype has the ability for providing isolation between multiple slices in a dynamic manner.

In 2019, Costanzo et al. [16] presented a prototype of C-RAN based on the platform of OAI and Docker container technology, with the aim of distributing the spectrum resources among multiple slices in an efficient manner. In addition, by using **I**nternet **o**f **T**hings (IoT) devices orchestrated by the **B**ehavior **C**rowd **C**entric **C**omposition (BeC3) framework and real smartphones, for validating the prototype ability in configuring in real time each slice. As a result, the prototype enabling efficient splits of spectrum resources between slices.

In 2019, Gupta et al. [17]; presented a ML based approach to slice allocation in 5G networks. The proposed approach was evaluated using a simulation study with a variety of traffic scenarios. The results showed that the proposed approach was able to allocate slices efficiently and achieve good QoS in all of the scenarios.

In 2020, Barmpounakis et al. [18] present Cross Layer Controller and SDN with **S**oftware-**D**efined **R**adio (SDR) in order to optimize the execution of RAN for 5G and providing the needed flexibility to the polices of the network slicing. The suggested system allows for real-time optimization of slicing rules and techniques between radio and network.

Preciado-Velasco et al. [19] suggested a service classifier in 2021 that employs ML to increase the accuracy of service categorization in 5G and **B**eyond **F**ifth **G**eneration (B5G) networks. The classifier employs supervised learning, which means it is trained on a collection of labeled data. The classifier may be used to categorize services based on **K**ey **P**erformance **I**ndicators (KPIs) and **K**ey **Q**uality **I**ndicators (KQIs). This enables network operators to choose the best network slice for each service, hence improving user QoS. The authors of the article ran simulations with several ML algorithms and discovered that a **R**andom **F**orest (RF) classifier had the greatest accuracy 96.6 %.

Gabilondo et al. [20] proposed two ways for dynamically categorizing the traffic categories of individual flows and sending them via a specified slice with an associated **5**G **Q**uality-of-service **I**dentifier (5QI) in 2022. The first method routes all flows via the most appropriate slice for the dominating traffic, while the second method routes each data flow individually through the slice associated with that category. The findings reveal that the second strategy outperforms the first by achieving more control and accuracy in settings with diverse traffic.

Mohammedali et al. [21] presented a **5**G **E**nd-**t**o-**E**nd **S**licing (5GE2ES) paradigm for network slicing in 5G networks in 2023. The model use ML to categorize and forecast slices across 5G networks in order to optimally distribute 5G slice resources and ensure QoS. The suggested model outperforms previous techniques in terms of forecasting 5G service in less time and with less computing resources. The model is trained using different ML methods, and the best model is picked to decrease computing power in subsequent networks. The experimental outcomes show the efficacy of the proposed 5GE2ES model for effectively distribute resources and guarantee QoS while decreasing processing power in future networks.

1.8.2 Transfer learning-based related works in 5G networks

In 2020, Zeng et al. [22] suggested an innovatory technique to wireless cellular traffic forecast. This method combines TL and cross-domain learning to increase the accuracy of 5G/B5G cellular network traffic forecast. The technique begins by training a model on traffic data from a single service or area. The model is then employed to evaluation traffic statistics from another service or location. In addition, the model is trained on data from several services, allowing it to identify the cohesions between various traffic patterns. Experimental results show that the model significantly improves the prediction accuracy of 5G/B5G cellular network traffic.

In 2020, Fan et al. [23]; proposed IoT Defender, a TL-based intrusion detection framework for 5G IoT. IoT Defender leverages the layered and distributed nature of 5G edge computing to perform data aggregation using federated learning and construct customized detection models through transfer learning. This allows IoT Defender to achieve high detection accuracy and a low false positive rate, while also protecting the privacy of the data. Experimental results show that IoT Defender outperforms traditional methods.

Yi Liu et. al. [24] suggested a safe and intelligent architecture for 5G networks in 2020, using federated learning as well as TL methodologies. The framework enhances network performance and ensures privacy preservation by employing federated learning and TL in the access selection strategies. The federated user authentication model also ensures privacy preservation and scalability during user authentication in 5G networks. The experimental findings suggest that the proposed architecture has the potential to alter 5G networks by delivering a safe, efficient, and smart alternative.

In 2021, Zeng et al. [25] proposed a **D**eep **T**ransfer **L**earning (DTL); to address the challenge of high training cost for downlink CSI feedback in 5G massive **M**ultiple **I**nput and **M**ultiple **O**utput (MIMO) systems. DTL fine-tunes a pre-trained model using a relatively small number of samples in a new wireless environment. This is accomplished by transferring information from the pre-trained model to the new model, allowing the new model to generalize more effectively to the new environment. The experimental findings show that the suggested DTL approach may achieve equivalent performance to **N**eural etwork (NN) training on bigger datasets while drastically lowering training costs.

Guan et al. [26] suggested a DTL based traffic categorization approach for 5G IoT situations with limited labeled data and compute power in 2021. First, a model is trained on a big collection of traffic data from other 5G IoT networks. This model was then fine-tuned using a small sample of target network traffic data. Experimental results show that the proposed method can achieve classification accuracies close to those obtained with the full training dataset. This makes the proposed method a promising approach for traffic classification in 5G IoT scenarios.

In 2021, Yang et al. [27] presented proposed a Transfer Leaning-enabled edge **C**onvolution **N**eural **N**etwork (CNN) framework for privacy-preserving image classification in 5G industrial edge networks. The framework uses TL to fine-tune a pre-trained CNN model on the edge server using limited datasets uploaded from the devices. This addresses the energy constraints of devices and limited communication bandwidth. Experimental results using ImageNet demonstrate that the proposed TL -enabled edge-CNN framework achieves a good prediction accuracy compared to the baseline.

Tianlun Hu et al. [28] proposed a new TL-aided **M**ulti-**A**gent **D**eep **R**einforcement **L**earning (MADRL) strategy for inter-cell inter-slice resource partitioning in 2023. To learn how to split resources across slices, the methodology employs a coordinated MADRL algorithm with information sharing. The integrated TL technique distributes information from more experienced to less experienced agents, accelerating policy deployment and improving approach performance. The authors used simulations to demonstrate that the suggested technique exceeds current technologies with respect to convergence speed, efficiency, and sample effectiveness.

The rise of 5G networks carries opportunities together with challenges in managing diverse user demands through network slicing. While previous research explored spectrum sharing, allocation, and real-time optimization for slices, a critical gap remains; accurately classifying user needs for optimal slice selection. Traditional ML/DL methods face data scarcity, calling for advanced solutions like TL. However, existing summaries of related works lack details on data used, evaluation metrics, and focus on the relevant tabular data format. Filling these gaps is crucial to bridge the gap between challenges and potential ML/DL and TL solutions for effective 5G network slicing.

Table 1.1 shows a summary of the earlier research with further information.

Table 1.1: Summary of literature reviews

Year	Authors	Objective	Approach	Key Findings
Related Works in 5G network based on Network Slicing				
2018	Costanzo et al. [15]	Efficient spectrum sharing between slices in C-RAN	Flex RAN prototype with SDN and OAI	Isolation between slices achievable
2019	Costanzo et al. [16]	Efficient spectrum allocation among slices	OAI and Docker container based C-RAN prototype with BeC3 and real devices	Efficient spectrum splits between slices
2019	Gupta et al. [17]	ML-based slice allocation	Simulation study with various traffic scenarios	Efficient slice allocation and good QoS in all scenarios
2020	Barmpounakis et al. [18]	Real-time optimization of network slicing	Cross Layer Controller and SDN with SDR	Flexibility for slice policies and real-time optimization
2021	Preciado-Velasco et al. [19]	ML-based service classifier for accurate slice selection	Supervised learning with Random Forest	96.6% accuracy in service categorization and improved QoS
2022	Gabilondo et al. [20]	Dynamic traffic categorization and routing for slices	Two methods: dominant traffic and individual flow routing	Individual flow routing outperforms dominating traffic for improved control and accuracy
2023	Mohammedali et al. [21]	ML-based slice categorization and forecasting for optimal resource allocation	5GE2ES model with various ML algorithms	Efficient resource distribution, QoS guarantee, and reduced computing power
Transfer Learning-based Related Works in 5G Networks				
2020	Zeng et al. [22]	5G/B5G Network Traffic Forecasting	Improve prediction accuracy	Significantly improves prediction accuracy

Year	Authors	Objective	Approach	Key Findings
2020	Fan et al. [23]	5G IoT Intrusion Detection	High detection accuracy & low false positives	Outperforms traditional methods
2020	Yi Liu et al. [24]	5G Network Architecture	Enhance performance & privacy	Promising approach for secure, efficient, and smart networks
2021	Zeng et al. [25]	Downlink CSI Feedback in 5G MIMO	Reduce training cost	Equivalent performance to NN training with less data
2021	Guan et al. [26]	5G IoT Traffic Categorization	Accurate classification with limited data	Classification accuracy close to full training dataset
2021	Yang et al. [27]	Privacy-Preserving Image Classification in 5G Edge Networks	Improve accuracy & reduce device/bandwidth requirements	Good accuracy compared to baseline with limited device data
2023	Tianlun Hu et al. [28]	Inter-Cell Inter-Slice Resource Partitioning	Improved efficiency & sample effectiveness	Faster convergence and better performance than existing methods

1.9 Dissertation Outline

The rest of the dissertation is organized as follows:

- Chapter 2: Theoretical Background showed an overview of the theoretical basics of the methods used in the proposed system.
- Chapter 3: Proposed Systems describes the dynamic stages of the proposed system design, discussing and detecting each stage in detail, as well as the algorithms used.
- Chapter 4: Experimental Results and Discussion describe the experimental outcomes of the proposed system and discusses them in detail.

- Chapter 5: Conclusions and Recommendations reviews the conclusions of the dissertation and provides some recommendations for future work.

Chapter Two
Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

This chapter provides an introduction to the essential ideas and methodologies used throughout this dissertation. The subjects included in this chapter affect to the evolutionary progress of 5G technology, the notion of (C-RAN), the practice of network slicing, the use of ML and DL techniques, the application PSO, the challenges associated with imbalanced data, the analysis of tabular data, the development of tabular Deep Learning models, the provision of dataset descriptions, the implementation of TL, and the valuation of evaluation measures. Finally, this chapter discusses the application of TL and the importance of using appropriate evaluation measures.

2.2 The Rise of 5G

The definitive standard for 5G mobile network technology was created in December 2017 by **3**rd **G**eneration **P**artnership **P**roject (3GPP) and the ITU is responsible for developing 5G technical standards (IMT-2020). The 5G mobile network uses a high-band spectrum to provide very fast speeds and minimal latency. Because of 5G higher bandwidth; it connect billions of devices with low latency and very high speed. Also, according to uploading and downloading speeds 5G will pass up to 100 times than the **F**ourth **G**eneration (4G) standards. Table 2.1 explains a comparison between mobile generations [29, 30].

Table 2.1: Comparison between Mobile Generations [30]

Criteria	**Mobile Generations**			
	Second Generation 2G	**Third Generation 3G**	**4G**	**5G**
Introducing year	1993	2001	2009	2018
Technology	GSM	WCDMA	LTE, WiMAX	MIMO, mmWaves
Access System	TDMA, CDMA	CDMA	CDMA	OFDM, BDMA
Switching Type	Circuit, packet	Circuit, packet	packet	packet
Network	PSTN	PSTN	packet Network	packet Network
Internet service	Narrowband	Broadband	Ultrabroadband	Wireless World Wide Web
Bandwidth	25 MHz	25 MHz	50 MHz	30-300 MHz
Speed	64 Kbps	8 Mbps	300 Mbps	10-30 Gbps
Latency	300-1000 ms	100-500 ms	20-30 ms	1-10 ms
Mobility	60 km	100 km	200 km	500 km

When comparing 5G to the previous generation, 4G, it becomes evident that 5G holds several unique characteristics and offers numerous benefits. Including [30, 31]:

1) **High capacity:** 5G can support up to 100 times more devices than 4G, making 5G ideal for use in crowded areas such as stadiums and concert venues.
2) **Reduced latency:** Latency, defined as the duration required for data to traverse between two points, exhibits a notable decrease in the context of 5G compared to 5G predecessor, 4G. This feature renders well-

suited for use in applications that need instantaneous communication, such as autonomous vehicles and virtual reality experiences.

3) The 5G technology has a superior connection density. This characteristic enables the network to accommodate a larger number of devices within a confined geographical region. The importance of this is showed in several applications, including IoT and smart cities.
4) The high throughput ability of 5G enables it to achieve maximum data speeds of up to 20 Gbps, demonstrating an important improvement of 20 times compared to the previous generation, 4G. This feature compatible for the purposes of downloading sizable files, streaming high-definition video, and engaging in online gaming activities.
5) The 5G of wireless communication technology shows enhanced spectral efficiency. This improvement allows 5G using the same allocation of frequency spectrum to transport a greater volume of data.
6) 5G technology capability to provide unified connectivity, enabling devices to transition between several cellular networks without facing any disruptions in their connection. The significance of this is seen in domains such as augmented reality and virtual reality.
7) **Extensive coverage:** 5G technology has broad coverage capabilities, enabling 5G use in both rural and urban regions.
8) **Increased network energy efficiency:** 5G is more energy efficient than 4G, which can help to reduce the environmental impact of wireless communication.

2.3 Cloud Radio Access Network (C-RAN)

The rapid growth in the number of connected devices is increasing the cost of deploying new network technologies. This is because the cost of both CAPEX and OPEX is rising. To meet the needs of a variety of services, the C-RAN architecture has been proposed as a possible design for 5G [3]. C-RAN an effective solution for constructing high performance and low-cost RAN in 5G. C-RAN depending on the concept of pooling the base station in central servers, that submit exceeding performance gains and reduces execution cost [32].

2.3.1 Cloud radio for mobile networks

In traditional cellular networks, a base station serving the cells that is in the same coverage area and users communicate with the base station. The main base station functions consist of radio and baseband processing functionalities. Power amplification, frequency filtering and digital processing are the responsibility of the radio function, while the functions of baseband processing are responsible of coding process, modulation process , **F**ast **F**ourier **T**ransform (FFT), etc.. In **F**irst **G**eneration (1G) and 2G mobile networks architecture, baseband processing and radio functionality is united inner a base station. Generally, the antenna is located in a few meters from the radio station and connected with it by coaxial cables that suffers from high losses, Figure 2.1 shows the Traditional Macro Station [33].

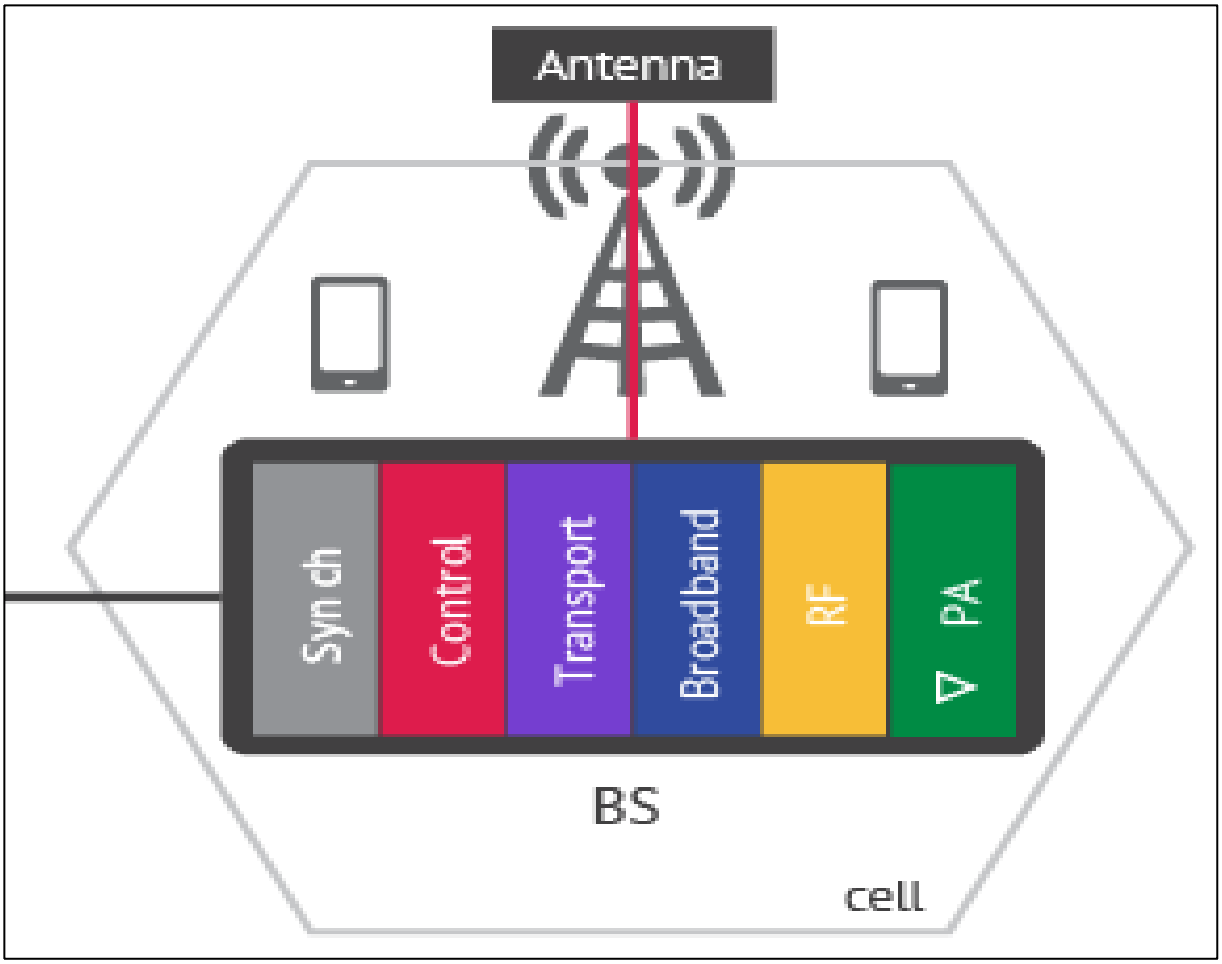

Figure 2.1: Traditional macro station [33]

While, in 3G and 4G the base station is isolated into a signal-processing unit and a radio unit, as shown in Figure 2.2 [33].

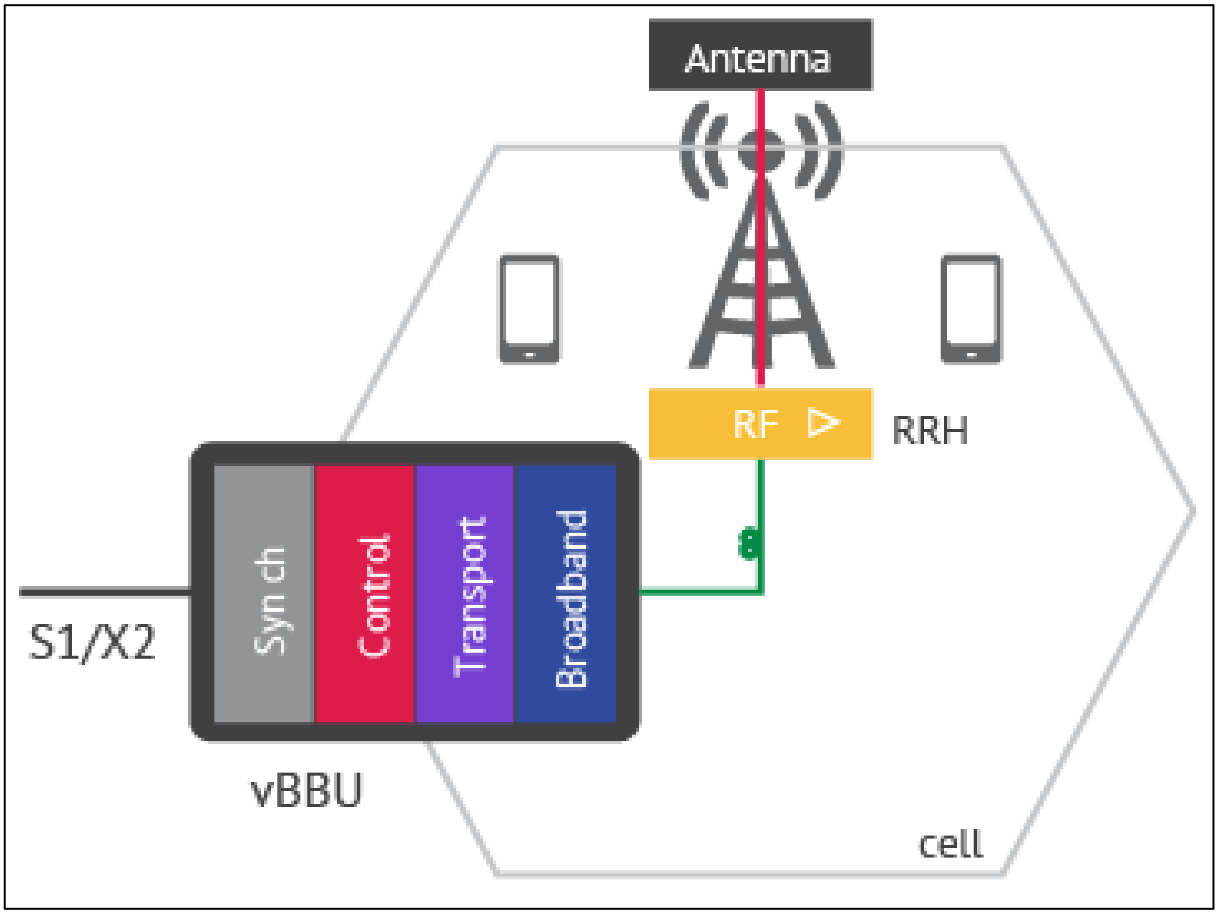

Figure 2.2: Base Station with Remote Radio Head [33]

The radio unit is called a Remote Radio Unit (RRU) or Remote Radio Head (RHH). On the other hand, processing part is called a Data Unit (DU) or BBU. In C-RAN, a BBU/DU is the entity that the BBUs are centralized and it is virtualized and shared between cell sites as shown in Figure (2.3) [33].

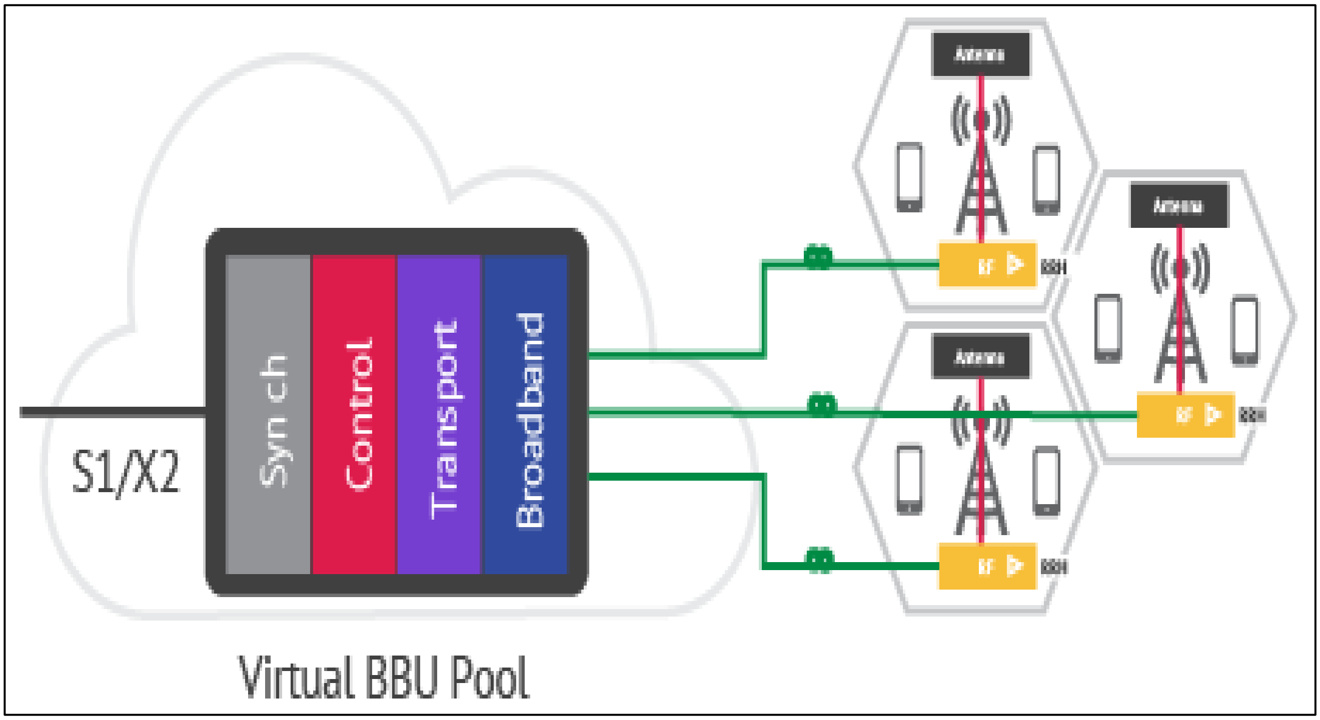

Figure 2.3: Cloud Radio Access Network with Remote Radio Head [33]

Lately, in order to offer best services for users and confirm a well profit, the operators transformed from previous network (shown in Figure 2.4) into new one [34].

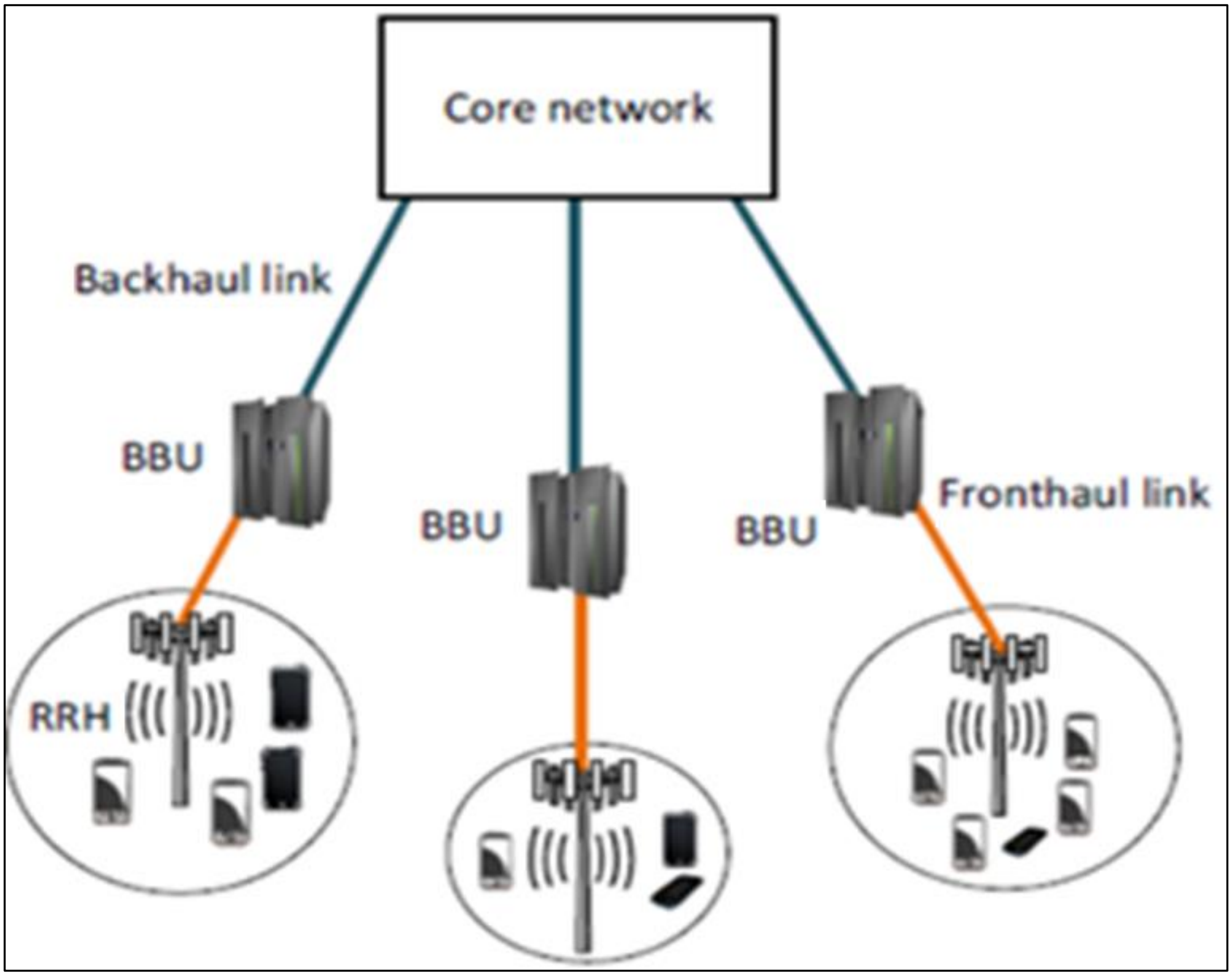

Figure 2.4: Radio access network [34]

Moreover, a cloud computing become a promising choice for both mobile operators and Information Technology service providers. Facilities of cloud computing are used by mobile operators to getting low-cost operation and to forming a pool of resources shared in a large geographical area. A new paradigm based on cloud computing with power efficient infrastructure, collaborative radio and centralized processing proposed by a few operators (China Mobile Research Institute) called C-RAN (shown in Figure 2.5) which considering as the infrastructure architecture to 5G [34].

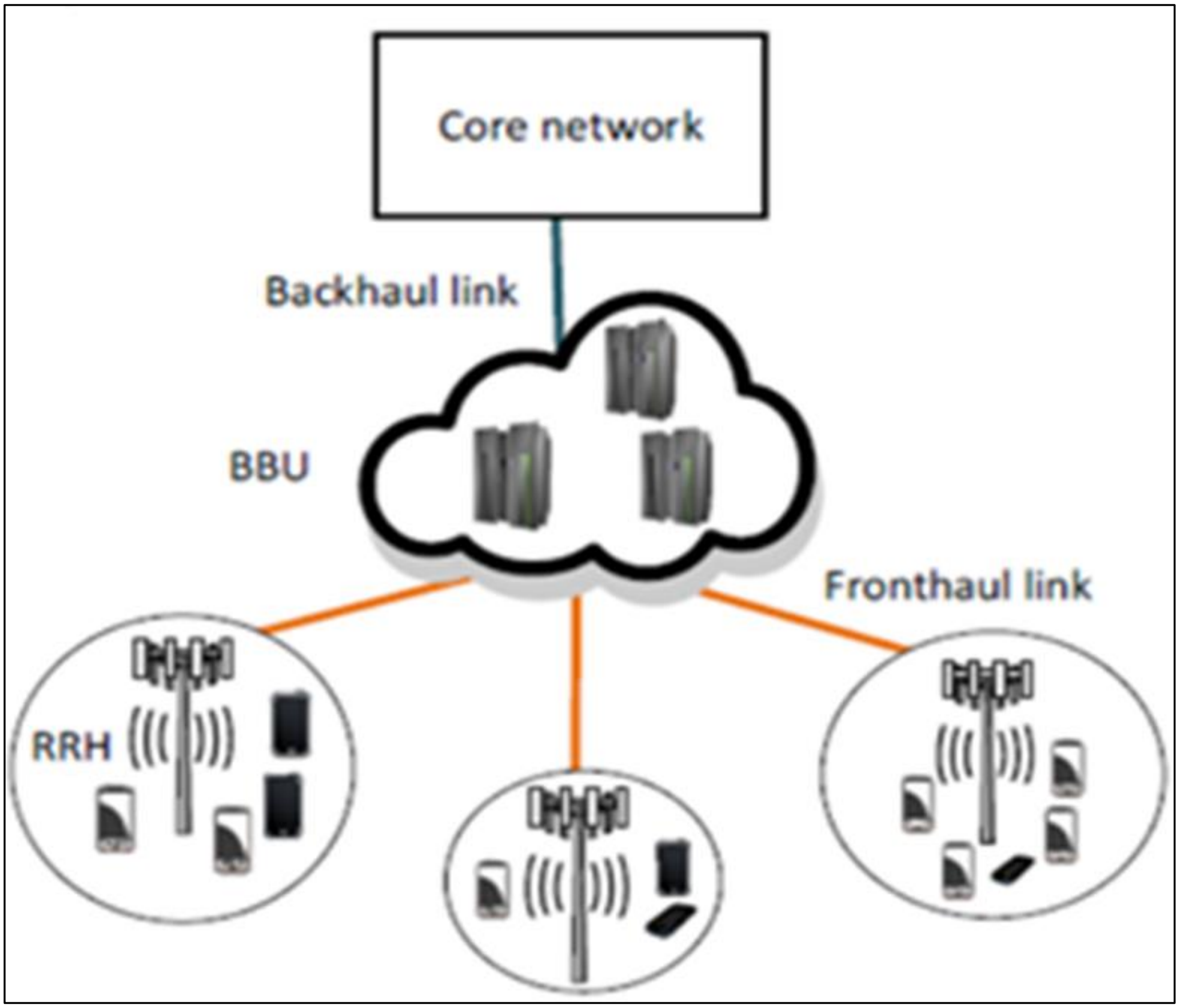

Figure 2.5: Cloud radio access network [34]

2.3.2 C-RAN architecture

Generally, the C-RAN architecture consists of three basic components: RRUs, the fronthaul network and the centralized BBU pool. In the downlink, the RRUs responsible for transmitting the Radio Frequency signals to user equipment, whereas in the uplink, responsible for forwarding the baseband signals from user equipment to the centralized BBU pool. Optimizing the network resource allocation for RRUs and processing baseband signals are the function of BBU pool, which run as virtual base stations. In addition, the unprocessed Radio Frequency signal is transport from antenna to virtual BBUs by the fronthaul network. Also, backhaul connecting a BBU pool with the mobile core network

[35]. Technology of virtualization is used in this architecture, where the BBU's functions installed as software on the physical servers called the virtual BS as shown in Figure 2.6 [32].

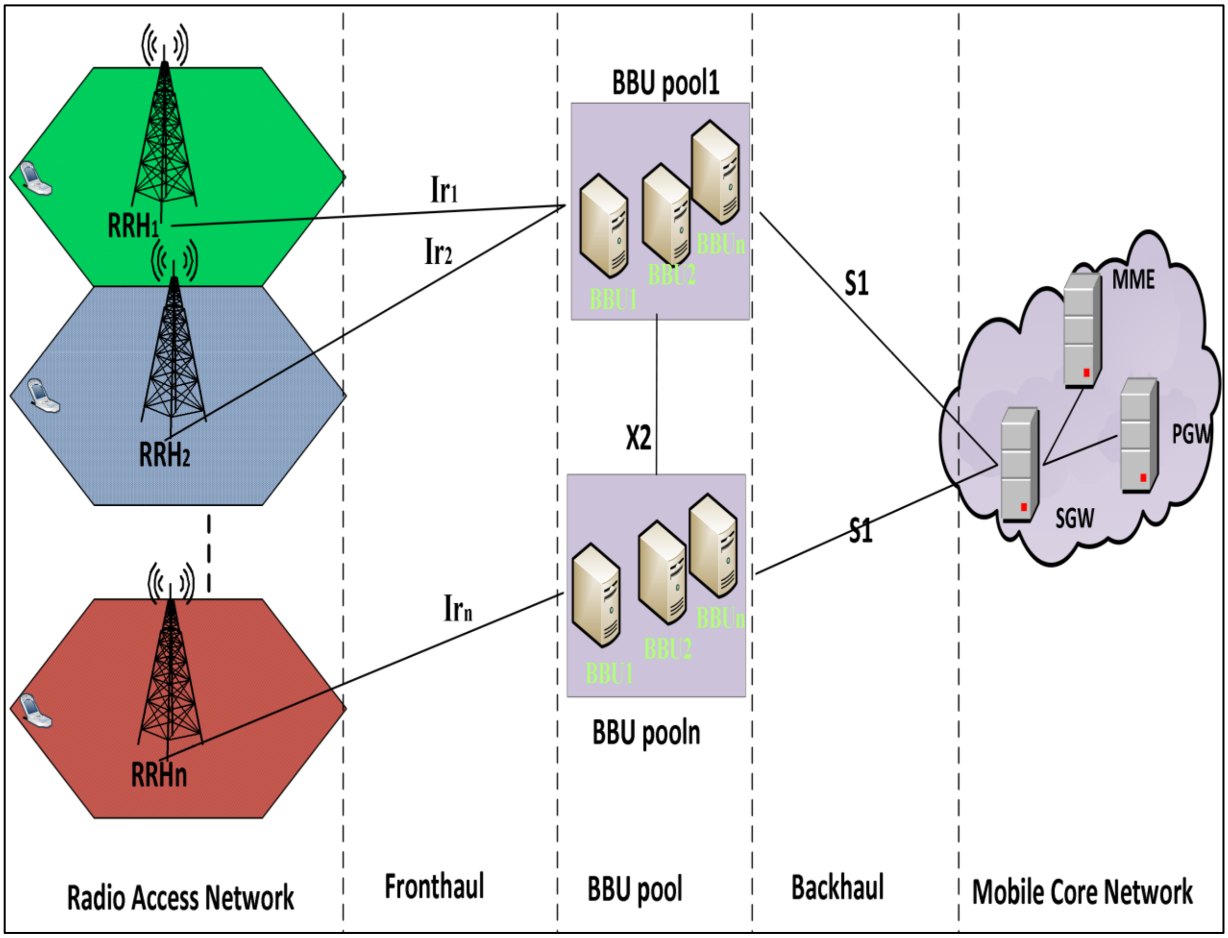

Figure 2.6: Virtualized cloud radio access network architecture [32]

2.3.3 C-RAN benefits

In C-RAN, **C** letter has several features such as **C**entralization, **C**loud computing, **C**ooperative radio, and **C**lean system [32]. C-RAN has many benefits [36]:

1) Reducing inter-channel interferences by processing and joint scheduling and increasing in capacity and this attributed to existence of centralized BBU pool in which the resources can be shared cooperatively and dynamically between multiple cells of different operators.

2) Improving system throughput; in areas that express high throughput needs C-RAN allows dense RRH deployment modules.
3) Reducing OPEX and CAPEX; this attributed to centralization of BBU and isolation from RRHs thus lead to great reduction in maintenance and deployment costs.
4) Coexistence of multiple standards; multiple standards can be useful used for each of user demands and supported by Centralized BBU.
5) Green Radio; in C-RAN number of cell sites is reducing thus improves energy efficiency also led to reducing in power consumption of site and equipment. Also, in the period of low traffic C-RAN has the ability of turn off the BBUs that under level of utilization and migrated their traffic to effective BBUs.

2.3.4 Virtualization concept in C-RAN

Virtualization can be defined as a technology that has an ability to create a logical, isolated entity from a physical entity. Furthermore, technology of virtualization is used for years for desktop, network, storage, and data virtualization. Virtualization of network is a significant technology to architecture of C-RAN. C-RAN comprises a number of virtual links and virtual nodes. Virtual networks are used in the heterogeneous network that will provide low cost, flexible control, diversified applications and efficient resource usage [32].

Technology of network virtualization separates applications, data storage, management control and operating systems. The **V**irtual **B**ase **S**tations (VBSs) called on the functions of a BS that are considered as software instances. A common resource such as systems and hardware are shared by multiple VBSs as shown in Figure 2.7, thus lead to flexible and efficient utilization [33].

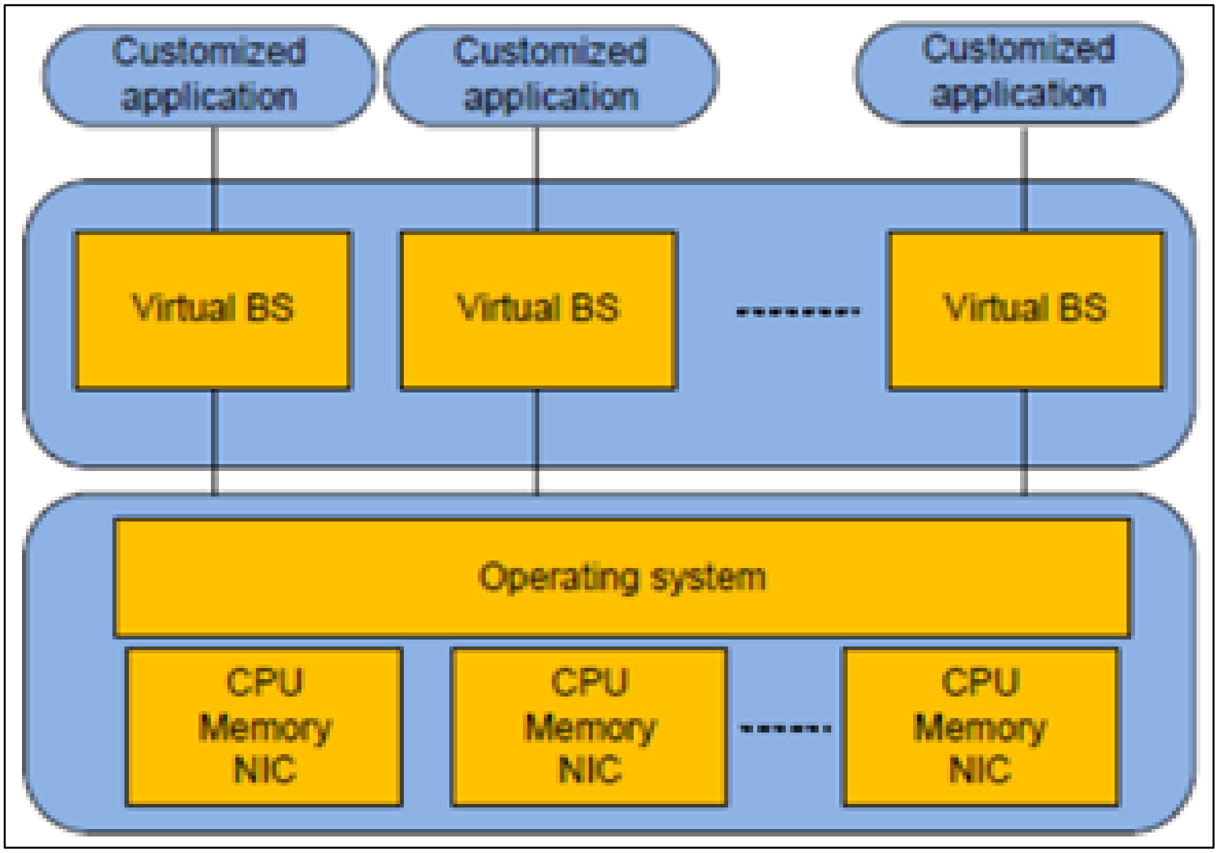

Figure 2.7: Virtualization Concept in Cloud Radio Access Network [33]

The most important key objectives in controlling and managing network; is decreasing operational costs and increasing network resource utilization. Two promising technologies come into seen SDN and NFV which have a massive benefits, including easier management needs, best resource utilization and reducing operational costs [37]. SDN shown in Figure 2.8.

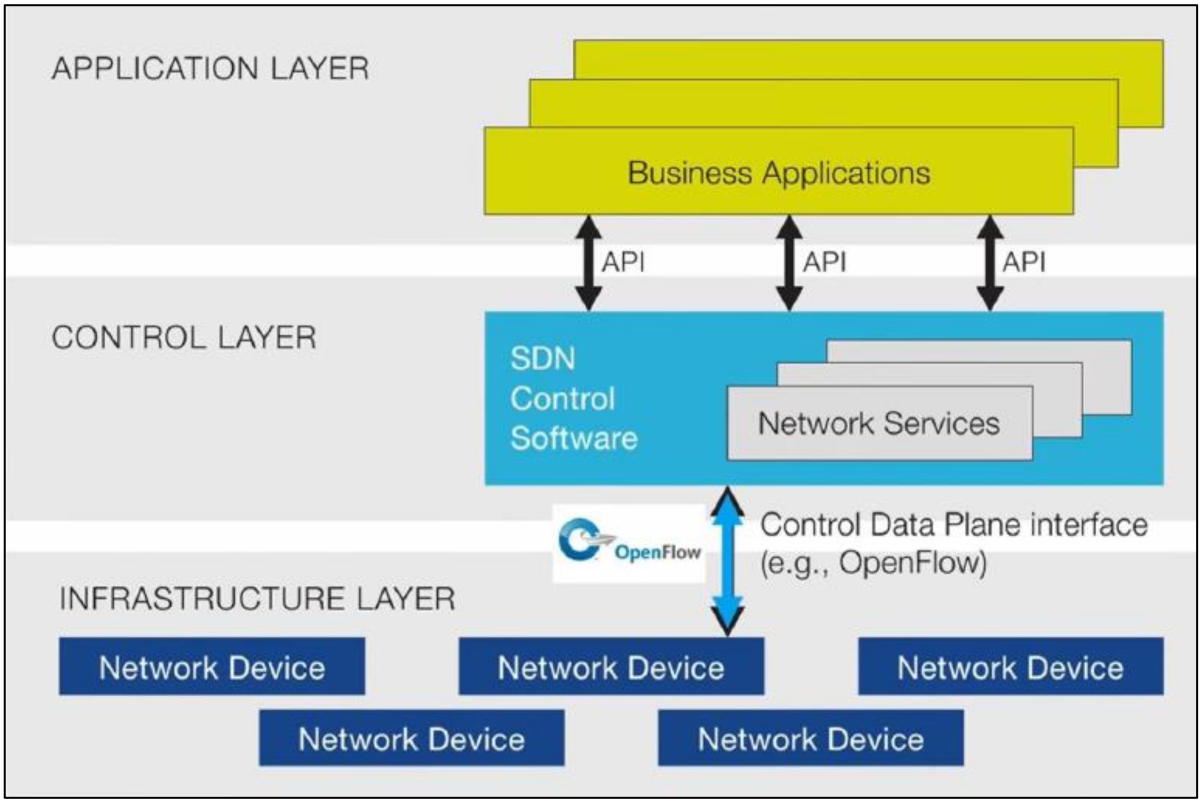

Figure 2.8: Software defined network [37]

While NFV shown in Figure 2.9.

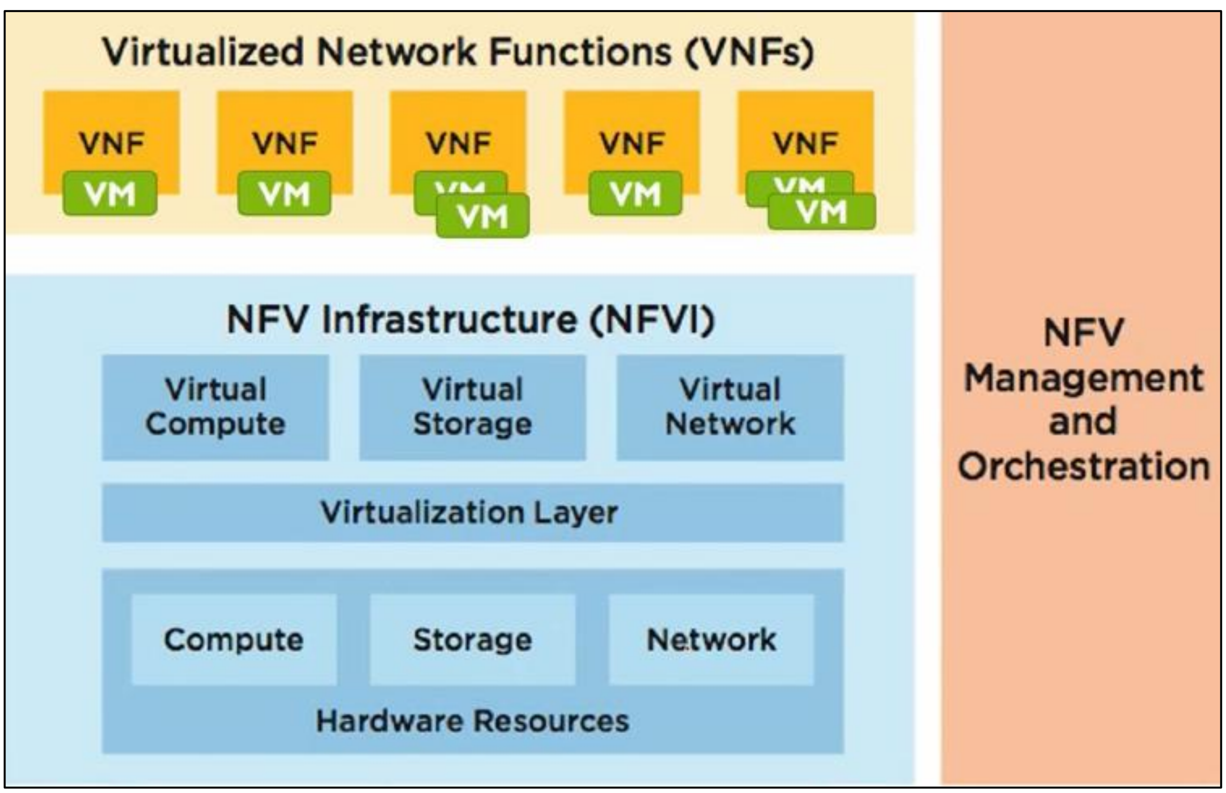

Figure 2.9: Networking function virtualized [37]

2.4 Network Slicing

The process of producing multiple virtual networks from partitioning one physical network is called Network Slicing.

2.4.1 The Main concept

Each network slice has own topology, virtual resources, provisioning rules and traffic flow. Resources can be shared between network slices even they are logically isolated. In the 5G network, there are different network slices that required for the needs of different users. The main goal of network slicing is to segregate the physical network and group traffic so that each tenant (tenant refers to users who have certain privileges and access rights to shared resources) is isolated. This can help to improve the performance and reliability of services, and reducing costs by more efficiently utilizing network resources. Network slicing is a key feature of 5G networks, that supporting a wide range of use cases with varying service requirements. For example, one slice could be dedicated to industrial automation, another to mobile gaming, and another to augmented reality. Each slice could have different performance levels and QoS guarantees [1, 38, 39]. In addition, network slicing is dominated by software components, which enable on demand and real-time reconfiguration. Moreover, underutilized resources can be leased as network slices, thus generating new gain chances for providers of infrastructure and maximizing utilization of resource [40].

As depicted in Figure 2.10, network slicing is a virtual network architecture that will allow the powerful and flexible capacity of establishing many logical networks on top of a single physical infrastructure [41, 42].

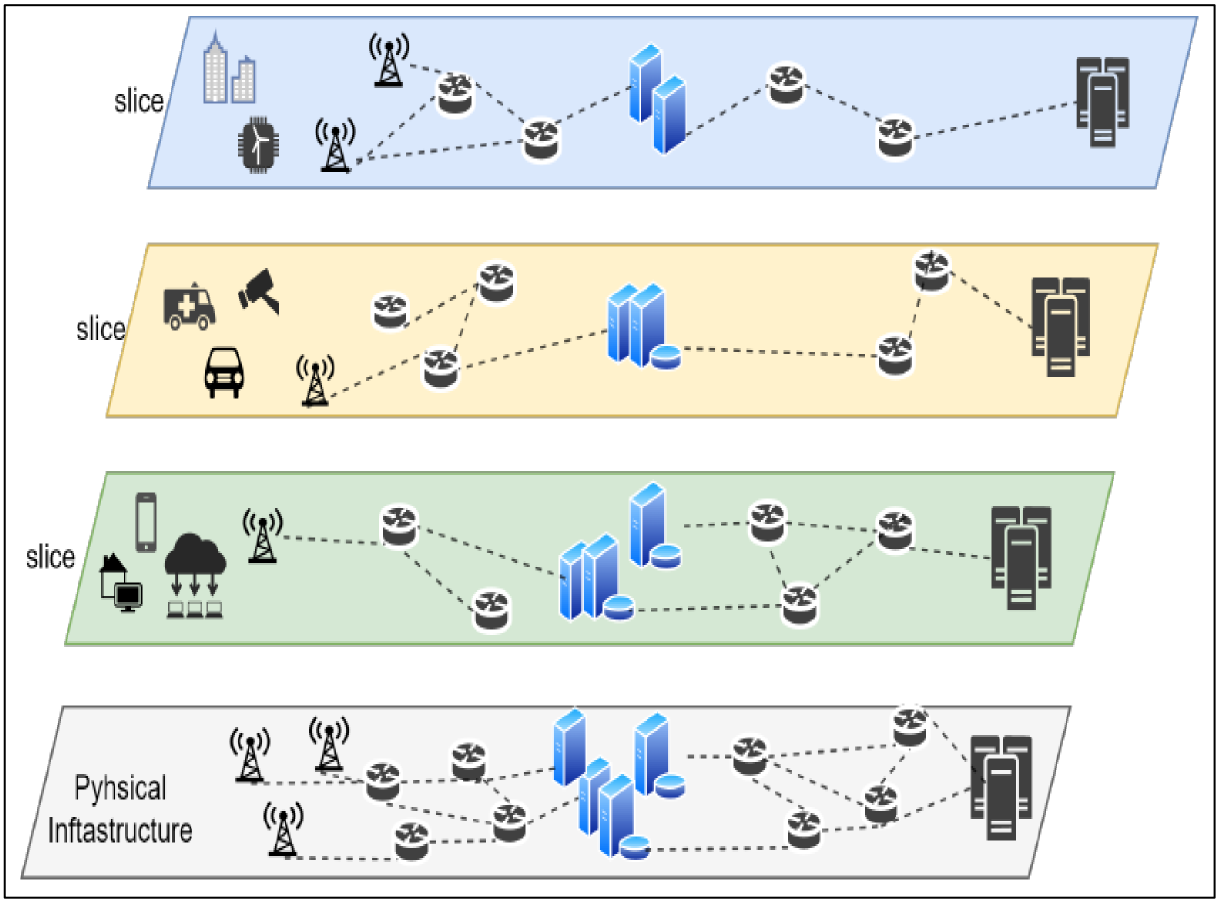

Figure 2.10: Network slicing concept in 5G [41]

2.4.2 Network slicing types

The promise of 5G networks is not simply a development of 4G networks with higher potential throughput and larger spectrum bands. 5G also aims to deal with new services and business opportunities that have own network slice. These services are classified by 3GPP into three main families of use cases [43-46]:

1) **E**nhanced **M**obile **B**roadband (eMBB): This service is a collection of use cases requiring high data rates across a wide coverage area. This service is designed to provide high-speed mobile internet access for users, eMBB can support data rates up to 100 Gbps, which is much faster than 4G LTE. This will enable new applications such as **V**irtual **R**eality (VR), **A**ugmented **R**eality (AR), and 4K/8K video streaming.

2) **U**ltra **R**eliable and **L**ow **L**atency **C**ommunications (URLLC): This service is for applications with very low latency and high reliability requirements, such as vehicle-to-everything (V2X), industrial control applications. URLLC can achieve latency as low as 1 millisecond and reliability as high as 99.999%.
3) **M**assive **M**achine **T**ype **C**ommunication (mMTC): This service is important for supporting a very large number of devices in a small area, forming an IoT; such as sensors and actuators. mMTC can connect to 1 million devices per square kilometer. This will enable new applications such as smart cities and smart agriculture.

Figure 2.11 presents the 5G services and opportunities. While Figure 2.12 shows 5G Network Slicing Types.

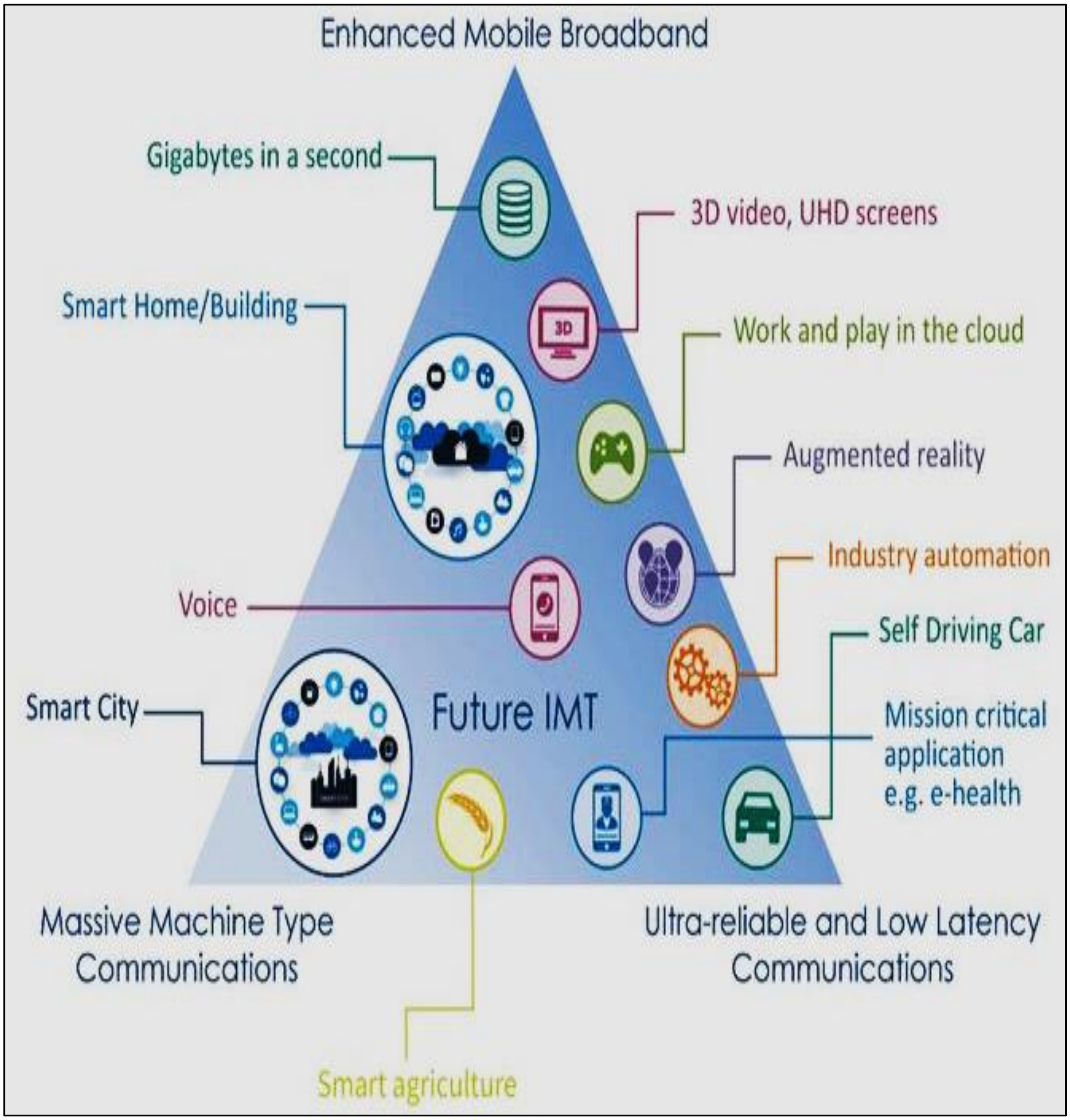

Figure 2.11: 5G services and opportunities [32]

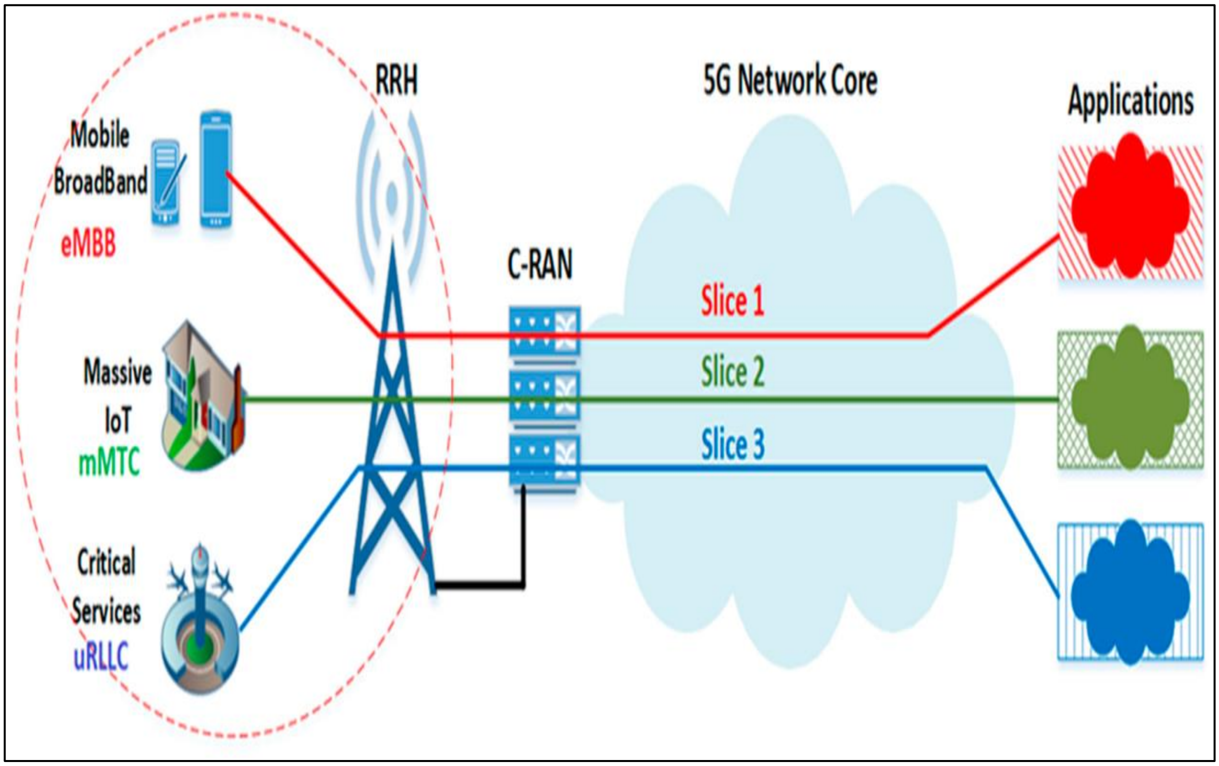

Figure 2.12: 5G Network Slicing Types [47]

2.4.3 Network slicing benefits

As number of use cases and the scale of deployments increases, the risk of resource starvation increases and radio resources become scarcer. Furthermore, services that supported by 5G system must include high bit rates and ultra-low latency. Network slicing facilitates the coordination of radio access network (RAN) resources, functions, and QoS regulations, while ensuring the allocation of resources to various service categories [48].

Here are some benefits of network slicing [6, 49]:

1) Enhanced dynamicity: The use of network slicing has the potential to enhance the dynamicity of networks, hence increasing their ability to promptly adapt to changing needs. The significance of this matter lies in many applications that include a broad spectrum of demands and need prompt deployment.

2) Enhanced security and privacy may be achieved by the use of network slicing, which offers the potential to enhance the security and privacy measures of various applications. By implementing a strategy of isolating each individual slice from the rest, it is possible to effectively mitigate potential threats and safeguard confidential information.
3) One potential benefit of network slicing is the optimization of resource consumption. This is achieved by distributing resources to different slices depending on their specific requirements. This has the potential to mitigate expenses and enhance operational efficiency.
4) One of the advantages of network slicing is the potential for more flexibility in network deployment and management for network operators. This may assist organizations in catering to the demands of a broader spectrum of clientele and diverse use cases.
5) One potential benefit of network slicing is the potential reduction in operating expenses. This is achieved via the automation of various network management duties. This has the potential to allocate resources for other endeavors, such as fostering innovation and enhancing customer service.

In general, network slicing is a robust technology that has the potential to enhance the integrity, and adaptability of 5G networks.

2.5 The Research Dataset

The primary constraint addressed in this research was to lacking of an actual dataset related to 5G Network Slicing.

2.5.1 Dataset Description

After extensive research, a 5G Network Slicing dataset was generated by utilizing information gathered by researchers at [19]. These researchers, associated with [19], gathered information by examining parameters derived from ITU standards papers, as well as other European studies and analysis documents conducted by telecommunications corporations. These parameters are utilized to evaluate and measure the quality and performance of the 5G network, which are referred to as KPIs. They gathered information about 13 parameters and 9 network slicing services. These 13 parameters are considered to be features of the generated 5G network slicing dataset, while the 9 services are represented as classes. The features and their corresponding descriptions are displayed in Table 2.2, while the classes and their corresponding descriptions are explained in Table 2.3. Table 2.4 shows the values of each of the 13 features for each of the 9 classes.

Table 2.2: Features of 5G Network Slicing dataset [19]

No	Features	Features description
1	Latency	The time it takes for data to travel from the source to the destination in 5G network is measured in milliseconds (ms).
2	Jitter	The variation in the delay of received packets in 5G network is measured in milliseconds (ms).
3	Bit Rate	The rate at which data is transmitted in 5G network is measured in megabits per second (Mbps).
4	Packet Loss Rate	The percentage of packets that are lost or fail to reach their destination is measured percent ratio (%).
5 and 6	Peak Data Rate DL/UL	The maximum achievable data rate in (Down Link/ Up Link) of the 5G network is measured in gigabits per second (Gbps).
7	Mobility	The ability of a device or user to move while maintaining a seamless connection to the 5G network is measured in kilometers per hour (km/h).
8	Reliability	The ability of the network to consistently deliver data without errors or interruptions is measured percent ratio (%).
9	Service Availability	The percentage of time that service or network is operational and accessible to users is measured percent ratio (%).
10	Survival Time	The duration for which a device can maintain a connection with the network when operating under challenging conditions is measured in milliseconds (ms).
11 and 12	Experienced Data Rate DL /UL	The perceived average data rate experienced by a user in a given time period in either the Down Link or Up Link of the 5G network is quantified in megabits per second (Mbps).
13	Interruption Time	The duration of a service interruption or loss of connectivity in a 5G network is measured in milliseconds (ms).

Table 2.3: 5G Network slicing dataset classes [19]

No	Classes	Classes description
1	Ultra-High Definition (UHD) Video Streaming	5G can deliver UHD video streaming with low latency and high reliability, which is ideal for live sports, concerts, and other events.
2	Immersive Experience	5G can be used to create immersive experiences, such as virtual reality and augmented reality. These experiences can be used for gaming, education, and training.
3	Smart Grid	5G can be used to improve the efficiency and reliability of the power grid. Sensors and actuators connected to 5G can be used to monitor and control the grid, and to respond to outages and other disruptions more quickly.
4	e-Health	5G can be used to improve the delivery of healthcare services. For example, 5G can be used to transmit medical images and data in real time, and to provide remote monitoring and diagnosis.
5	Intelligent Transportation Systems (ITS)	5G can be used to improve the efficiency and safety of transportation. For example, 5G can be used to connect vehicles to each other and to infrastructure, such as traffic lights and road sensors. This can help to prevent accidents and to optimize traffic flow.
6	Voice over 5G (Vo5G)	5G can be used to provide high-quality voice and video calling services. Vo5G services can be used for mobile phones, tablets, and other devices.
7	Connected Vehicles	5G can be used to connect vehicles to each other and to infrastructure. This can help to improve safety and efficiency, and to provide new services, such as real-time traffic information and autonomous driving.
8	Industry Automation	5G can be used to automate industrial processes. For example, 5G can be used to connect sensors and actuators to control machines and robots. This can improve efficiency and productivity.
9	Video Surveillance	5G can be used to improve the quality and efficiency of video surveillance. 5G can be used to transmit video data in real time, and to provide more detailed and accurate images. This can help to improve security and to prevent crime.

Table 2.4: Features values of generated 5G network slicing dataset [19]

Classes \ Features		(UHD) Video Streaming	Immersive Exper.	Smart Grid	e-Health	(ITS)	(Vo5G)	Connected Vehicles	Industry Auto.	Video Surv.
Latency	Min	4	7	5	1	10	20	3	1	10
	Max	20	15	50	10	100	150	100	50	50
Jitter	Min	1	1	0.1	1	1	1	0.1	0.01	1
	Max	5.84	20	1	10	20	30	0.44	0.1	5
Bit Rate	Min	1	1	0.1	1	0.1	1	1	1	1
	Max	0	50	1	16	0.5	10	10	10	10
Packet Loss Rate	Min	0	0	0	0	0	0	0	0.0000001	0.01
	Max	1	5	0.0001	1E-08	0.1	0.1	0.001	0.00000001	0.001
Peak Data Rate DL	Min	1	1	1	0.1	1	1	1	1	0.01
	Max	20	20	20	0.3	20	20	20	20	0.05
Peak Data Rate UL	Min	1	1	1	0.1	1	1	0.001	1	0.01
	Max	10	10	10	0.3	10	10	0.025	10	0.12
Mobility	Min	0	0	0	0	50	0	50	0	0
	Max	500	30	0	120	500	500	250	30	320
Reliability	Min	95	95	99.9	99.9999	99.999	99.9	99.999	99.999	99
	Max	99	99	99.99	99.99999	99.9999	99.99	99.9999	99.9999	99.99
Service Availability	Min	99	99	99.999	1	99	95	95	99.99	99
	Max	99.999	99.99	99.9999	50	99.9999	99	99	99.9999	99.9
Survival Time	Min	8	1	10	1	1	1	1	0	10
	Max	16	10	25	100	50	100	50	100	100
Experienced Data Rate DL	Min	1	1	1	1	1	1	1	1	1
	Max	1000	1000	10	100	10	50	50	10	10
Experienced Data Rate UL	Min	0	0	5	10	1	1	1	1	1
	Max	500	50	10	100	10	25	25	10	100
Interruption Time	Min	1000	0	0	0	1000	0	0	0	0
	Max	3000	1	1	1	10000	1	1	100	20

2.5.2 Dataset challenges

The dataset generated within this dissertation is classified under tabular data type and unbalancing dataset. Tabular data, which includes a collection of items with a set of characteristics, is the most prevalent data format in real-world applications. Many issues emerge when working with tabular data, including lack of locality, mixed feature types (numeric, ordinal, categorical), data sparsity, and a lack of previous understanding of the dataset structure (unlike with text or images) [11].

Although ML and DL approaches succeed at classification and data production tasks on homogenous data (e.g., image, audio, and text data), tabular data remains a barrier. Tabular data sets named after the most recent "**unconquered castle**" for deep models. In contrast to image or language data, tabular data is diverse, resulting in dense numerical and sparse category characteristics. Furthermore, the feature correlation is weaker than the spatial or semantic association in images or audio data Therefore, there are many models appeared recently to deal with tabular data [55].

Moreover, unbalancing data can be more challenging to deal with. In an unbalanced dataset, the number of instances in one class is significantly larger than the number of instances in other classes. Distinguishing between the different classes can become challenging for ML and DL models due to this difficulty. This is because the model can simply predict the majority class for all instances and achieve a low error rate [12].

2.5.3 Dataset preprocessing

Data preprocessing is the process of transforming raw data into a format that can be used for ML and DL, and other data science tasks. Data preprocessing typically performed at the beginning of the model's development pipeline. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for analysis [50]. This dissertation employs a scaling techniques called Z-score Normalization or Z-score standardization; because they can improve the efficiency of the forecasting process.

The importance of its implementation is determined by the need to reduce the artificial intelligence model's sensitivity to the values of the features in the dataset to improve the adequacy of the trained model. This scaling technique is a common technique in statistics. The task done by subtracting the mean from the variable and then dividing by the standard deviation, as illustrated in Equation 2.1 [51]. After applying this scaling to the data, the mean of each feature will be 0 and the standard deviation will be 1. This process is helpful, to confirm that features are on a similar scale, preventing certain features from dominating the learning process due to their larger magnitudes [51, 52].

$$\mathrm{X}' = \frac{(x-\mu)}{\sigma} \tag{2.1}$$

Where X' is the standardized variable, x is the original variable, μ is the mean of the original variable and σ is the standard deviation of the original variable.

2.6 Network Slicing Classification Techniques

This section contains a detailed examination of various classification techniques employed within network slicing and involves clarifications on the structures, functionalities, and enhancements. This section explores various ML models, including DT, RF, and TabNet and then investigates ANN and PSO.

2.6.1 Decision tree

A **D**ecision **T**ree (DT) is an example of a supervised learning approach. This model is based on the structure of a tree. The tree's root, in contrast side, is at the very top. The branches are constructed using objective criteria based on the attributes of the dataset, and the decision tree is continuously grown; or works similarly to a flowchart (Figure 2.13 illustrates a structure of DT). A DT node can be compared as a point of intersection that leads to two separate branches, or "leaf nodes." These nodes, which might become decisions themselves, each reflect a distinct consequence of a particular decision. All of the decisions made will result in a final categorization [53-55].

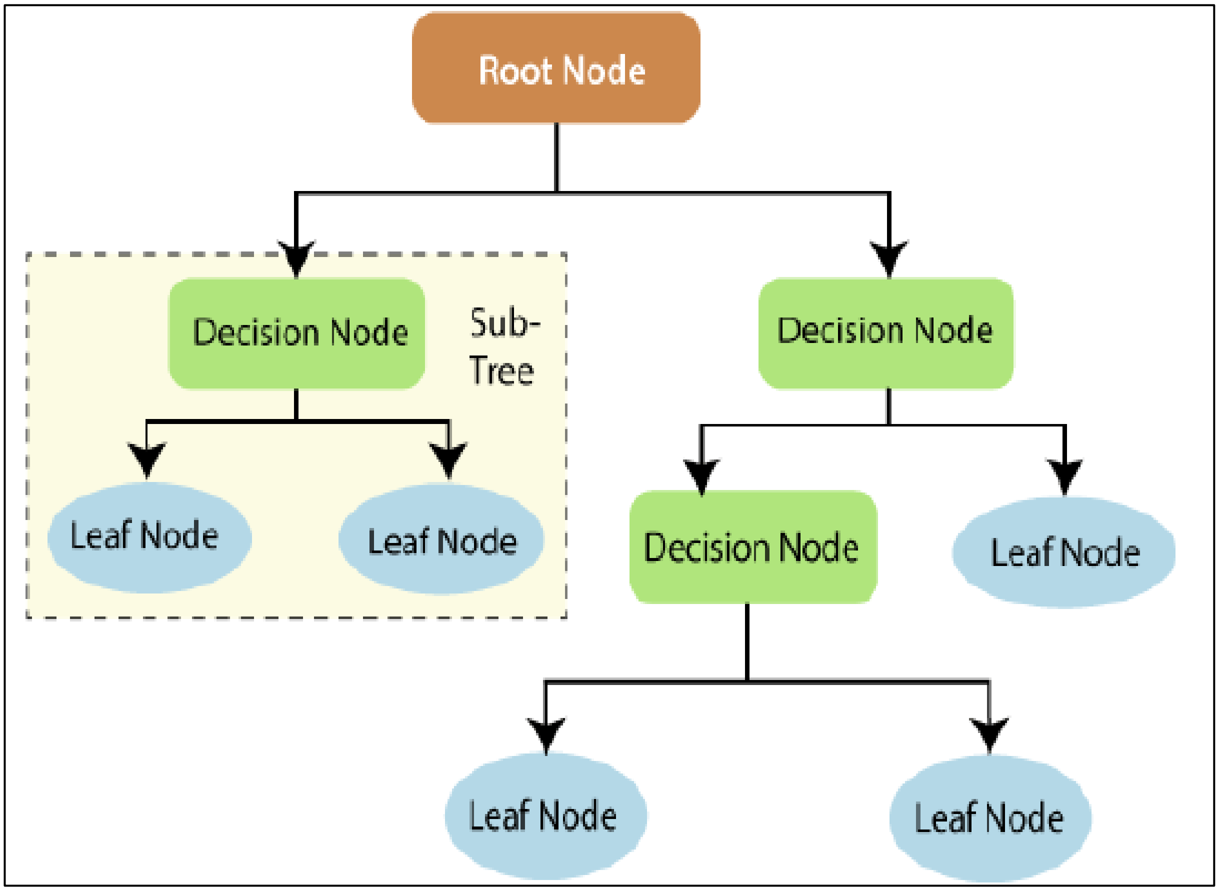

Figure 2.13: A structure of decision tree [56]

2.6.2 Random Forest

The **R**andom **F**orest (RF) is a technique for supervised learning. RF method predicts one (or a group of) outcomes by combining many 'random' binary DTs that comprise the forest. The primary RF working processes are as follows: picking random samples from a dataset, for each sample a DT was built; and receiving a prediction result from each DT, holding a vote for each predicted result, and as a final prediction the prediction result with the most votes were chosen as shown in Figure 2.14 [57, 58].

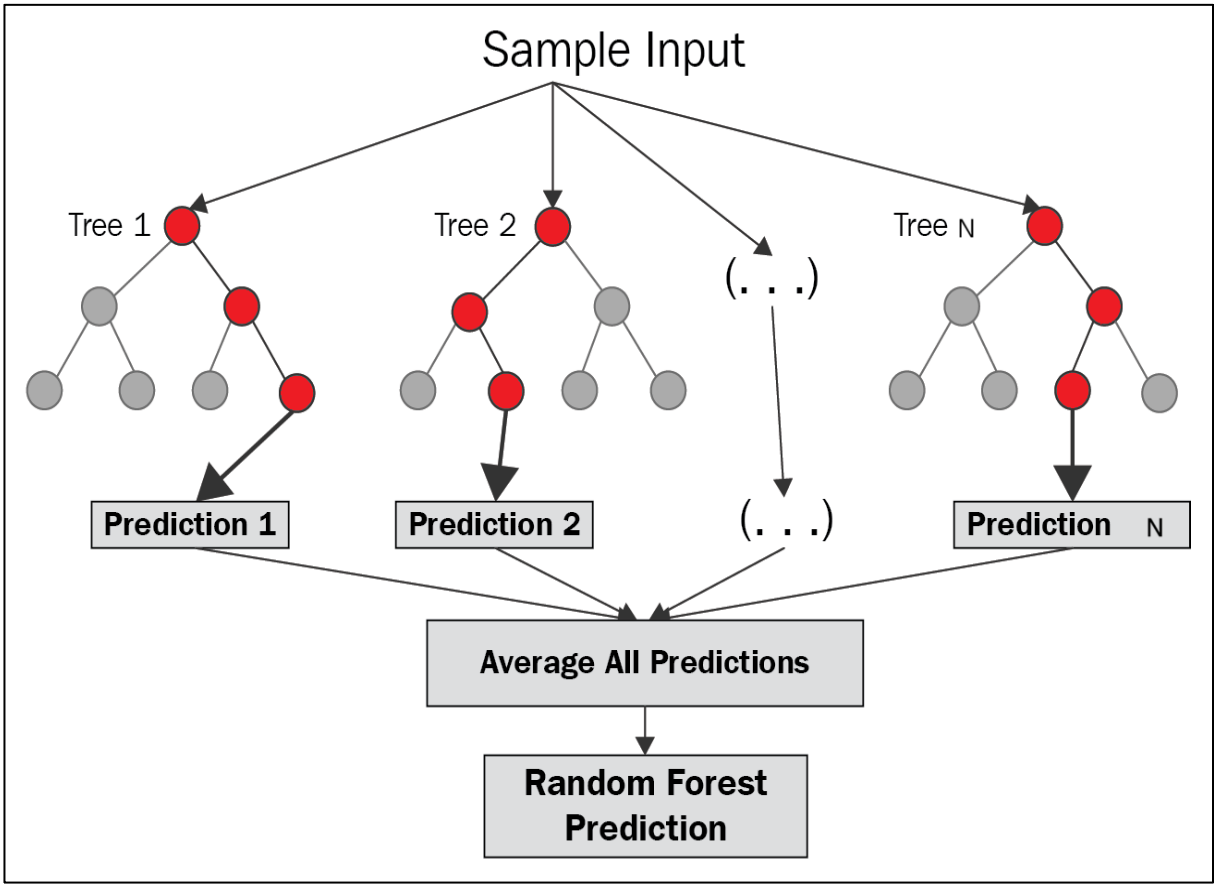

Figure 2.14: The primary random Forest working [59]

2.6.3 Attentive interpretable tabular learning: TabNet

TabNet is a tabular data learning model proposed by (**Arik and Pfister**) at Google Cloud in **2019**, represents a significant innovation in the field of interpretable deep learning models for structured tabular data analysis [11].

2.6.3.1 The main concept

DNNs can improve classification performance on large datasets because they can be trained end-to-end using gradient descent. Tree-based methods, on the other hand, do not use backpropagation, which limits their performance on large datasets. TabNet is a new deep neural architecture for tabular data that combines the benefits of tree-based methods and DNN-based methods. One such approach includes training a DNN to make predictions based on the output of a decision tree [60].

2.6.3.2 TabNet contributions

TabNet is a powerful and flexible DL technique designed especially for the analysis of tabular data and has many important contributions [61]:

1) TabNet apply a sequential attention mechanism to determine the selection of features at each phase of the model. This implies that rather than treating all information with equal importance; the model gains the ability to prioritize the most significant characteristics for each choice.
2) TabNet has a higher degree of interpretability compared to other deep learning models. This implies that there exists a potential to comprehend the decision-making process of the model, hence facilitating the identification and improvement of the model's flaws.
3) The TabNet technique has high computational efficiency, enabling efficient training on extensive datasets. This attribute makes TabNet a favorable option for use cases that prioritize rapid processing, such as real-time fraud detection.

2.6.3.3 TabNet architecture

The TabNet model is specifically intended to handle tasks that include tabular data. In this context, the input data is characterized by a defined structure, with a batch size denoted as B and a feature dimension represented as D. The objective of the model is to generate either a vector or a singular output. The architecture of TabNet, as seen in Figure 2.15 [62].

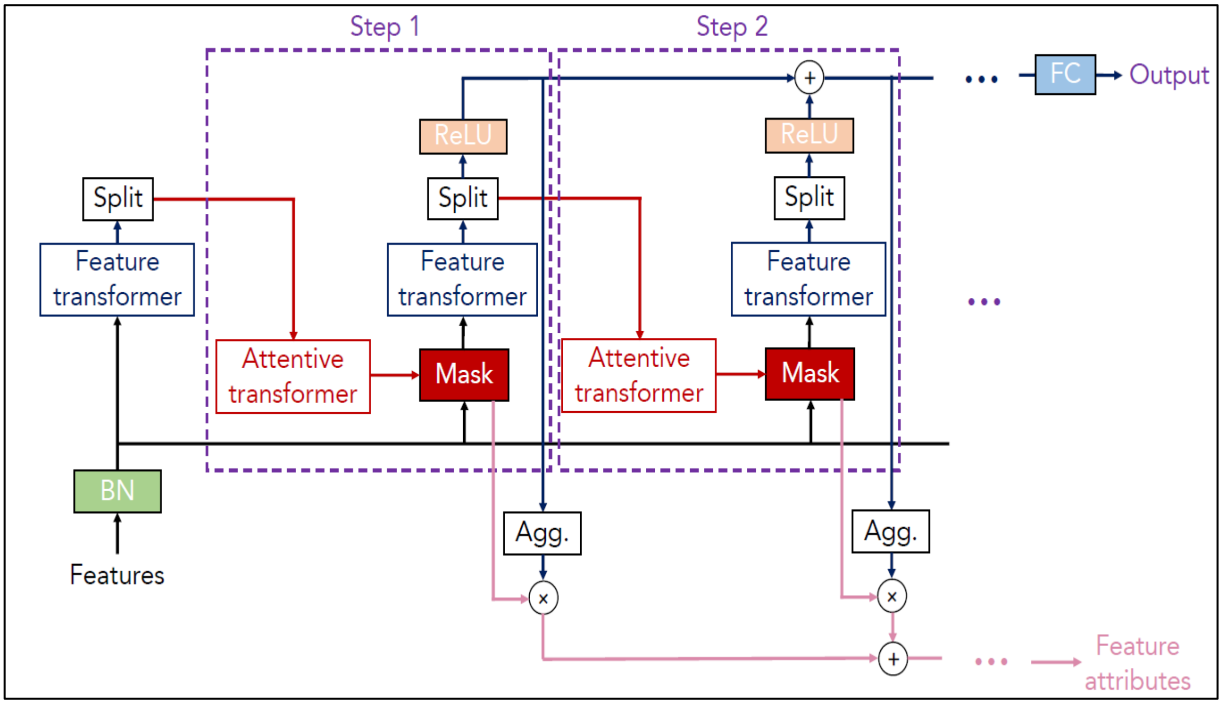

Figure 2.15: TabNet's architecture [62]

TabNet consists of several crucial components [61, 63, 64]:

1) **Feature Transformer Layer:** This layer is responsible for feature calculation and is divided into two parts as shown in Figure 2.16.

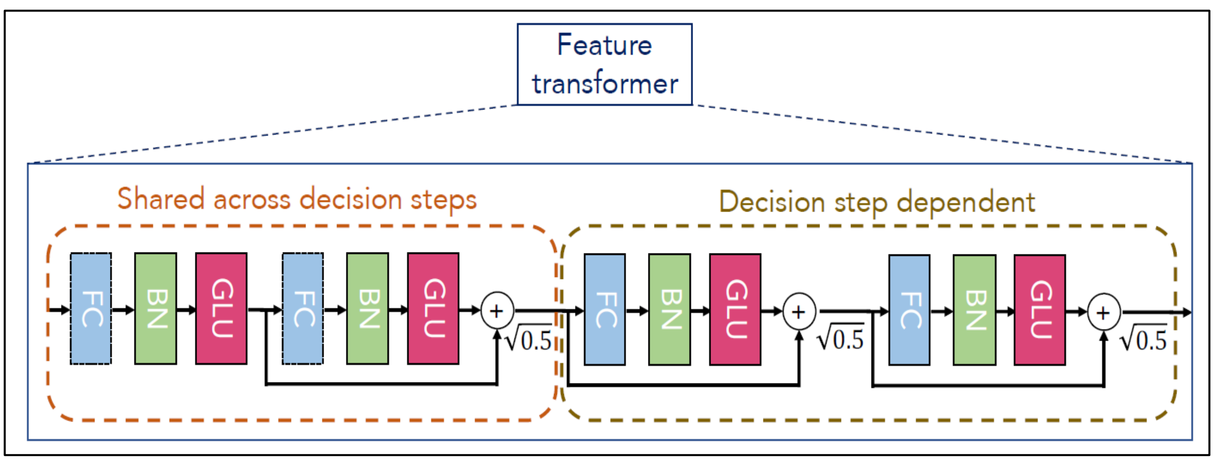

Figure 2.16: The structure of feature transformer layer [62]

The first half of the layer shares parameters across all steps, while the second half has separate parameters for each step.

Each half consists of two blocks of (**F**ully **C**onnected (FC), **B**atch **N**ormalization (BN), and **G**ated **L**inear **U**nit (GLU)):

A. The **FC layer** is a standard NN layer that takes the input data as input and outputs a new representation of the data.

B. The **BN layer** normalizes the output of the fully connected layer, which helps to stabilize the training process.

C. The **GLU layer** is a type of activation function that combines two linear transformations. GLU layer helps to learn nonlinear relationships between the features.

In feature transformer layer; The first FC layer, FC_1( ), performs a common feature calculation on all features. The second FC layer, FC_2( ), performs a separate calculation for each feature. The GLU function then combines the outputs of the two layers. The residual connection (the input features added to the output of the feature transformer layer) is also employed, scaled by $\sqrt{\mathbf{0.5}}$ to maintain network stability. The feature transformer layer can learn more expressive features than traditional DTs because it can perform different operations for each step. This allows the model to capture more complex feature relationships. The mathematical equations for the feature transformer layer are [62]:

$$f\_i(x) = GLU(FC\_1(x) + FC\_2(x)) \times \sqrt{0.5} \tag{2.2}$$

Where $f\_i(x)$ is the feature transformer function at the ith step, x is the input features, $GLU()$ is the gated linear unit function is defined as follows:

$$GLU(x) = x * sigmoid(W * x + b) \tag{2.3}$$

$$sigmoid(x) = {}^{1}/_{(1 + exp(-x))} \tag{2.4}$$

Where W and b are the weight matrix and bias vector, $FC\_1()$ and $FC\_2()$ are the fully connected layers and $\sqrt{0.5}$ is a scaling factor.

D. **Attentive Transformer Layer**: The input data is first passed through a feature transformer layer as mentioned above, which learns a representation of the data. The output of the feature transformer layer is then passed to the attentive transformer layer. The attentive transformer uses attention to learn how the features are related to each other, then generates a mask, which indicates which features are important. The masked features are then passed to the output layer, which makes the prediction. Figure 2.17 represent the structure of attentive transformer layer. Its structure consist of:

- **The FC layer**; which is a standard NN layer that takes the output of the feature transformer as input and outputs a new representation of the data.
- **The BN layer** normalizes the output of the fully connected layer, which helps to stabilize the training process.
- **The sparsemax layer** is a type of activation function that outputs a sparse vector, where each element is either 0 or 1. This layer helps to select the most important features at each decision step.
- **Prior scales** control how often each feature is selected by the attentive transformer. They are calculated using the previous activations of the attentive transformer and the relaxation factor (γ). A higher value of γ means the model will be more likely to select features that have been used in previous steps. A lower value of γ means the model will be more likely to select new features. Prior scales can be used to control the interpretability and predictive power of the model. The mathematical equations for the attentive transformer layer are [62]:

$$M[i] = Sparsemax(P[i - 1] * h\_i(a[i - 1])) \tag{2.5}$$

Where $M[i]$ is the mask layer of the current step and $Sparemax(\,)$ defined as follows:

$$Sparsemax(x) = softmax\big(x - max(x)\big) \tag{2.6}$$

$$softmax(x) = {exp(x)}\big/{sum(exp(x))} \tag{2.7}$$

Where $P[i - 1]$ is the prior scale term at the $(i - 1)th$ step, $h\_i(\,)$ is a trainable function and $a[i - 1]$ is the processed features from the $(i - 1)th$ step.

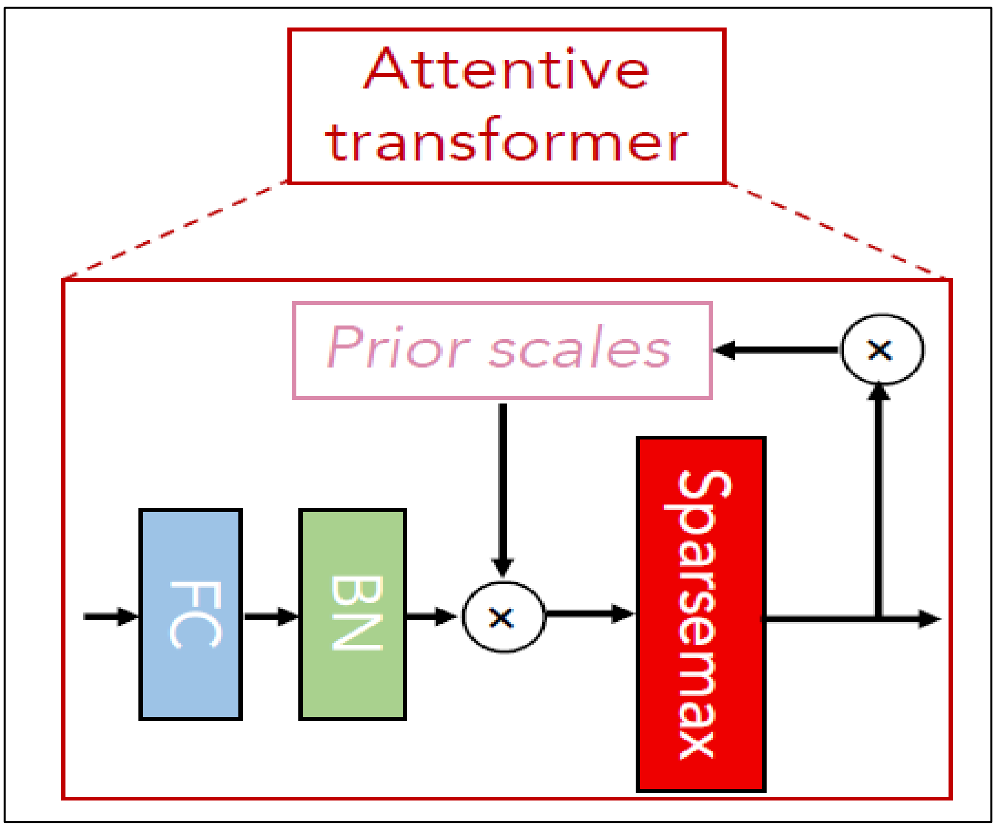

Figure 2.17: The structure of attentive transformer layer [62]

2) **Split Layer:** The split layer first applies the mask layer $M[i]$ to the input features f. This result in a new vector of features that only contains the features that are selected in the current step. The split layer

then splits this vector into two parts, $d[i]$ and $a[i]$. The $d[i]$ part is used to calculate the final output of the model, and the $a[i]$ part is used to calculate the mask layer of the next step. The split layer is a simple but effective layer that allows the TabNet model to learn a sparse representation of the data. The sparse representation makes the model more interpretable and reduces the risk of overfitting.

3) **Like Rectified Linear Units (ReLU) layer:** The input data is first passed through a feature transformer layer, which learns a representation of the data. The output of the feature transformer layer is then split into two parts: a subset that is passed to the attentive transformer and a subset that is passed to the next decision step. The attentive transformer uses attention to learn how the features are related to each other and generates a mask, that indicates which features are important. The masked features are then passed to the ReLU layer, which outputs a non-linear representation of the data. This non-linear representation is then passed to the next decision step. ReLU function is defined as [62]:

$$ReLU(x) = max(0, x) \tag{2.8}$$

Where x is the input value and $max(a, b)$ returns the larger of the two values a and b.

4) **Global Feature Importance**: TabNet computes the global importance of each feature by aggregating the outputs of each step. This process sums the output vectors from each step to obtain a scalar, which reflects the importance of that step in the final result. This scalar is then multiplied by the Mask matrix of that step to determine the importance of each feature for that step. The importance scores from all steps are then summed to obtain the global feature importance.

2.6.4 The Artificial Neural Network (ANN)

The ANN has been discovered to be a highly new and effective paradigm for problem solving. ANNs are a type of ML model inspired by the human brain. They are composed of interconnected nodes, called neurons, which learn to represent the relationships between input and output data. ANNs may be created for specialized applications such as data classification in some scenarios. Perceptron is a simple type of ANN with one layer of neurons. They can only learn linear relationships between input and output data. DNNs are ANNs with multiple layers of neurons. They can learn more complex relationships between input and output data than perceptron. The main difference between ANN and DNNs is the number of hidden layers. ANN have one or two hidden layers, while DNNs can have many hidden layers. The more hidden layers a DNN has, the more complex relationships it can learn [65].

ANN is formed of inputs that are multiplied by weights (is the value of interconnection lines between neurons). These weights are then calculated by a mathematical algorithm that influences neuron activity. Some other function calculates the artificial neuron's output. These functions are called activation functions [66].

The artificial neurons are structured in layers, and their responses are sent "forward" by using forward propagation. Forward propagation is a fundamental procedure in which input data is sent through ANN to generate an output. During the process of forward propagation, the input data undergoes a multiplication operation with the respective weights and then becomes aggregated at each individual neuron. Subsequently, the outcome is subjected to an activation function in order to generate the neuron's output. The procedure is iteratively executed for every individual neuron inside the network, until the final layer, known as the output layer, is attained. The following equations describe the process of forward propagation [67]:.

$$Net_{ij} = \sum_{i=1}^{N} \left(W_{ij} \, * \, xi \right) \tag{2.9}$$

$$output = f(net) \tag{2.10}$$

Where net is the net input to the neuron, W_{ij} is the weight of the connection between the input neuron i and the current neuron, xi is the output of the input neuron i and f is the activation function.

On the other hand, backpropagation is a supervised learning technique used for the training of ANNs. The backpropagation method is characterized by its recursive nature, since ANN operates in a backward manner, starting at the output layer and progressing towards the input layer. At each layer, the method computes the error, which represents the discrepancy between the expected and actual outputs of the neurons inside that particular layer. Equation 2.11 illustrates the computation of the error for a neuron situated in the output layer [67]:

$$error = \left(target_{output} - actual_{output} \right) * act.\_fun.\_der. \tag{2.11}$$

Where $target_{output}$ refers to the desired output of the neuron, $actual_{output}$ represents the actual output of the neuron and $act.\_fun.\_der.$ represents the derivative of the activation function used by the neuron.

Subsequently, the erroneous signal is sent in a backward manner across the network, progressing through each layer. In each layer, the error signal is used to modify the synaptic weights of the connections linking the neurons in the present layer with those in the preceding layer.

Equation 2.12 demonstrates the adjustment of connection weights via the use of backpropagation [67]:

$$w = w - learning\_rate * error \tag{2.12}$$

Where w refers to the magnitude of the connection and $learning\_rate$ is a hyper-parameter that determines the rate at which the learning process takes place.

The backpropagation process is iteratively executed until the discrepancy between the projected outputs and the actual outputs of the network is reduced. Back propagation training is based on the gradient decent rule, which seeks to adjust weights and minimize system error in the network [68]. Figure 2.18 shows a structure of ANN.

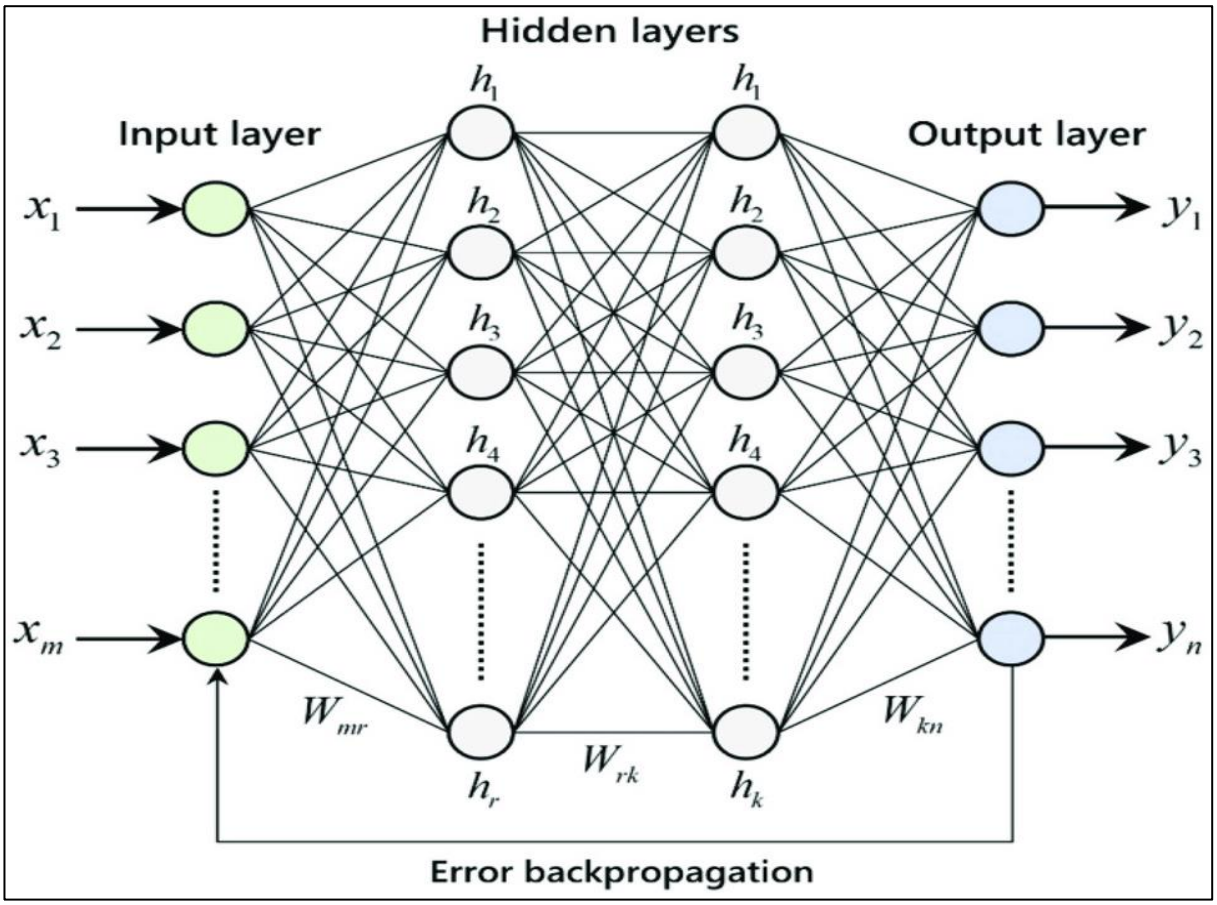

Figure 2.18: A structure of artificial neural network [67]

2.6.5 Particle Swarm Optimization (PSO)

PSO algorithm was created by (Kennedy and Eberhart 1995). PSO was inspired by the behavior of birds and schooling fish. PSO is a strategy that is based on the relationship between the particles in a swarm where there is a stochastic optimization and the population in the search space. PSO keeps a population of solutions in the search space that are iteratively updated. Every particle has a velocity and a location. The velocity of a particle determines how quickly moves across the search space, whereas the position of a particle indicates the answer to the problem being investigated. Each iteration's computations are based on two previously discovered positions: the personal best location and the best place obtained in a neighborhood. Particle's velocity and location are calculated using mathematical formulas [69, 70]:

$$v_{i(t+1)} = w * v_{i(t)} + c1 * r1 * \left(pbest_i - x_{i(t)}\right) + c2 * r2 * \left(gbest - x_{i(t)}\right) \quad (2.13)$$

$$x\_i(t+1) = x\_i(t) + v\_i(t+1) \quad (2.14)$$

Where $v\_i(t+1)$ is the velocity of particle i at time t, w is the inertia weight, which controls how much the particle's previous velocity influences particle's new velocity, $c1$ and $c2$ are the cognitive and social coefficients, respectively, which control how much the particle is attracted to particle's personal best position and the global best position of the swarm, $r1$ and $r2$ are random numbers between 0 and 1, $pbest\_i$ is the personal best position of particle i and $gbest$ is the global best position of the swarm.

The particles are graded according to the quality of the solution they generate. The optimal particle placements in minimization problems are those that provide the least fitness based on the function used. In the case of a maximization problem, the reverse is true. The neighbourhood represents how information is exchanged among particles; in a completely linked neighborhood, every particle has access to all accessible information, whereas other network topologies restrict the amount of information available to each particle [69, 70]. The representation of PSO is demonstrated in Figure 2.19.

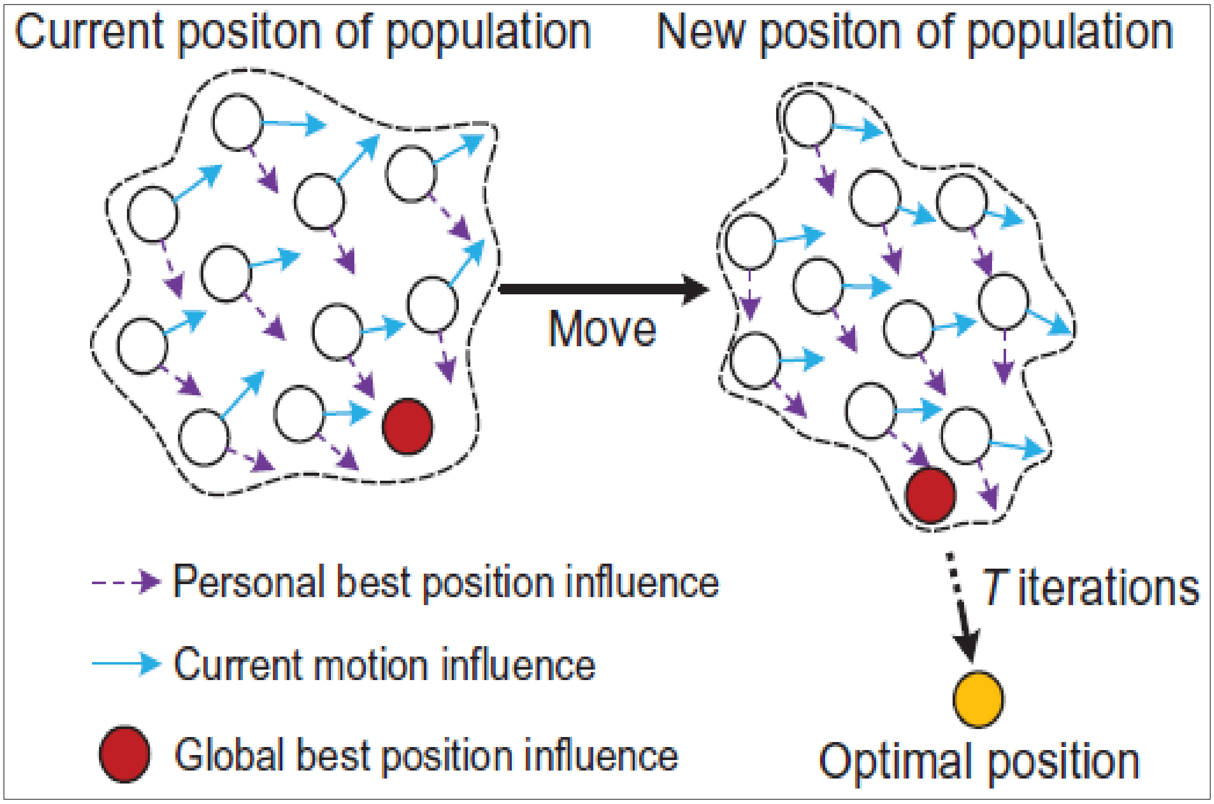

Figure 2.19: The representation of particle swarm optimization [71]

2.7 Transfer Learning (TL)

TL is a technique that leverages knowledge gained from solving one task and applies it to another related task. TL addresses the challenge of limited labeled data in the target domain by utilizing pre-trained models trained on large-scale datasets from a source domain [13].

2.7.1 The main concept

While ML technology has reached great success and has been successfully applied in a wide range of practical applications, ML still has some limitations for certain real-world scenarios. The ideal ML scenario has many labeled training instances with the same distribution as the test data. However, gathering enough training data is frequently costly, time-consuming, or even unrealistic in many scenarios. TL, which focuses on knowledge transfer across domains, is a promising ML methodology for addressing the problem [72]. Traditional ML process a single dataset at once, while TL can handle two datasets at the same time by fallows three basic steps:

Step 1: Data set 1 with large amount of data/labels handled by machine learning model 1,

Step 2: Knowledge transfer learning, finally

Step 3: Benefits from the knowledge of model 1 and handled by model 2.

Figure 2.20 shows the difference between ML and TL [73].

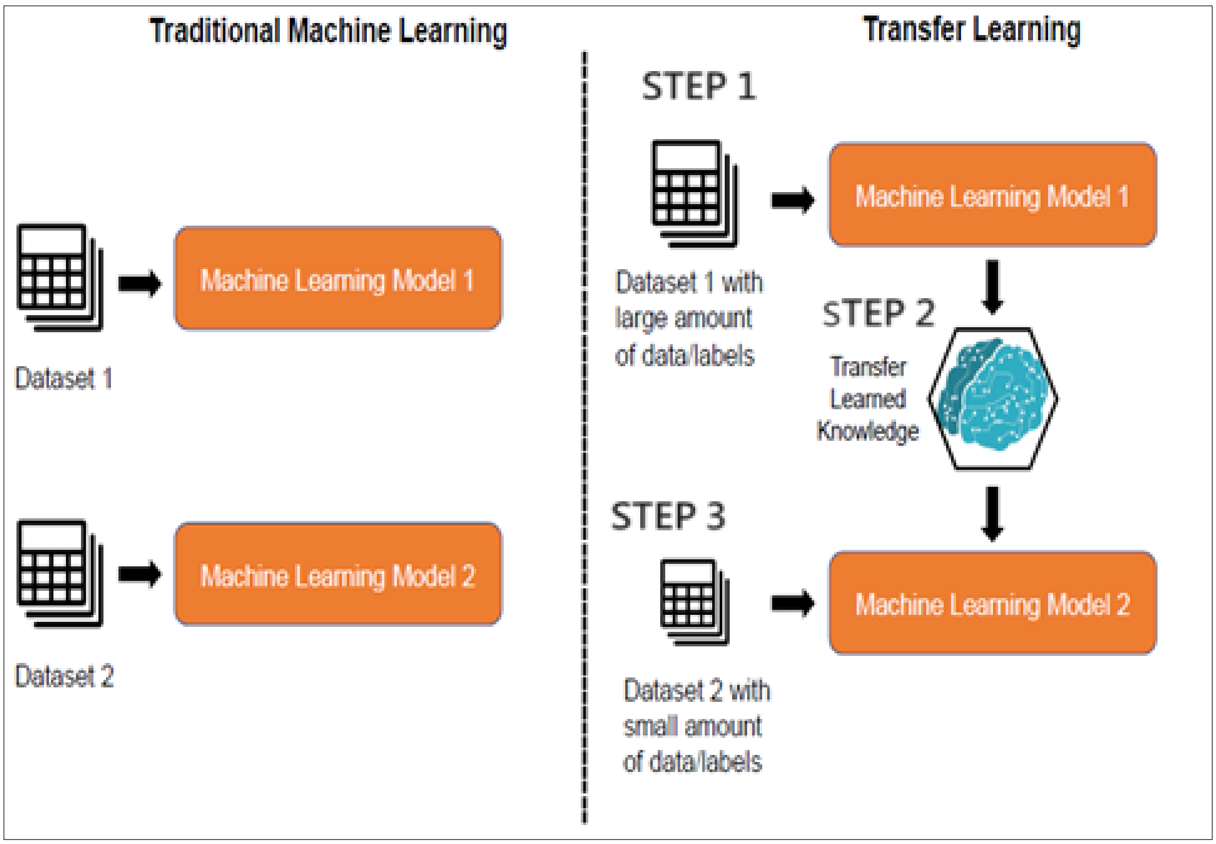

Figure 2.20: Difference between machine learning and transfer learning [73]

By transferring knowledge from the source domain to the target domain, TL enables effective learning even when faced with scarce data. The pre-trained models capture general features and patterns that are applicable across domains, allowing the model to generalize better and achieve higher performance in the target task. TL has been successfully applied in various domains, including computer vision, natural language processing, and speech recognition, and has demonstrated significant improvements in model performance and efficiency [13]. For more clarity ; the data set 1 called source domain data set that is training by a model; this process followed by saving the model by keeping all its parameters to initialize a new model which is used to train data set 2 (target domain dataset) as explained in Figure 2.21 [26].

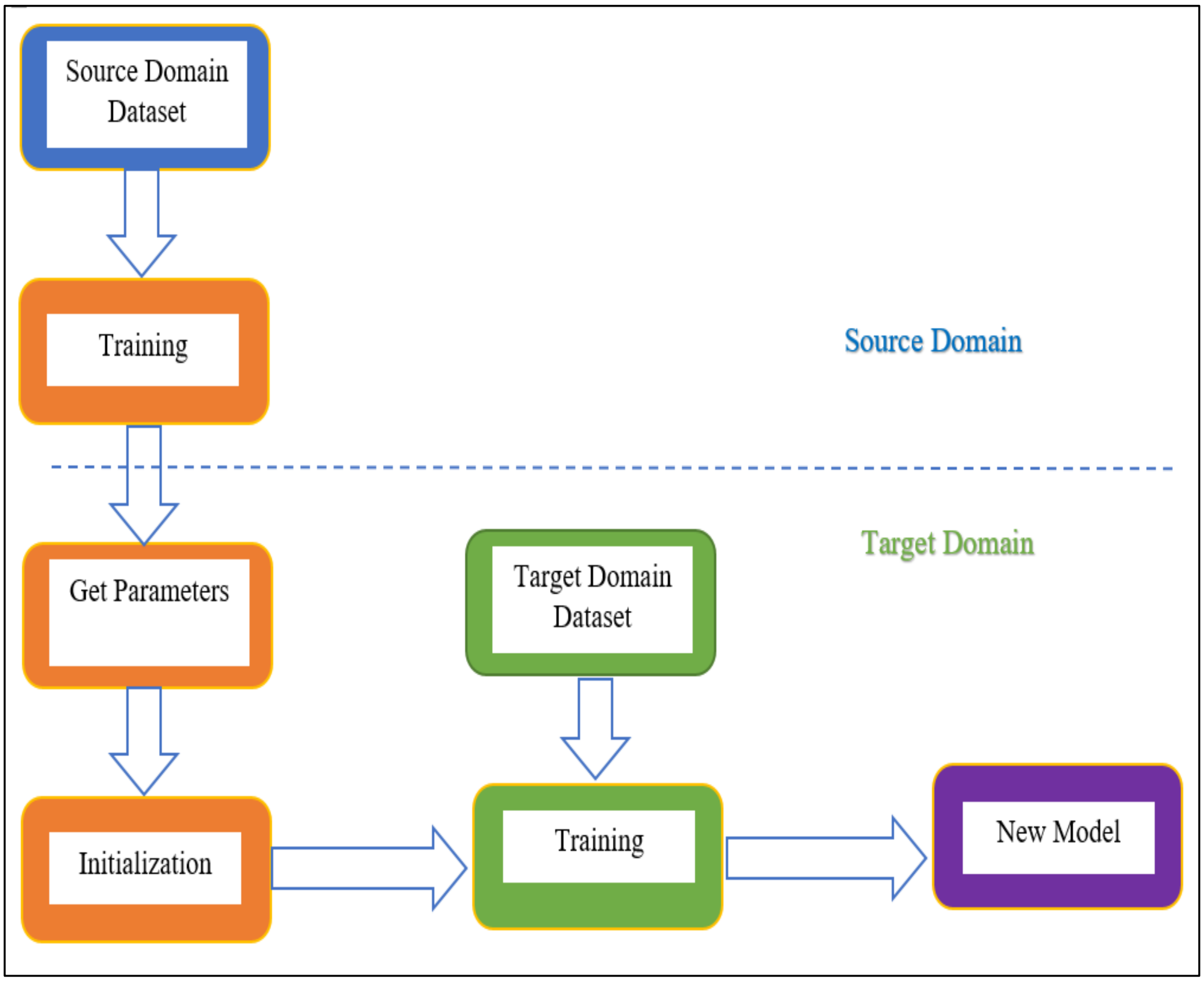

Figure 2.21: The procedure of transfer learning [26]

2.7.2 Transfer learning advantages

TL offers several advantages [73]:

1) TL addresses the challenge of acquiring sufficient and high-quality training data, a common hurdle for conventional ML methods. TL resolves this issue by leveraging knowledge from similar domains that

possess ample high-quality data. Consequently, TL emerges as a promising solution for ML-based wireless networks in the foreseeable future.

2) TL accelerates the learning process by capitalizing on valuable knowledge shared from analogous domains or acquired in previous tasks, eliminating the need to start from scratch. This significant reduction in training time enhances the learning rate, which is particularly crucial for developing ultra-low latency applications intended for future wireless networks.
3) TL minimizes computing requirements, unlike conventional ML approaches that necessitate substantial computational resources. Since TL exploits pre-trained models from source domains before transferring them to the target domain, TL significantly alleviates the computing demands of the training process. This aspect proves advantageous for wireless devices like smartphones and edge devices, which typically have hardware limitations.
4) TL mitigates communication overhead in wireless networks. Instead of transmitting large-sized raw data, TL only requires the transfer of knowledge, such as trained model weights. As a result, the communication overhead experienced in wireless networks can be significantly reduced.
5) TL ensures data privacy by avoiding the need to learn from raw data from other domains. Instead, the focus is on learning from pre-trained models expressed through their weights. This safeguarding of data privacy makes TL particularly valuable for privacy-sensitive wireless applications, including healthcare and military communication networks.

2.7.3 Transfer learning with tabular dataTL with tabular data involves utilizing pre-trained models on large-scale datasets from related domains to improve the performance of classification or regression tasks on new tabular datasets with limited labeled data. By leveraging the knowledge learned from the pre-training phase, the pre-trained models can capture useful patterns, feature representations, and relationships that can be beneficial for the target task. There are many advantages of TL g with tabular data: it allows for effective training even when the labeled data for the target task is scarce or insufficient. By utilizing pre-trained models, the need for a large labeled dataset is reduced. Pre-trained models capture general patterns and relationships from large-scale datasets, enabling better generalization to new tabular datasets. This helps in improving the model's performance on unseen data. Faster Training: TL with pre-trained models speeds up the training process. The pre-trained model's initial weights provide a good starting point, allowing for faster convergence during fine-tuning on the target task [13, 74].

2.8 Evaluation Measures

The performance of ML and DL models can be evaluated using various measurements. Also, there are 5G network performance evaluation Measurements. The following metrics are used in the evaluation stage to ensure the effectiveness and quality of the system.

2.8.1 Confusion matrix

Confusion **M**atrix (CM) is one of the most important methods. A CM is a table that shows the true and predicted classifications of a model. CM is divided into four quadrants [75]:

1) **True Positive (TP):** The number of instances that were correctly classified as positive.

2) **False Positive (FP):** The number of instances that were incorrectly classified as positive.
3) **True Negative (TN):** The number of instances that were correctly classified as negative.
4) **False Negative (FN):** The number of instances that were incorrectly classified as negative.

A. The Accuracy metric

The accuracy metric measures the overall accuracy of the model. The accuracy metric calculated as follows [76]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2.15}$$

B. The Precision metric

The precision metric measures the proportion of true positives out of all the positive predictions made by the model. The precision metric calculated as follows [76]:

$$precision = \frac{TP}{(TP + FP)} \tag{2.16}$$

C. The Recall metric

The recall metric measures the proportion of true positives out of all the actual positives. It is calculated as follows [76]:

$$Recall = \frac{TP}{(TP + FN)} \tag{2.17}$$

D. The F1 score metric

The F1 score is a weighted average of precision and recall. It is calculated as follows [76]:

$$F1\ score = \frac{2 * (precision * recall)}{(precision + recall)} \tag{2.18}$$

2.8.2 5G network performance evaluation measurements

The most important measures deal with network slicing is latency and reliability. Both are crucial for providing a reliable user experience in 5G networks, especially considering the diverse range of applications and services that depend on on low latency and high reliability for optimal performance [77].

A. Latency: often measured as Round-Trip Time (RTT), represents the time it takes for a signal or data packet to travel from the source device to the destination and back to the source. In 5G networks, the lower the latency, the more responsive the network is perceived to be by users. The latency metric is often expressed as [77].

$$latency = RTT \tag{2.19}$$

B. Reliability in the framework of 5G network performance refers to the ability of the network to establish and maintain successful connections consistently. The reliability metric is often expressed as [77].

$$Reliability\ (\%) = \frac{Number\ of\ Successful\ Connections}{Total\ Attempts} * 100 \tag{2.20}$$

Chapter Three

The Methodology

Chapter Three

The Methodology

3.1 Introduction

This chapter explains the details of the proposed system of an improved Network Slicing classification in 5G network based on DL approach. This chapter also introduces the description of many used techniques and proposes and designs many algorithms to be applied for each stage of proposed model. In pervious chapters, the importance of the classification mechanisms for Network Slicing in 5G network has been presented, which is becoming increasingly significant for effective resources utilization and network management. However, the availability of labeled data for training accurate classification models is frequently lacked or limited; making reliable Network slicing classification difficult. Therefore, this chapter introduces a proposed method for generating a relevant data attributes highly associated with network slicing. Moreover, to classify the network slices. Three Models have been proposed two of them for to realizing the Network slicing classification and one of them for scarce network slicing data. The following sections includes an exhaustive description of the proposed algorithms, models, and formulas.

3.2 The Proposed Model

The general overview of the proposed model is shown in Figure 3.1. The proposed model is implemented by the 5G core network. In this dissertation, two cases are presented, the first case contains a large data for users, while the second case contains scarce data. In both cases the main purpose is to classify the network slice and this is done by the 5G core network which will choose the optimal network slice; that satisfy the demands of the users of the network.

In first case, two models are suggested. **Model 1** is model consists of five main stages (Data generation, Data normalization, level 1: TabNet, RF and DT, level 2: Hybrid classifier and Model Evaluation), see Figure 3.1.

While **Model 2** model consists of five main phases (Data Generation, Data Normalization, level 1: DNN, RF and DT, Level 2: Enhanced TabNet and Model Evaluation). While in second case the situation where data is scarce is addressed by relying on the principal of TL. This case involves fine-tuning a pre-trained TabNet model using limited labeled data specific to 5G service classification.

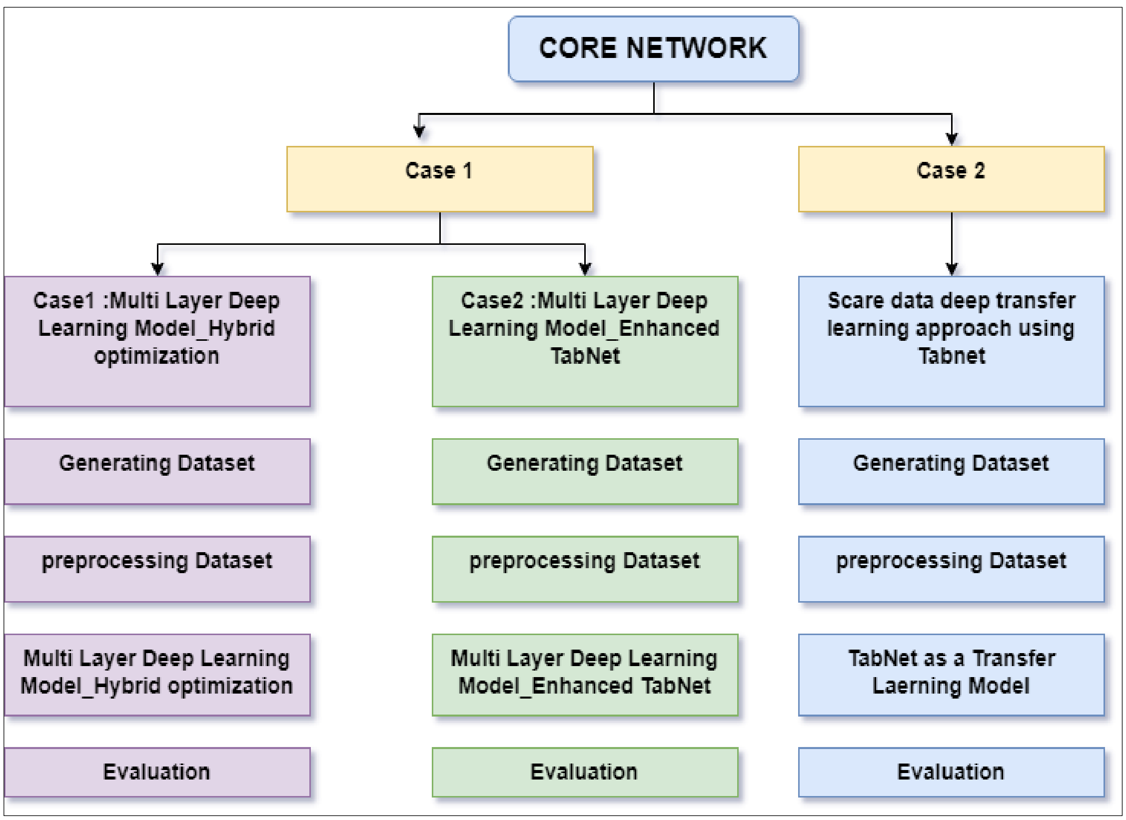

Figure 3.1: A general overview of the proposed model

3.2.1 Generating 5G network slicing dataset

In all models that were previously explained, the data generation step and the data processing step are the same, so they will be explained once. As explained and clarified in Chapter 2, Paragraph **2.5**; For Case1 a dataset in [19] which consists of (165 user demand) is used, also generating a data set that consist of (3000 user demands) as explained in Table 3.1.

Table 3.1: Case 1 5G network slicing dataset

No.	Classes	No. of instances for the dataset in [19]	No. of instances for the dataset in Network 1
1	UHD Video Streaming	22	150
2	Immersive Experience	22	450
3	Smart Grid	21	255
4	e-Health	21	84
5	ITS	19	720
6	Vo5G	19	375
7	Connected Vehicles	14	510
8	Industry Automation	14	264
9	Video Surveillance	13	192
	The total no. of instances	165	3000

Algorithm 3.1 illustrates the method for generating 5G Network slicing Dataset. Any number of slices can be generated.

Algorithm 3.1: Generating 5G network slicing dataset

Input: sources ($\mathbf{n\_slice_1}, \mathbf{n\_slice_2} \ldots . \mathbf{n\_slice_m}$), where **n_slice** is no. of network slices types, **m** is no. of network slices.

Output: Generating_5G_network_slising_dataset

Encoding_F.={($\mathbf{n\_slice_1}$, $\text{feature}_1, \text{feature}_2 \ldots . \text{feature}_k$), ($\mathbf{n\_slice_2}$, $\text{feature}_1, \text{feature}_2 \ldots . \text{feature}_k$), ...} , where **f** is no. of features types, **k** is no. of features.

Begin

1. sources ← a vector representing the input quantities of each network slices.

2. Create an empty set network: network= { }

For each $\mathbf{n\_slice_i}$ in **Sources :**

//**Add** $\mathbf{n\_slice_i}$ **names to the set network based on input quantities.**

network ← network ∪ {$\mathbf{n\_slice_i}$}

End For

// **Define Feature_List as a set of features**($\mathbf{f_1, f_2 \ldots . f_k}$)**:**

3. **Feature_List** ← {Latency, Jitter, BitRate,PacketLossRate,PeakDataRateDL,PeakDataRateUL,Mobility,ServiceReliability,Availability,SurvivalTime,ExperienceDataRateDL,ExperienceDataRateUL,InterruptionTime}
4. For each $\mathbf{n\_slice_i}$ in **network:**
5. For each $\mathbf{feature_j}$ in **Features_List:**

// **Generate random values for** $\mathbf{feature_j}$ **between (min_**$\mathbf{feature_j}$ **and max_**$\mathbf{feature_j}$) **that explained in Table (2.4).**

6. $\mathbf{Value_{ij}}$ ← value of $\mathbf{feature_j}$ in $\mathbf{n\_slice_i}$
7. **End For**
8. **End For**
9. Write the feature data $\mathbf{Value_{ij}}$ for each $\mathbf{n\_slice_i}$ along with corresponding feature names into a file **F** row by row.

// Shuffle the rows of the file to randomize the dataset.

10.Shuffle_F← Shuffle (F)

//Perform label encoding on the class column by converting Network slicing names into numerical labels.

11.Encoding_F ← Label_encoding (**Shuffle_F**)

//Save the resulting dataset to a new file.

12. Generating_5G_network_slising_dataset ← **Encoding_F**.

End

3.2.2 Data preprocessing

Algorithm 3.2 explained the data preprocessing step by utilizing Z-score scaling (normalization) technique. This scaling technique is a common technique in statistics; This process done by subtracting the mean from the variable and then dividing by the standard deviation, based on **Equation 2.1.**

Algorithm 3.2: Data preprocessing

Input: Generating_5G_network_slising_dataset with **S** samples and **F** features.

Output: $Norm\_5G$ represents the normalized dataset after the process.

Begin

1. X ← Generating_5G_network_slicing_dataset, where $\boldsymbol{x_{ij}}$ denotes the value of feature **j** in sample **i**.

//Calculate Mean of each feature across all samples:

2. For each $\mathbf{f}_j$ from 1 to F:

3. Calculate $Mean[j] \leftarrow \frac{1}{S}\sum_{i=1}^{S} \boldsymbol{x_{ij}}$

4. **End For**

//Calculate Standard Deviation of each feature across all samples:

5. For each $\mathbf{f}_j$ from 1 to F:

6. Calculate $StdDev\,[j] \leftarrow \frac{\sum_{i=1}^{S}(\boldsymbol{x_{ij}}-Mean[j])^2}{S-1}$

7. **End For**

8. For each $\mathbf{f}_i$ from 1 to F:

9. For each sample j from 1 to S:

// Normalize the value of feature i in sample j using equation (2.1):

10. X_norm[i][j] ← (($\boldsymbol{x_{ij}}$ - Mean[j]) / StdDev[j])

13. End For

14. End For

// Save the resulting normalized dataset to a new file.

11. $Norm\_5G$ ← X_norm

12.End

3.2.3 Case 1 network slicing classification models

A 5G network slicing dataset, containing 3000 users, is employed in case 1, where the network slices are classified through the application of two models:

1) Model 1

Model 1 proposed design consist of two levels as shown in **Figure 3.2**. Level 1 consist of DT, RF and TabNet (as explained in sections 2.6.1, 2.6.2 and 2.6.3); While level 2 consist of hybrid classifier (PSO-ANN).

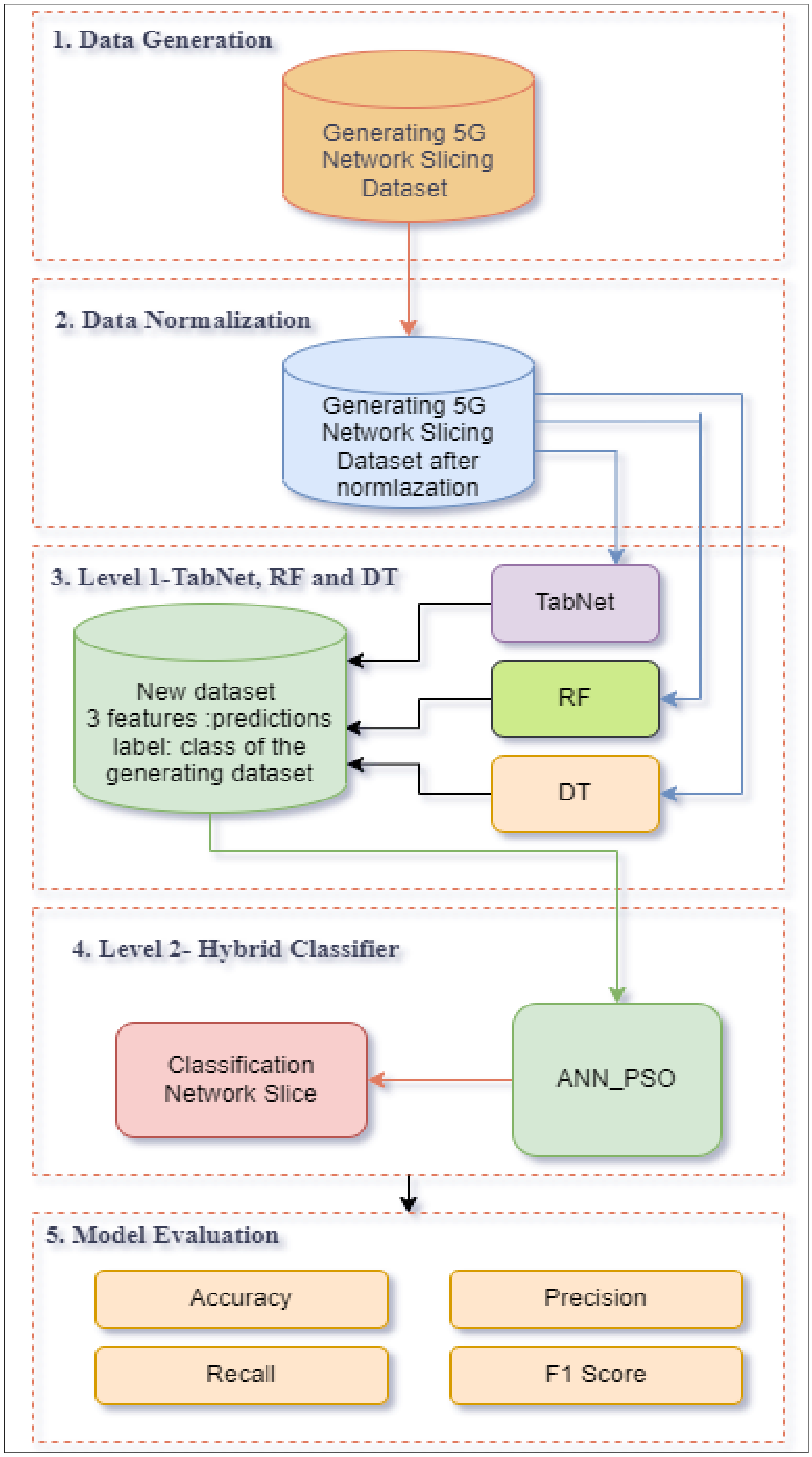

Figure 3.2: 5G network slicing classification model 1

In this dissertation, a hybrid classifier was implemented where PSO worked instead of backpropagation algorithm for choosing optimal weights as shown in **Figure 3.3.**

Backpropagation, like every gradient descent method, employs the hill-climbing technique to function minimization, this characteristic makes it prone to early convergence towards a local minimum. Alternative training techniques, such as the PSO algorithm, can be employed to solve the problem of early convergence ANN training is an optimization problem in which the goal is to find the best set of weights and biases to minimize the ANN error [70].

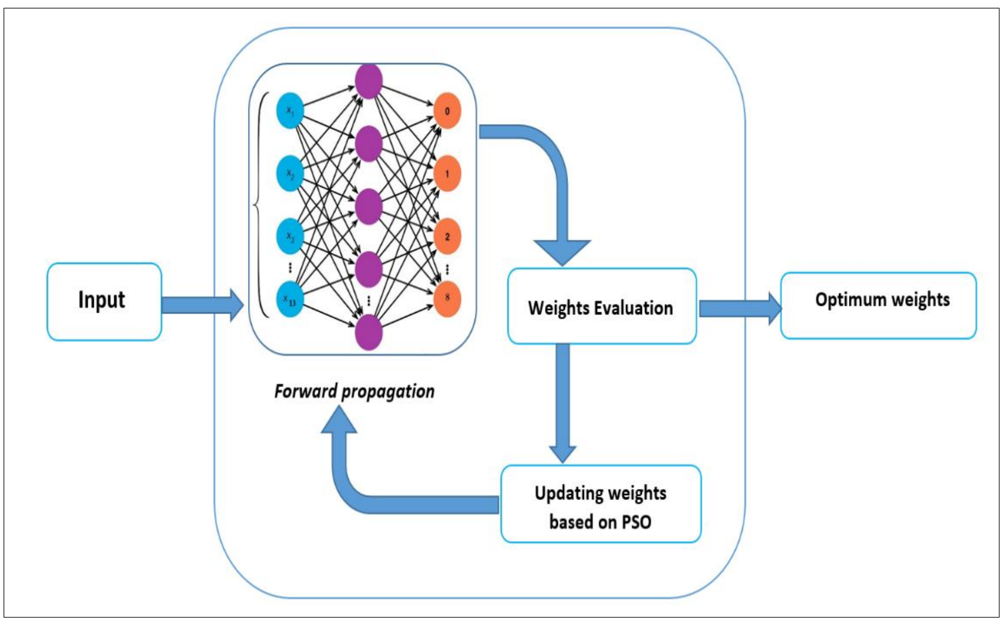

Figure 3.3: The process for choosing optimal weights using particle swarm optimization

The approach employs a PSO technique to iteratively improve candidate weights. It utilizes equation below to determine the weight inertia ω:

$$\omega = \frac{2 + \varepsilon}{2} \tag{3.1}$$

Where ε is a random value between 0 and 1.

The **M**ean **S**quare **E**rror (MSE), represent a cost function in this approach; it's used for evaluate the performance of the process as explained in Equation 3.2:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(E_i)^2 \tag{3.2}$$

Where E_i is different between i actual output and i desired output.

Following the computation of the fitness of a candidate weight, the process proceeds to update the velocity that will be included into the weights, resulting in the generation of a new candidate weight. The weight matrix w undergoes regular updates, until the process satisfies the termination conditions. The population size of PSO is dynamically adjusted to align with the dimensions of the weight's matrix of the ANN.

After generating a dataset and normalizing it, the data entered to a model with two levels as explained in **Algorithm 3.3**; **Algorithms 3.4, 3.5, 3.6 and 3.7** explain respectively the models of the first level of our model. After each model in first level make its prediction, each prediction represents a new feature of a new dataset which in turn is an input to the second level.

Algorithm 3.3: network slicing classification Case1- Model 1

Input: $Norm\_5G$ with **F** features and **L** labels.

Output $case1\_model1$ represents the predicted dataset.

Begin

1. X_data ← Norm_5G

//Define Level 1 models

2. Models = [Decision Tree Classifier (explained in **Algorithm (3.4)**) , Random Forest Classifier(explained in **Algorithm (3.5)**, TabNet Classifier(explained in **Algorithm (3.6)**]

//Initialize an array for predictions of Level 1 models

3. Predictions_Level1 ← [ ]

4. For each Model in Models (1,2,3):

// Train the model

// Make predictions on the entire dataset (X_data)

5. Prediction_Model ← Model.predict(X_data)

// **Append the predictions to the Level 1 predictions array**

6. Predictions_Level1.append(Prediction_Model)

7. End For

// Create a new dataset using the predictions of Level 1 models

8. New_Dataset ← Transpose(Predictions_Level1) // Each prediction is a new feature

// Define the level 2 PSO-DNN hybrid Classifier Hybrid classifier model

9. Hybrid_Model ← **PSO-ANN** HybridClassifier (explained in **Algorithm (3.7)**)

// train the Hybrid classifier to make predictions on the new dataset

10. Predicted_Labels ← Hybrid_Model.predict(New_Dataset)

// Evaluate the model using Confusion Matrix (CM) : Explained in **Equations (2.15) to (2.18)**

11. CM ← Calculate_Confusion_Matrix(True_Labels, Predicted_Labels)

12. Evaluate the model using network performance measurement Explained in Equations (2.19) and (2.20).

// Return the predicted output labels and evaluation metrics

13. Return Predicted_Labels, CM

End

Algorithm 3.4 illustrates a DT Classifier for 5G Network Slicing Dataset where the Input tis 5G Network slicing Dataset with labeled classes (features and output labels). While the output is Predicted output labels using the DT Classifier.

Algorithm 3.4: A Decision Tree Classifier for 5G Network Slicing Dataset

Input: $Norm\_5G$ with **F** features and **L** labels.

Output: $Predicted\_output$ labels using the Decision Tree Classifier.

Begin

1. X ← Norm_5G.

   //Define the dataset as X_data (input features) and Y_data (output labels)

2. X_data← Features from the dataset
3. Y_data ← Output labels from the dataset

   // Split the dataset into training and testing sets indicating an 70-30 split

4. X_train, Y_train ← 70% of the data will be used for training
5. X_test, Y_test ← 30% for testing.

   // Create a Decision Tree Classifier with a specified depth

6. DecisionTree_Model ← Create_DecisionTreeClassifier(max_depth)

   // Fit the model using the training data

7. DecisionTree_Model ← fit(X_train, Y_train)

   // Make Predictions using the trained model on the entire dataset

8. DecisionTree_Model ← predict(X_data)
9. Predicted_Labels ← DecisionTree_Model

   // Save Predictions to a file for further analysis

10. **Predictions_DecisionTree**← Predicted_Labels

End

In addition, **Algorithm 3.5** illustrates a RF Classifier for 5G Network Slicing Dataset where the Input is 5G Network slicing dataset with labeled classes (features and output labels). While the output is Predicted output labels using the RF Classifier.

Algorithm 3.5: A Random Forest Classifier for 5G Network Slicing Dataset

Input: $Norm\_5G$ with **F** features and **L** labels.

Output: $Predicted\_output$ labels using the **Random Forest Classifier**.

Begin

1. X ←Norm_5G.

//Define the dataset as X_data (input features) and Y_data (output labels)

2. X_data← Features from the dataset

3. Y_data ← Output labels from the dataset

// Split the dataset into training and testing sets indicating an 70-30 split

4. X_train, Y_train ← 70% of the data will be used for training

5. X_test, Y_test ← 30% for testing.

// Create a Random forest Classifier with a specified depth

6. Random forest _Model ← Create_Random forest Classifier(max_depth)

// Fit the model using the training data

7. Random forest _Model ← fit(X_train, Y_train)

// Make Predictions using the trained model on the entire dataset

8. Random forest _Model ← predict(X_data)

9. Predicted_Labels ← Random forest _Model

// Save Predictions to a file for further analysis

10. Predictions_ Random forest ← Predicted_Labels

End

Furthermore, **Algorithm 3.6** illustrates a TabNet Classifier for 5G Network Slicing dataset where the Input is 5G Network slicing Dataset with labeled classes (features and output labels). While the output is Predicted output labels using the TabNet Classifier.

Algorithm 3.6: A TabNet Classifier for 5G Network Slicing Dataset

Input: $Norm\_5G$ with **F** features and **L** labels.

Output: *Predicted_output* labels using the **TabNet Classifier**.

Begin

1. X ← Norm_5G.

//Define the dataset as X_data (input features) and Y_data (output labels)

2. X_data← Features from the dataset

3. Y_data ← Output labels from the dataset

// Split the dataset into training and testing sets indicating an 70-30 split

4. X_train, Y_train ← 70% of the data will be used for training

5. X_test, Y_test ← 30% for testing.

// Train a TabNet Classifier:

6. - Initialize the model with hyper-parameters:

7. - Number of decision blocks (B), attention features (D), gamma value (γ).

8. - Apply input features to batch normalization layer.

// **Feature Transformer Layer:**

9. - Calculate common features (C_i) using Fully Connected Layer 1.

10. $C_i \leftarrow FC1$

11. - Calculate separate features (S_i) using Fully Connected Layer 2.

12. $S_i \leftarrow FC2$

13. - Combine outputs using GLU activation function as explained in Equations (2.2), (2.3) and (2.4):

14. $f_i \leftarrow \mathrm{GLU}(C_i + S_i) * \sqrt{0.5}$

// **Split Layer:**

15. - Apply mask M to input features.

16. - Divide masked features M_f into two parts:

17. - Part 1 $Final\_out$ for calculating the final output of the model.

18. – Part2 M_next for calculating the Mask layer for the next step.

19. For steps = 1 to s:

//Attentive Transformer Layer:

20. - Generate new feature representation using Fully Connected Layer

21. $A_i \leftarrow FC$

22. - Pass output to Batch Normalization layer:

23. $BN \leftarrow FC$

24. - Apply attention using Sparsemax activation function for prior scale term at the (i - 1)th step multiply by trainable function

25. to the processed features from the (i - 1) step as explained in Equations (2.5), (2.6) and (2.7):

26. $M[i] \leftarrow Sparsemax(P[i-1] * h\_i(a[i-1]))$

27. - Indicate important features for the current step using Mask layer.

28. - Repeat Feature Transformer Layer.

29. - Repeat Split Layer.

30. - Pass output to ReLU Layer:

31. - Apply ReLU activation function to features explained in equation (2.8) .

32. $ReLU(\text{M\_}f\ ) \leftarrow max(0, \text{M\_}f\ )$

33. End for

34.- Sum the output vectors from each step.

//Make Predictions: Use the trained TabNet model to make predictions on the entire dataset (X_data).

35. Predicted_Labels ← Random forest _Model

//Save Predictions to a file for further analysis.

36. Predictions_ TabNet ← Predicted_Labels

End

Formally, the dataset was submitted for the first, second, and third models, and the prediction results were stored in a new dataset, which will attend as input for the proposed model's second level (the hybrid classifier level), as explained in chapter 2 paragraph 2.5.6. **Algorithm (3.7)** represents the hybrid classifier.

Algorithm 3.7: PSO_ANN_Hybrid_Classifier

Begin

Input: New Dataset from level 1 with **F** features (p1, p2, p3) and **L** labels**.**

Output: *Predicted_output* labels using the **PSO_ANN Classifier**.

// Load and Preprocess the Data:

1. **-** X← New dataset.

2. - Split the dataset into input features (X_data) and output labels (y_data).

// Define the Forward Propagation Function:

3. - Implement forward propagation function for the neural network:

4. - Function forward_prop (weights, biases, X):

5. **-** Calculate predicted outputs using weights (W) and biases (b) explained in equations (2.9) and (2.10):

6. $output \leftarrow f\left(\sum(wi\ *\ xi)\right)$

// Define the Objective Function for PSO: //Modified Steps

7. **-** Create an objective function f(x) that computes the loss:

8. - Function objective_function(candidate_position):

9. - Use forward_prop to compute loss (cross-entropy) for candidate_position explained in equations (2.11):

$$error \leftarrow \left(target_{output} - actual_{output}\right) * act.\_fun.\_der.$$

//Perform PSO Optimization:

10. **-** Define dimensions of the search space based on ANN architecture.

11. - Initialize PSO optimizer with particles P, dimensions D, and options.

12. - Run PSO optimization for a specified number of iterations:

//Initialization:

13. - Set population size for candidate weights.

14. - Initialize inertia weight ω **(Equation 3.1)**.

15. - Initialize constants for c1 and c2 (e.g., c1 = 1.5, c2 = 2.0), which represent the cognitive
and social learning coefficients respectively.

16. -The initial velocity for each particle (v_i(t)) can be set to zero at the beginning of the optimization process.

17. - Randomly initialize candidate weights for each particle $x\_i(t)$.

//Evaluation:

18. - Compute fitness of candidate weights using objective_function.

19. - Update particle velocities explained in equation (2.13):

20. $v\_i(t+1) \leftarrow w * v\_i(t) + c1 * r1 * (pbest\_i - x\_i(t)) + c2 * r2 * (gbest - x\_i(t))$

21. - Update positions of particles based on velocities explained in equation (2.14):

22. $x\_i(t+1) \leftarrow x\_i(t) + v\_i(t+1)$

23. -Calculate biases based on fitness error.

//**Weight updating and reconstructing:**

24. - Redistribute weights generated by PSO to form weight matrix

25. - Modify biases based on error term.

26. - Reconstruct the weight matrix for forward propagation.

//**Termination criteria:**

27. - If termination conditions are met, proceed to step 6.

28. - Else go back to step 2.

//**Output:**

29. - Return the optimum weight matrix obtained from the training process.

// **Perform Predictions:**

30. - Function predict (X_test, optimized_weights):

31. - Use trained weights to predict classes on test data.

32. - Calculate accuracy of the ANN on the test set.

// **Output:**

33. - Return trained ANN weights and biases.

34. - Provide evaluation measurements of the ANN model on the test data.

End

2) Model 2

Model 2 proposed design as shown in **Figure 3.4** consist of two levels. Level 1 consist of DT, RF and ANN (explained in sections 2.6.1, 2.6.2 and 2.6.4); While level 2 consist of enhanced classifier (enhanced TabNet)**.**

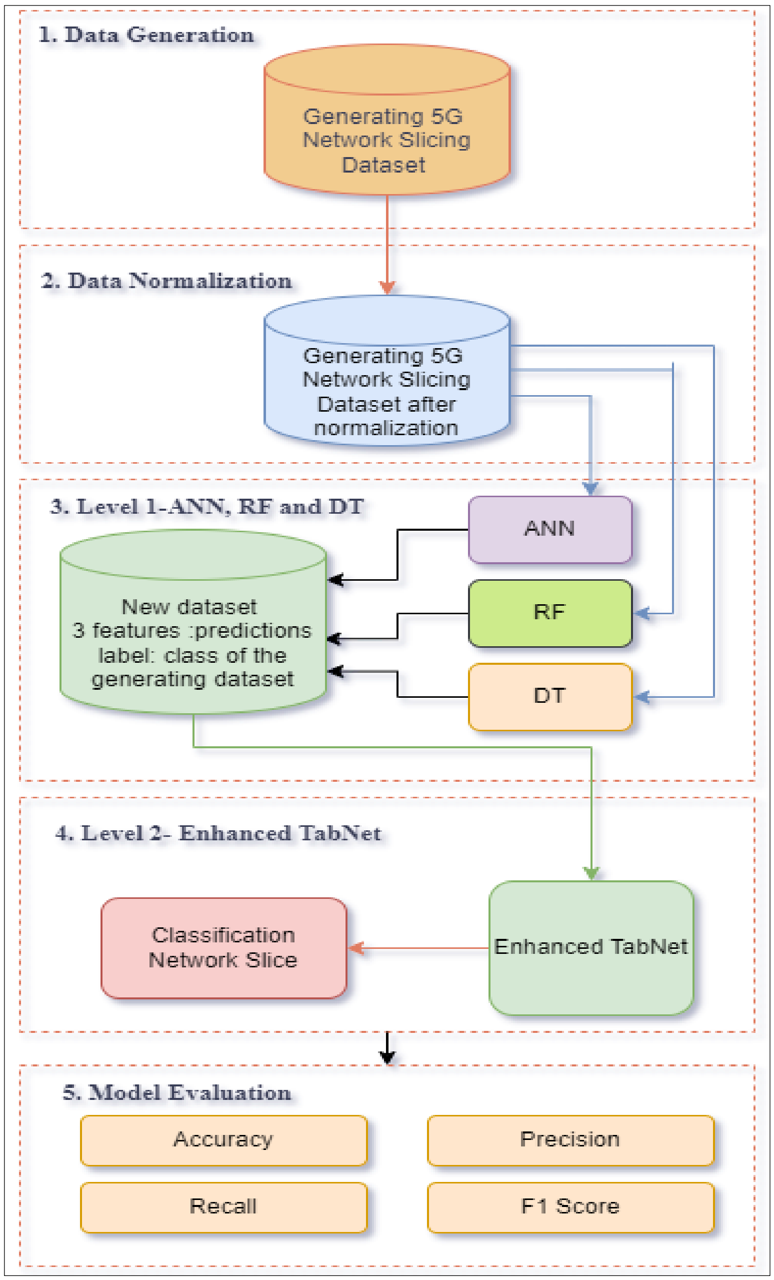

Figure 3.4: 5G network slicing classification model 2

In this dissertation, an enhancement was implemented at TabNet model in two places: at feature transformer block, utilizing a different activation function called Self-Gated Activation Function (swish) in place of GLU. A switch activation function was introduced by researchers in [78] in 2017; their experiments shows that this activation function outperforms some other common functions.

Swish mixes the identity function′s linearity with the sigmoid functions' nonlinearity. This combination of features enables the activation to transition seamlessly between linear and nonlinear behavior. In addition, the effectiveness of an activation function can vary across different layers to better optimization and generalization [79]. The Swish activation function is defined as [78]:

$$Swish(x) = x * Sigmoid\ (beta * x) \tag{3.3}$$

Figure 3.5 shows the structure of feature transformer block after enhancement.

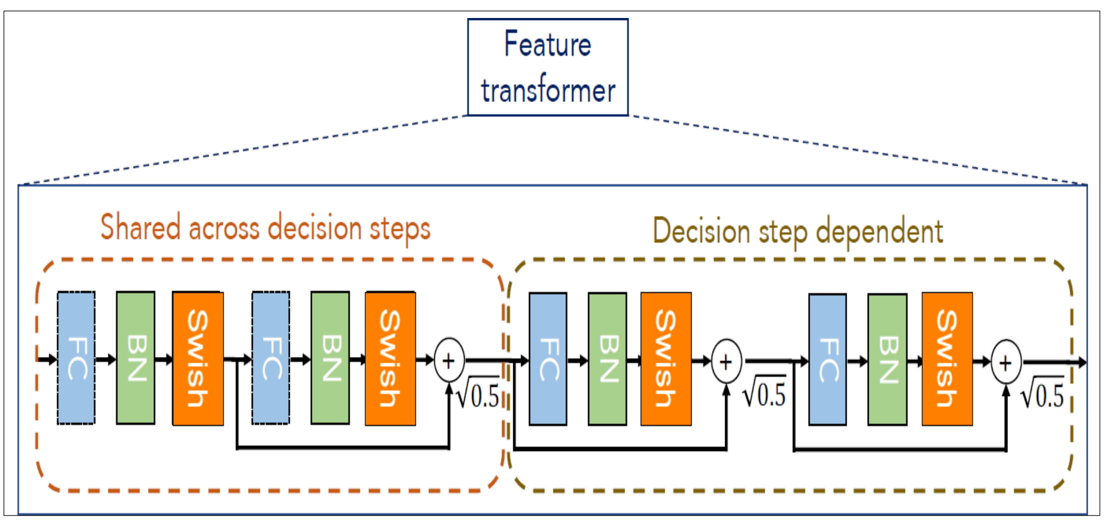

Figure 3.5: The structure of attentive transformer block

Another enhancement was implemented in this dissertation by using **E**xponential **L**inear **U**nit activation function (ELU) instead of ReLU. ELU proposed by researchers in [80] their experiments shows that this activation function speeds up learning and leads to higher classification accuracy. And it outperforms some other common activation functions.

In contrast to ReLU, which has an abrupt kink at 0, ELU is continuous everywhere, even zero. This smoothness can improve optimization stability, resulting in faster training convergence.

In terms of training time, convergence, and generalization across several deep learning tasks, ELU outperforms ReLU [81]. The ELU activation function is defined as [80]:

$$ELU(x) = \begin{cases} x & if\ x \geq 0 \\ \alpha * (e^x - 1) & if\ x < 0 \end{cases} \qquad (3.4)$$

In Equation 3.4, x is the input to the function and α is a hyper-parameter controlling the slope. **Figure 3.6** shows the structure of TabNet model after enhancement.

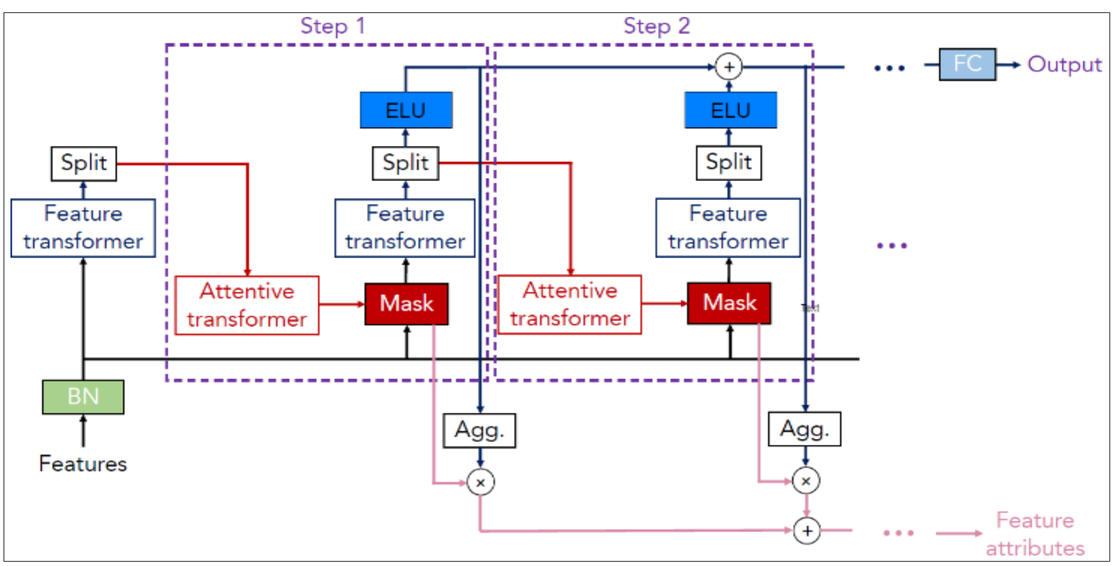

Figure 3.6: The structure of TabNet model after enhancement

After generating and normalizing a dataset, the data entered to a model with two levels as explained in **Algorithm 3.8. Algorithms 3.4, 3.5, and 3.9** explain respectively the models of the first level.

Algorithm 3.8: network slicing classification Model 2 -Case1

Input: $Norm\_5G$ with **F** features and **L** labels.

Output: $Case1\_Model2$ represents the predicted dataset.

Begin

1. X ←Norm_5G

//Define Level 1 models

2. Models = [Decision Tree Classifier (explained in **Algorithm (3.4)**) , Random Forest Classifier(explained in **Algorithm 3.5**, ANN Classifier(explained in **Algorithm 3.9**]

//Initialize an array for predictions of Level 1 models

3. Predictions_Level1 ← [ ]

4. For each Model in Models:

// Train the model

// Make predictions on the entire dataset (X_data)

5. Prediction_Model ← Model.predict(X_data)

// **Append the predictions to the Level 1 predictions array**

6. Predictions_Level1.append(Prediction_Model)

7. End For

// Create a new dataset using the predictions of Level 1 models

8. New_Dataset ← Transpose(Predictions_Level1) // Each prediction is a new feature

// Define the level 2 Enhanced TabNet Classifier model

9. Enhanced_Model ← **Enhanced TabNet** (explained in **Algorithm 3.10**)

// train Enhanced TabNet classifier to make predictions on the new dataset

10. Predicted_Labels ← Enhanced_Model.predict(New_Dataset)

// Evaluate the model using Confusion Matrix (CM): Explained in **Equations 2.15 to 2.18**

14. CM ← Calculate_Confusion_Matrix(True_Labels, Predicted_Labels)

15. Evaluate the model using network performance measurement Explained in Equations 2.19 and 2.20.

// Return the predicted output labels and evaluation metrics

11. Return Predicted_Labels, CM

End

Algorithm 3.4 illustrates a DT Classifier for 5G Network Slicing Dataset where the Input is 5G Network slicing dataset with labeled classes (features and output labels). While the output is Predicted output labels using the DT Classifier. In addition, **Algorithm 3.5** illustrates a RF Classifier for 5G Network Slicing dataset where the Input tis 5G Network slicing Dataset with labeled classes (features and output labels). While the output is Predicted output labels using the RF Classifier. Moreover, **Algorithm 3.9** illustrates ANN Classifier for 5G Network Slicing Dataset where the Input tis 5G Network slicing Dataset with labeled classes (features and output labels). While the output is Predicted output labels using the ANN Classifier.

Algorithm 3.9: ANN Classifier for 5G Network Slicing Dataset

Input: $Norm\_5G$ with **F** features and **L** labels.

Output: *Predicted_output* labels using the **ANN** Classifier.

Begin

1. X ←Norm_5G.

//Define the dataset as X_data (input features) and Y_data (output labels)

2. X_data← Features from the dataset

3. Y_data ← Output labels from the dataset

// Split the dataset into training and testing sets indicating an 70-30 split

4. X_train, Y_train ← 70% of the data will be used for training

5. X_test, Y_test ← 30% for testing.

// Create ANN Classifier with a specified parameters

6. ANN_Model ← Create_ANNClassifier(**parameters**)

// Fit the model using the training data

7. ANN_Model ← fit(X_train, Y_train)

// Make Predictions using the trained model on the entire dataset

8. ANN_Model ← predict(X_data)

9. Predicted_Labels ← ANN_Model

// Save Predictions to a file for further analysis

10. **Predictions_ANN**← Predicted_Labels

End

Strictly, the dataset was submitted for the first, second, and third models, and the prediction results were stored in a new dataset, which will attend as input for the proposed model's second level (the enhanced TabNet classifier level). **Algorithm 3.10** represents the enhanced TabNet classifier.

Algorithm 3.10: Enhanced TabNet Classifier for 5G Network Slicing Dataset

Input: New Dataset from level 1 with **F** features (p1, p2, p3) and **L** labels**.**

Output: *Predicted_output* labels using the **Enhanced TabNet Classifier**.

Begin

1. X← Norm_5G

//Define the dataset as X_data (input features) and Y_data (output labels)

2. X_data← Features from the dataset

3. Y_data ← Output labels from the dataset

// Split the dataset into training and testing sets indicating an 70-30 split

4. X_train, Y_train ← 70% of the data will be used for training

5. X_test, Y_test ← 30% for testing.

// Train a TabNet Classifier:

6. - Initialize the model with hyper-parameters:

7. - Number of decision blocks (B), attention features (D), gamma value (γ).

8. - Apply input features to batch normalization layer.

//Feature Transformer Layer:

9. - Calculate common features (C_i) using Fully Connected Layer 1.

10. $C_i \leftarrow FC1$

11. - Calculate separate features (S_i) using Fully Connected Layer 2.

12. $S_i \leftarrow FC2$

13. - Combine outputs using Swish activation function as explained in Equation (3.3):// **modified step**

14. $\boldsymbol{f_i \leftarrow \mathrm{Swish}(C_i + S_i) * \sqrt{0.5}}$

//Split Layer:

15. - Apply mask M to input features.

16. - Divide masked features M_f into two parts:

17. - Part 1 $Final\_out$ for calculating the final output of the model.

18. – Part2 M_next for calculating the Mask layer for the next step.

19. For steps = 1 to s:

//Attentive Transformer Layer:

20. - Generate new feature representation using Fully Connected Layer

21. $A_i \leftarrow FC$

22. - Pass output to Batch Normalization layer:

23. $BN \leftarrow FC$

24. - Apply attention using Sparsemax activation function for prior scale term at the (i - 1)th step multiply by trainable function

25. to the processed features from the (i - 1) step as explained in Equations (2.5), (2.6) and (2.7)::

26. $M[i] \leftarrow Sparsemax(P[i - 1] * h\_i(a[i - 1]))$

27. - Indicate important features for the current step using Mask layer.

28. - Repeat Feature Transformer Layer.

29. - Repeat Split Layer.

30. - Pass output to ReLU Layer:

31. - Apply ELU activation function to features explained in equation **(3.4) .** **//Modified Step**

32. $$\boldsymbol{ELU(\mathrm{M}\_f) \leftarrow \begin{cases} x & if\ x \geq 0 \\ \alpha * (e\hat{}x - 1) & if\ x < 0 \end{cases}}$$

33. End for

34.- Sum the output vectors from each step.

12. Make Predictions: train Enhanced TabNet classifier to make predictions on the new dataset

13. Predicted_Labels ← Enhanced_Model.predict(New_Dataset)

// Evaluate the model using Confusion Matrix (CM)

35. End

3.2.4 Case 2 network slicing classification Model

To address the challenge of limited data availability, the methodology based on the concept of TL. Specifically, a fine-tuning process is applied to a pre-existing TabNet model, utilizing a restricted set of labeled data tailored for 5G service classification. By initializing the TabNet model with weights from a pre-trained model, the extensive information gained during pre-training is effectively leveraged. Subsequently, the model undergoes fine-tuning with a limited amount of labeled data, enabling it to adapt to the distinct attributes of the 5G network slice classification issue. For case 2, two types of datasets are generated: a source dataset comprising 13,000 user demands and a target dataset with three cases (100, 200, and 300 user demands), as shown in **Table 3.2.**

Table 3.2: Case 2 5G network slicing dataset

No.	Classes	Source domain dataset	Target domain dataset		
			Case 1	Case 2	Case 3
1	UHD Video streaming	1000	5	12	30
2	Immersive experience	309	3	27	28
3	Smart grid	950	18	15	44
4	e-health	99	13	9	20
5	ITS	5000	9	17	55
6	Vo5G	598	14	38	8
7	Connected vehicles	600	2	14	39
8	Industry automation	3000	26	20	16
9	Video surveillance	1444	10	48	60
	The total no. of instances	13000	100	200	300

The TabNet Model used for TL with Tabular Data: The method is in two stages as shown in **Figure 3.7.**

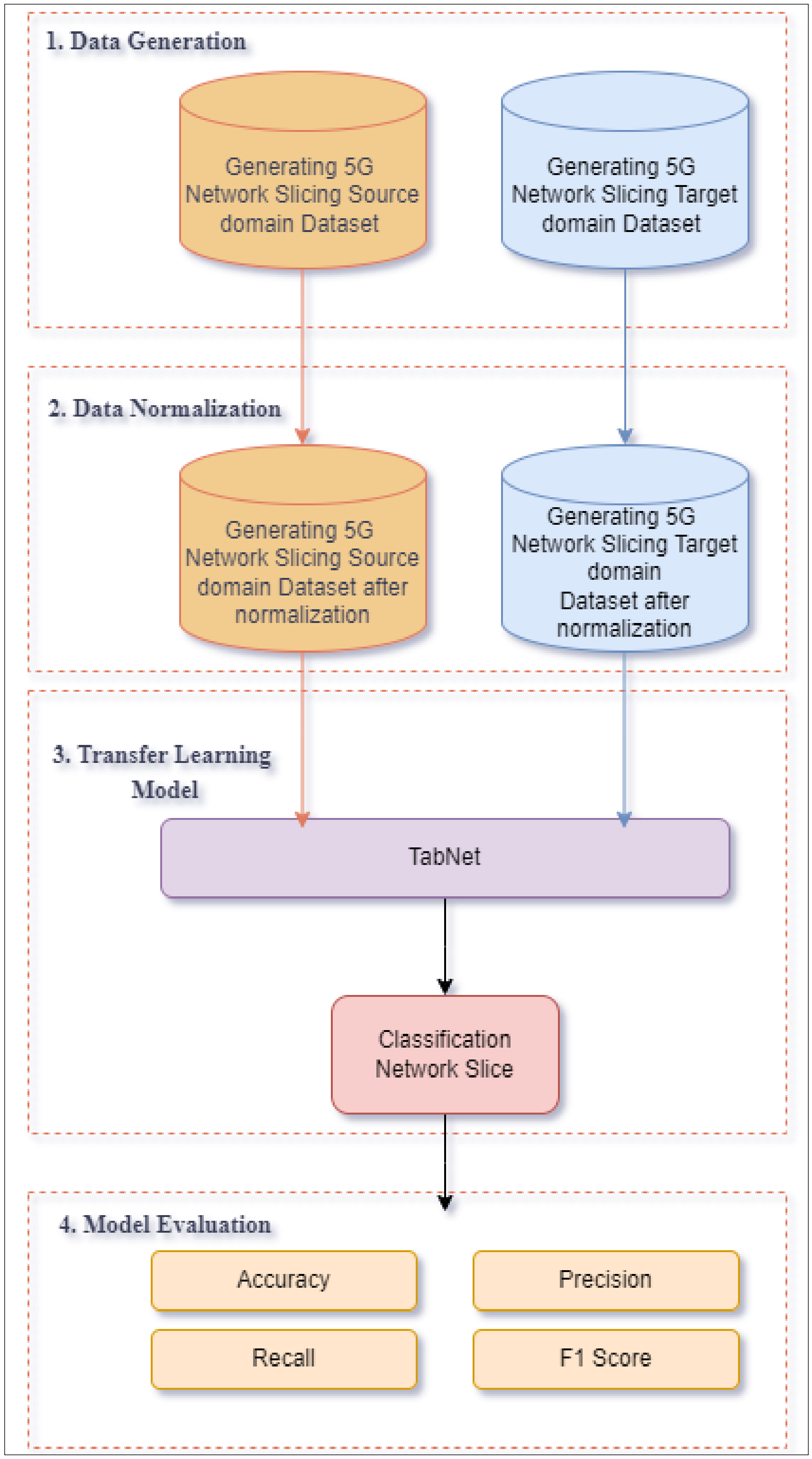

Figure 3.7: 5G network slicing classification case 2

In this dissertation, TabNet Model used for TL with Tabular Data, TabNet is primarily used as a DL model for tabular data and doesn't have established pretraining on large-scale datasets like image-based models. The method is in two stages, firstly: Training on a Source Domain; The model learns to extract meaningful representations and patterns from the input features in the tabular data. Secondly: Fine-Tuning on Target Domain: During fine-tuning, the model adjusts its weights based on the labeled data in the target domain to learn domain-specific patterns and improve performance. **Figure 3.8** shows a TabNet as a TL model.

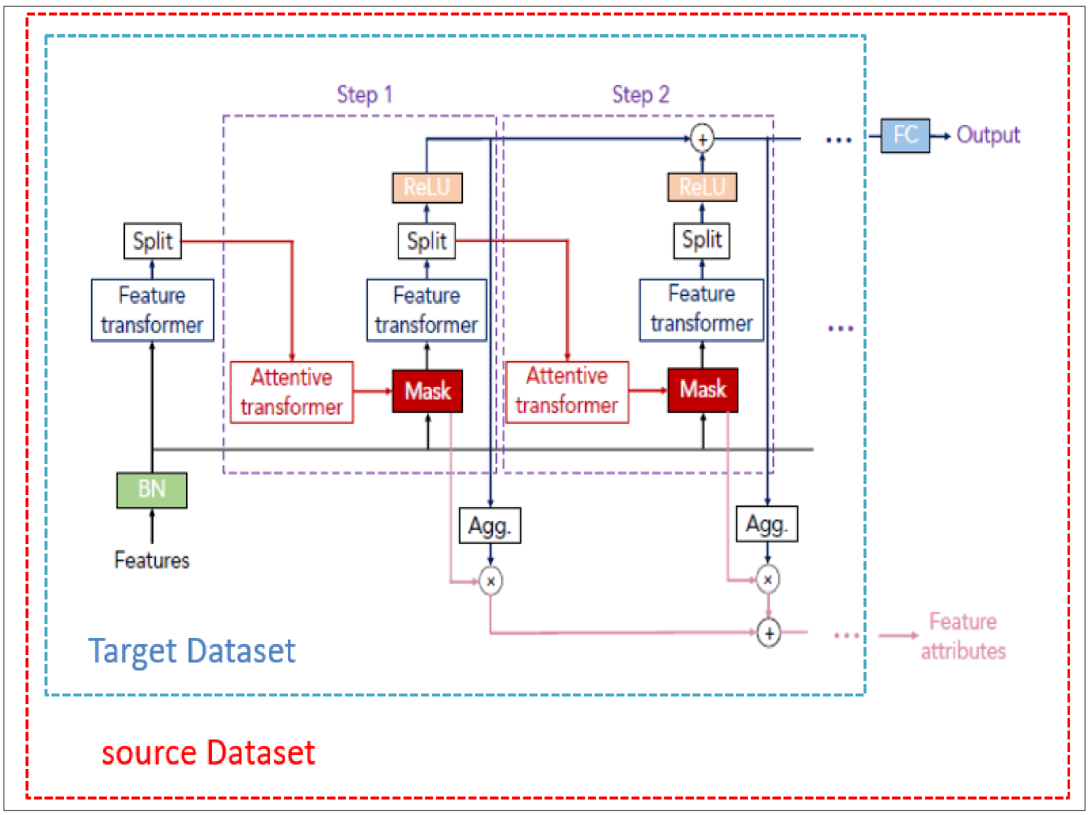

Figure 3.8: A TabNet as a TL model

Algorithm 3.11 explained a TabNet as a TL model:

Algorithm 3.11: Case 2 network slicing classification Model

Input: $Source\_Domain\_5G$ with **F** features and **L** labels and $Target\_Domain\_5G$.

Output: predictions.

Begin

// **Load the source domain dataset**

// **Split the dataset into input features (X_data) and output labels (Y_data).**

1. X_data← Features from the dataset
2. Y_data ← Output labels from the dataset

// **Split the dataset into training and testing sets indicating an 70-30 split**

3. X_train, Y_train ← 70% of the data will be used for training
4. X_test, Y_test ← 30% for testing.

// **Train a TabNet Classifier by Creating an TabNet Classifier model with a specified hayper-parameters**

5. // Fit the model using the training source domain data (**X_train**, **Y_train**).

   // **Save the trained source domain model.**
6. Saved_model ← trained model

   // **Load the source domain dataset**
7. Split the dataset into input features (**Xt_data**) and output labels (**Yt_data**).
8. Split the input features (**Xt_data**) and output labels (**Yt_data**) into training and testing sets.

//Train a TabNet Classifier

9. Creating an TabNet Classifier model with a specified hayper-parameters and load the weights from the trained source domain model. Fit the model using the training source domain data (**Xt_train**, **Yt_train**) .

//Make Predictions:

10. Use the trained TabNet model to make predictions .

11.//Evaluation measurement of the TabNet model.

End

Chapter Four

Results and Discussion

Chapter Four

Results and Discussion

4.1 Introduction

This chapter provides an overview of the implementation and outcomes of the proposed system. The suggested system undergoes testing using a benchmark dataset as well as producing datasets that are used in the Case 1 and Case 2 models. The following subsections provide the descriptions of the datasets. Additionally, this chapter presents a detailed description and presentation of the experimental data relevant to the phases of the proposed Model.

4.2 Generating 5G Network Slicing Dataset

In this dissertation, the use of data from [19] in addition to five other sets of data, which were previously explained and clarified in the previous chapters, and these datasets are used with models in the first case and the second case. **Figure 4.1** shows the 3D plot of instance distribution for the generated dataset 1, which contains **3000** instances (as explained in **Table 3.1**). Dataset1 represents the data set used for case1 in both models**.**

In addition**, Figures 4.2** shows the 3D plot of instance distribution for the generated dataset 2, which contains **13000** instances (as explained in **Table 3.2**). Dataset2 represents the dataset used for case 2 as a source domain dataset.

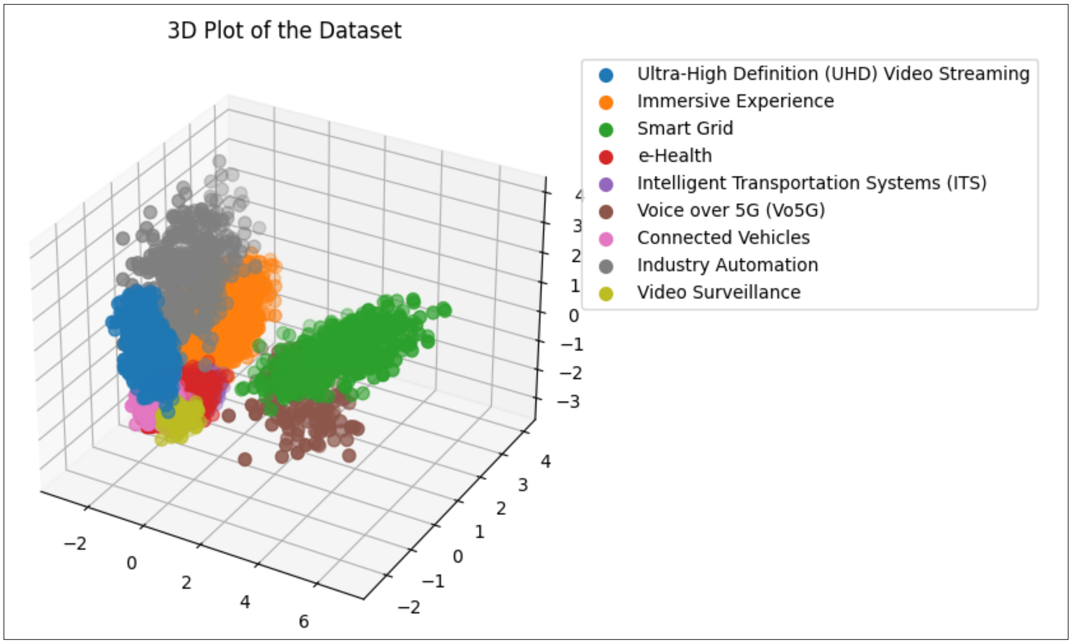

Figure 4.1: 3D plot for instance distribution of the dataset 1

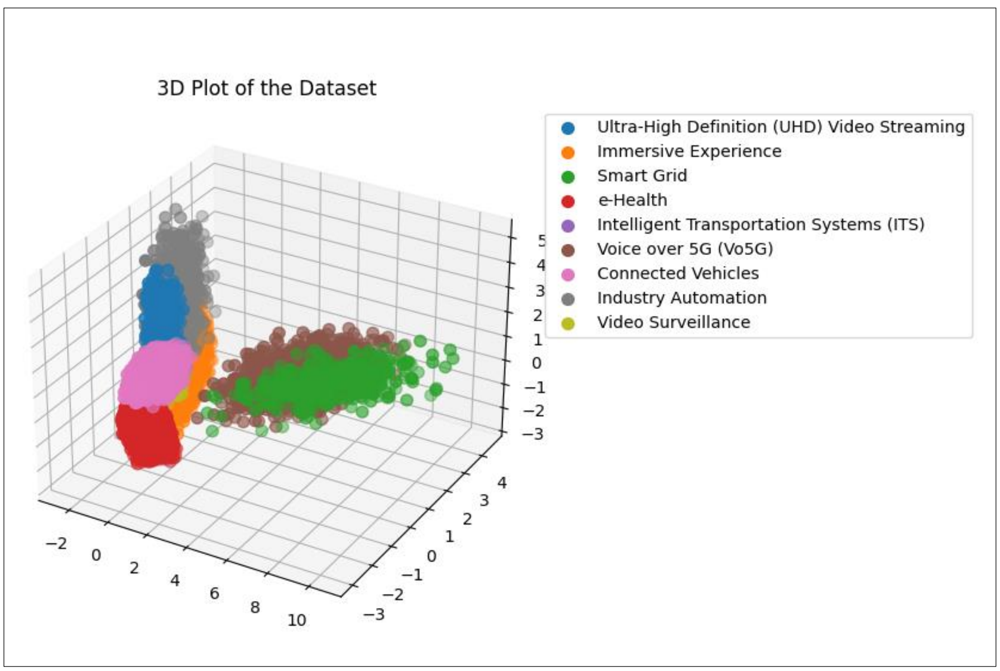

Figure 4.2: 3D plot for instance distribution of the dataset 2

While**, Figure 4.3** shows the 3D plot of instance distribution for the generated dataset 3, which contains 100 instances as explained in **Table 3.2**. Dataset3 represents the data set used for case 2 as a first case for the target domain dataset.

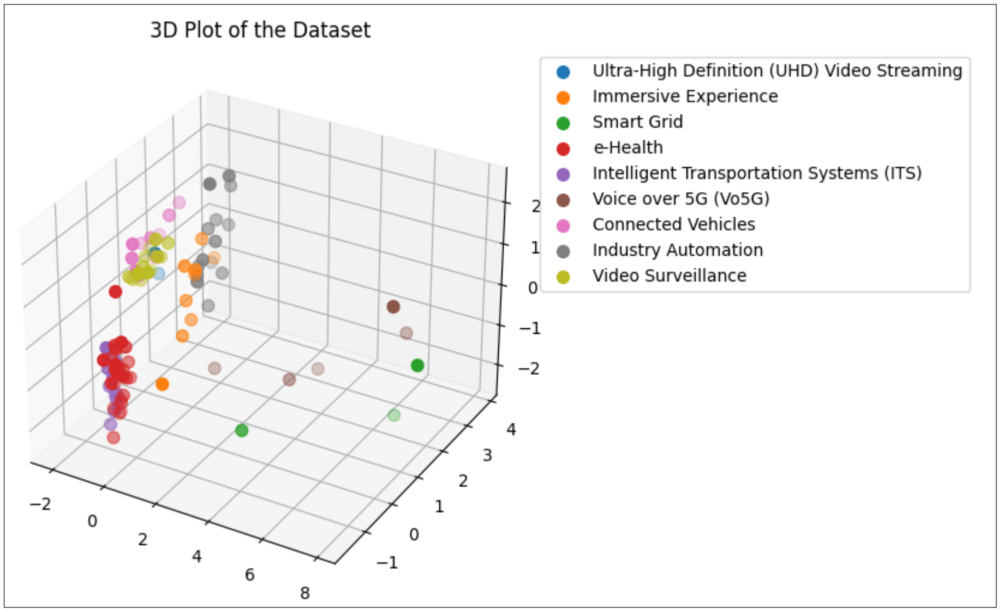

Figure 4.3: 3D plot for instance distribution of the dataset 3

Moreover**, Figure 4.4** shows the 3D plot of instance distribution for the generated dataset 4, which contains **200** instances (as explained in **Table 3.2**). Dataset 4 represents the data set used for case 2 as a second case for the target domain dataset. Furthermore, **Figure 4.5** shows the 3D plot of instance distribution for the generated dataset 5, which contains 200 instances (as explained in **Table 3.2**). Dataset 5 represents the data set used for Case 2 as a third case for the target domain dataset.

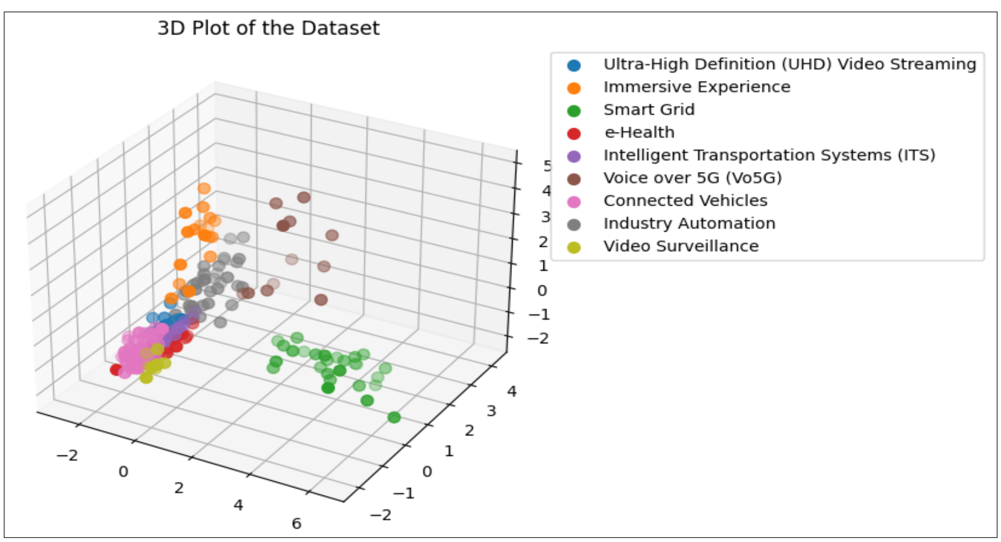

Figure 4.4: 3D plot for instance distribution of the dataset 4

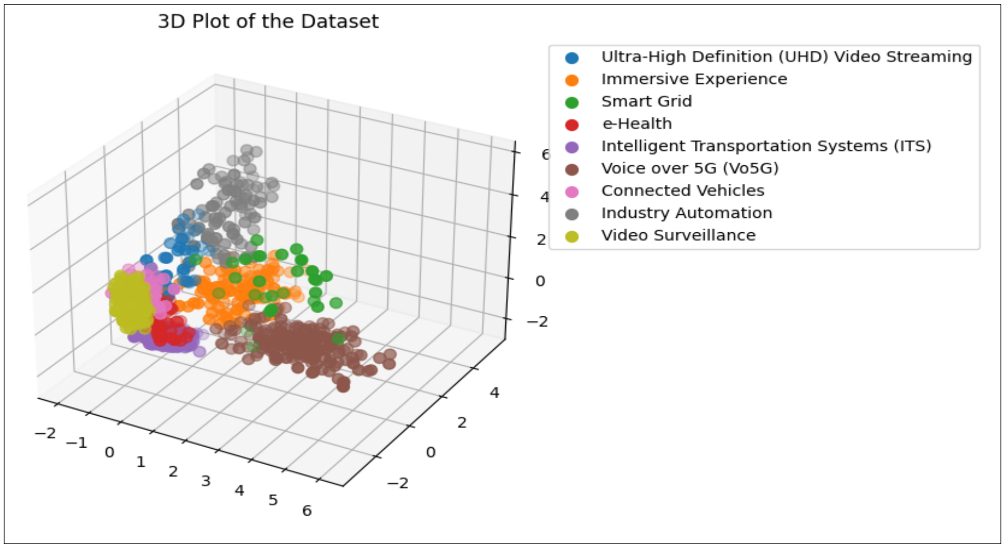

Figure 4.5: 3D plot for instance distribution of the dataset 5

4.3 Case 1 Network Slicing Classification Models

In case 1 a 5G network slicing dataset, comprising 3000 users, is utilized, and the network slices are classified through the application of two Models.

1) Model 1

Model 1 proposed design consist of two levels. Level 1 consist of DT, RF and TabNet; While level 2 consists of hybrid classifier (PSO-ANN) as mentioned in section 3.2. In this case, two experiments were implemented. Experiment 1 is about implementing the proposed model on dataset in [19] .

The results show that the proposed model exceeds the model in [19] on all measures (accuracy, precision, recall, and F1-score). It has a greater accuracy, precision, and recall, implying that it makes more accurate predictions, particularly for positive classifications.

In addition, a higher F1-score, indicating a better balance of precision and recall. These findings indicate that the proposed model outperforms the model in [19] in terms of predictive performance.

When considering the implications for latency and reliability in 5G, the enhanced performance of the proposed model suggests potential advantages in real-time processing and reliable decision-making. The higher accuracy and precision imply a reduced likelihood of misclassifications, contributing to lower latency in delivering accurate results.

Additionally, the improved recall and F1-score indicate a better balance between sensitivity and precision, enhancing the model's reliability in capturing relevant information without losing precision. These factors collectively contribute to the potential for reduced latency and increased reliability, critical aspects in the context of 5G networks. The comparison results are shown in **Table 4.1**.

Table 4.1: Results comparison between Model in [19] and the proposed model 1

Model	Accuracy	Precision	Recall	F1-score
Model in [19]	96.6	97.2	96.3	96.2
Proposed model	97	98	97	97

In Experiment 2, the dataset created in the approach was used for the application of the suggested model. The outcomes and findings indicate that the suggested model produced good results in comparison to other standalone models, including DT, RF, TabNet and DNN+PSO. **Table 4.2** shows the results comparison between the proposed model with models (DT, RF, TabNet and DNN+PSO).

Table 4.2: The comparison results between models (DT, RF, TabNet and DNN+PSO) with the proposed model

Model	Accuracy	Precision	Recall	F1-score
DT	81	79	79	79
RF	90	81	81	81
DNN + PSO	91	82	82	82
TabNet	94	81	81	81
Proposed Model	97.6	83	83	83

As shown in the **Table 4.2**, the findings indicate that the DT displays a reasonable level of performance across all evaluation metrics. Thus, in the precise field of network slice classification, it is considered appropriate for simpler slice classification tasks. In contrast, RF has higher performance across many matrices, demonstrating its capability to handle a broader range of slice types.

In a similar way, the combination of DNN and PSO confirmed worthy outcomes, showing its ability in capturing difficult patterns within network slice data and its potential applicability across many classification tasks.

The TabNet model showed robust performance, achieving a harmonious balance between accuracy and recall, as well as a commendable F1-score. These results highlight the model's effectiveness in addressing complex categorization tasks.

The Proposed Model confirmed excellent performance in terms of accuracy, precision, recall, and F1-score, locating it as a very promising option for the classification of network slices across several categories. The excellent performance shown by the Proposed Model indicates a deep considerate of the underlying patterns in the data, making it highly suitable for accurate classification in the field of network slicing.

The Proposed Model displayed good accuracy, precision, recall, and F1-score. These robust results suggest the potential for reduced latency, as the model offers more accurate predictions, particularly crucial for real-time applications. Additionally, the deep understanding of underlying data patterns showcased by the Proposed Model enhances its reliability, making it well-suited for critical services requiring precise and timely decision-making in the situation of 5G networks.

Figure 4.6 presents a summary of evaluation matrices, comparing models such as DT, RF, TabNet, and DNN+PSO with the proposed model. Additionally, an overall comparison is made with the benchmark model.

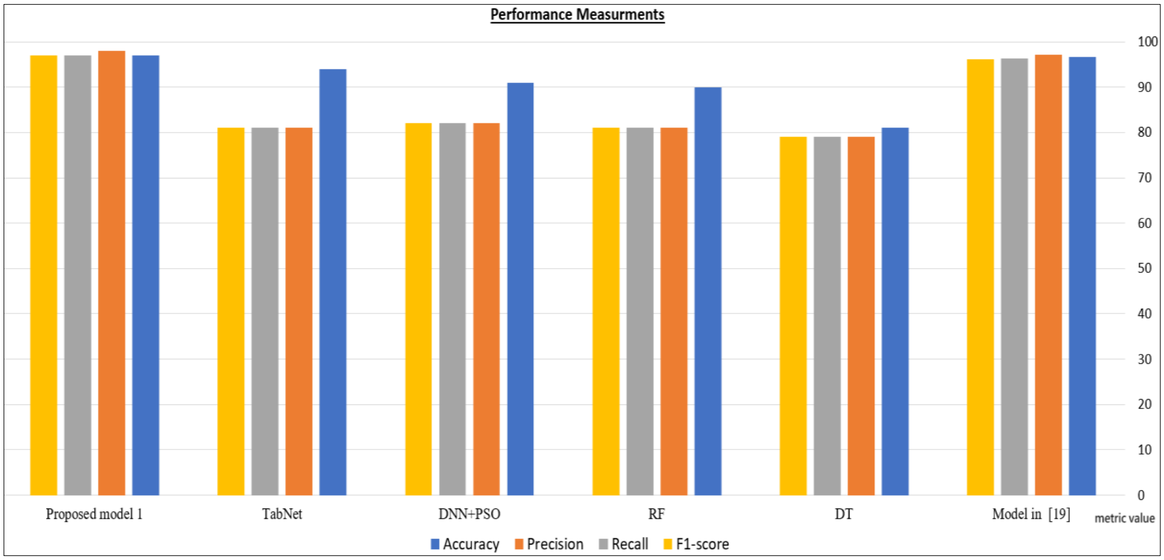

Figure 4.6: Evaluation matrices comparison for model 1

2) Model 2

In this scenario, two experiment were showed. **Experiment 1** contains the application of the proposed model to the dataset described in reference [19]. The findings specify that the suggested model outperforms the model presented in [19] across all evaluation metrics, including accuracy, precision, recall, and F1-score. The model shows enhanced accuracy, precision, and recall, indicating its ability to provide more precise predictions, especially in the situation of positive classifications. Moreover, the F1-score is greater, suggesting a more optimal trade-off between accuracy and recall.

The results of this study suggest that the model we have developed exhibits good predictive performance compared to the model described in [19]. The comparison results are shown in **Table 4.3.**

Table 4.3: Results comparison between model in [9] and proposed model 2

Model	Accuracy	Precision	Recall	F1-score
Model in [19]	96.6	97.2	96.3	96.2
Proposed model	98	98	97	97

In terms of latency and reliability in 5G network slicing, these findings suggest promising implications. The heightened accuracy and precision of the proposed model imply a potential reduction in latency, as the model can deliver faster and more precise predictions. This is particularly crucial for applications demanding real-time responsiveness like e-health. Moreover, the increased recall and F1-score indicate a better ability to capture relevant information without sacrificing precision, contributing to the overall reliability of the model in network slicing scenarios.

For **Experiment 2**, the proposed model was applied to the dataset that was generated in the method. The outputs and results demonstrate that the proposed model yielded butter outcomes compared to other models alone such as (DT, RF, DNN, and TabNet). **Table 4.4** shows the results comparison between models (DT, RF, DNN, and TabNet) with the proposed model.

Table 4.4: The results comparison between models (DT, RF, DNN, and TabNet) with the proposed model 2

Model	Accuracy	Precision	Recall	F1-score
DT	81	82	79	78
RF	91	90	91	86
DNN	91	92	91	90
TabNet	94	95	94	94
Proposed Model	98	98	98	98

After implementing some models and the proposed model the results shows that the DT presents a moderate performance across all matrices, so in the context of network slice classification it is suitable for simpler slice classification. While RF, on the other hand, demonstrate a higher performance across matrices, so it is capable for a wider array of slice categories. Similarly, DNN achieved competitive results, exhibiting its ability to catch subtle patterns within network slice data and its potential for numerous categorization tasks.

The TabNet model demonstrated strong performance, with balanced precision and recall and a solid F1-score, demon- starting its efficacy in dealing with complicated categorization challenges.

In terms of latency and reliability within the context of 5G and its services, the performance of Proposed Model 2 suggests potential advantages. The good accuracy and precision imply a potential reduction in latency, making the model suitable for real-time applications. Additionally, the balanced precision and recall, as well as the solid F1-score, indicate a reliable ability to handle complicated categorization tasks, contributing to the overall reliability of the model in the dynamic environment of 5G network slicing.

Figure 4.7 shows a summary of evaluation matrices, comparing models such as DT, RF, DNN, and TabNet with the proposed model. Additionally, an overall comparison is made with the benchmark model.

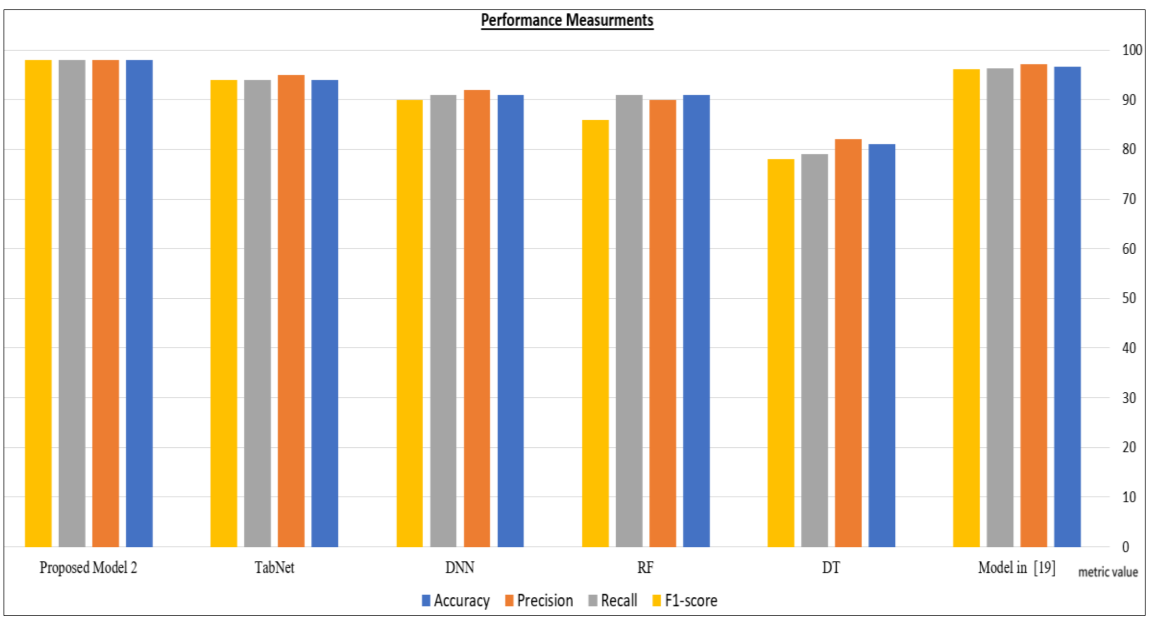

Figure 4.7: Evaluation matrices comparison for model 2

4.4 Case 2 Network Slicing Classification Model

In this case, a source dataset including **13,000** instances is assumed, with three other dataset representing scarce data. TabNet is trained on a domain-specific tabular dataset, where the model acquires the capability to extract meaningful representations and patterns from the input features. During the fine-tuning process, the model adjusts its weights based on labeled data in the target domain, aiming to discern domain-specific patterns and enhance overall performance. A DT was employed as the classifier for the target data. TL was executed on three cases of target data (**100, 200, and 300 instances**) representing scarce data. Accuracy was assessed using DT as a classification model both before and after TL using TabNet, revealing a significant increase in accuracy as shown in **Table 4.5.**

Table 4.5: Results before and after using Transfer Learning

Model	**Decision tree**		
Data	Data1:100	Data2: 200	Data3:300
Acc. before using TL	70%	68%	73%
Acc. after using TL	72%	73%	77%

The experimental results demonstrate that the proposed approach achieves good classification accuracy after using TL.

For more clarity, the rest of evaluation metrices was computed for data set 1 (as an example) and the classification report for each class before and after TL clarified in **Figures 4.6 (a) and 4.6 (b)**. As observed some classes are improved in term of precision, recall and accuracy after using TL.

Class	Class Label	Precision	Recall	F1-Score
1	Ultra-High Definition (UHD) Video Streaming	0.00	0.00	0.00
2	Immersive Experience	0.00	0.00	0.00
3	Smart Grid	0.00	0.00	0.00
4	e-Health	1.00	1.00	1.00
5	Intelligent Transportation Systems (ITS)	1.00	1.00	1.00
6	Voice over 5G (Vo5G)	0.00	0.00	0.00
7	Connected Vehicles	0.50	1.00	0.67
8	Industry Automation	1.00	1.00	1.00
9	Video Surveillance	0.57	1.00	0.73

a

Class	Class Label	Precision	Recall	F1-Score
1	Ultra-High Definition (UHD) Video Streaming	0.00	0.00	0.00
2	Immersive Experience	0.00	0.00	0.00
3	Smart Grid	0.00	0.00	0.00
4	e-Health	1.00	1.00	1.00
5	Intelligent Transportation Systems (ITS)	1.00	1.00	1.00
6	Voice over 5G (Vo5G)	0.00	0.00	0.00
7	Connected Vehicles	0.50	1.00	0.67
8	Industry Automation	1.00	1.00	1.00
9	Video Surveillance	0.57	1.00	0.73

b

Figure 4.6: Classification report for target dataset 1 (a) before TL, (b) after TL

As observed from Figures 4**.6 (a) and 4.6 (b),** the model struggles to provide meaningful metrics for several classes before transfer learning. Some classes have precision, recall, and F1-score values of zero, indicating a lack of correct predictions.

After TL, the model shows improvement, providing meaningful metrics for each class. Precision, recall, and F1-score values are now non-zero for all classes, indicating the model's ability to make accurate predictions.

TL has significantly enhanced the model's performance, enabling it to provide meaningful predictions for each class, which is crucial for accurate classification in the context of network slicing.

Chapter Five
Conclusions and Future Work

Chapter Five

Conclusions and Future Work

5.1 Introduction

The evolution of telecommunication networks, specifically focusing on the significant advancements brought by the 5G. With the rise in the number of connected users and data volume, the challenges related to effective resource utilization, network management, and meeting diverse service requirements have become paramount. These challenges have shown the pivotal role of C-RAN and network slicing. Network slicing has emerged as a key enabler, allowing the creation of isolated logical networks tailored to specific service needs. In the related literature, ML and DL classification methods are considered for network slicing classification.

5.2 Conclusions

This dissertation has arrived at the following conclusions:

1) Overcoming the challenge of tabular data in DL, especially in contrast to more common data types like images and text, is achieved through the implementation of specialized tabular DL models.
2) Handling unbalanced data, known for its complexity, is addressed, recognizing the potential bias towards the majority class and mitigating inaccuracies in predicting the minority class.
3) The selection of a network slice to meet user demands is accomplished by employing a model based on both tabular and ML models, followed by a hybrid optimization classifier.
4) An alternative approach involves selecting a network slice using a model based on ANN and ML, complemented by an enhanced TabNet model.

5) Overcoming the challenge of scarce data is addressed through the application of TL principles, specifically utilizing a TabNet model.

5.3 Suggestions for Future Works

In this dissertation, the following are suggested for future works:

1) **Exploring Various Data Sources:** Future research can focus on exploring diverse data sources and methodologies for generating relevant datasets. Collaborations with telecommunication companies, research institutions, for providing access to real-world data.
2) **Advanced DL Models:** Continued exploration and development of advanced DL models specifically tailored for tabular data could be a promising path. Research can explore into creating DL architectures optimized for handling the challenges posed by tabular datasets, including data sparsity and mixed feature types.
3) **Hybrid Approaches:** Future studies can explore hybrid approaches that combine DL, traditional ML techniques, and domain-specific knowledge to further enhance the efficiency and accuracy of network slicing and service classification processes.
4) **Evaluation Metrics:** Developing comprehensive evaluation metrics specific to 5G network slicing and classification tasks would provide a standardized way to assess the performance of different models. These metrics should consider factors such as resource utilization, Throughput, energy efficiency, and user satisfaction and other QoS factors.

References

References

[1] 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," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Daegu, Korea (South) :2020 ,IEEE, pp. 19-24, doi: https://doi.org/10.23919/APNOMS50412.2020.9237008 .

[2] A. Gausseran, "Optimization algorithms for network slicing for 5G," PhD Thesis, Université Côte d'Azur, 2021 .

[3] N. Mharsi, "Cloud-Radio Access Networks: design, optimization and algorithms," PhD Thesis, Université Paris-Saclay (ComUE), 2019 .

[4] M. J. Abood and G. H. Abdul-Majeed, "Classification of network slicing threats based on slicing enablers: A survey," *International Journal of Intelligent Networks*, vol. 4, pp ,2023 ,112-103 .doi: https://doi.org/10.1016/j.ijin.2023.04.002.

[5] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, "Consideration on automation of 5G network slicing with machine learning," in *2018 ITU Kaleidoscope: Machine Learning for a5 G Future (ITU K)*, Santa Fe, Argentina, 2018: IEEE, pp. 1-8, doi: https://doi.org/10.23919/ITU-WT.2018.8597639 .

[6] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957-994, 2021, doi: https://doi.org/10.1109/COMST.2021.3067807.

[7] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE access*, vol. 7, pp. 137184-137206, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2942390.

[8] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," *Computers*, vol. 12, no. 5, p. 91, 2023, doi: https://www.mdpi.com/2073-431X/12/5/91#.

[9] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2019: IEEE, pp. 0762-0767, doi: https://doi.org/10.1109/UEMCON47517.2019.8993066 .

[10] M. A. Kamal, H. W. Raza, M. M. Alam, M. M. Su'ud, and A. b. A. B. Sajak" ,Resource allocation schemes for 5G network: A systematic review," *Sensors*, vol. 21, no. 19, p. 6588, 2021, doi: https://doi.org/10.3390/s21196588.

[11] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-21, 2022, doi: https://doi.org/10.1109/TNNLS.2022.3229161.

[12] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249-259, 2018, doi: https://doi.org/10.1016/j.neunet.2018.07.011.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp ,2009 ,1359-1345 . doi: https://doi.org/10.1109/TKDE.2009.191.

[14] A. Zhang *et al.*, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Applied Sciences*, vol. 8, no. 12, p. 2416, 2018, doi: https://www.mdpi.com/20.#2416/12/8/3417-76

[15] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "A network slicing prototype for a flexible cloud radio access network," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 201 :8IEEE, pp. 1-4, doi: https://doi.org/10.1109/CCNC.2018.8319259 .

[16] S. Costanzo, S. Cherrier, and R. Langar, "Network Slicing Orchestration of IoT-BeC 3 applications and eMBB services in C-RAN," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019: IEEE, pp. 975-976, doi: https://doi.org/10.1109/INFCOMW.2019.8845076 .

[17] R. K. Gupta and R. Misra, "Machine learning-based slice allocation algorithms in 5G networks," in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, India, 2019: IEEE, pp. 1-4, doi: https://doi.org/10.1109/ICAC347590.2019.9036741 .

[18] S. Barmpounakis, N. Maroulis, M. Papadakis, G. Tsiatsios, D. Soukaras, and N. Alonistioti" ,Network slicing-enabled RAN management for 5G: Cross layer control based on SDN and SDR," *Computer Networks*, vol. 166, p. 106987, 2020, doi: https://doi.org/10.1016/j.comnet.2019.106987.

[19] J. E. Preciado-Velasco, J. D. Gonzalez-Franco, C. E. Anias-Calderon, J. I. Nieto-Hipolito, and R. Rivera-Rodriguez, "5g/b5g service classification using supervised learning," *Applied Sciences*, vol. 11, no. 11, p. 4942, 2021, doi: https://www.mdpi.com/2076-3417/11/11/4942#.

[20] Á. Gabilondo García *et al.*, "Traffic Classification for Network Slicing in Mobile Networks," *Electronics*, vol. 11, no. 7, 2022.

[21] N. A. Mohammedali, T. Kanakis, A. Al-Sherbaz, and M. Opoku Agyeman, "Management and Evaluation of the Performance of end-to-end 5G Inter/Intra Slicing using Machine Learning in a Sustainable Environment," *Journal of Communications Software and Systems,* vol. 19, no. 1, pp. 91-102, 2023, doi: https://doi.org/10.24138/jcomss-2022-0163.

[22] Q. Zeng, Q. Sun, G. Chen, H. Duan, C. Li, and G. Song, "Traffic prediction of wireless cellular networks based on deep transfer learning and cross-domain data," *IEEE access,* vol. 8, pp. 172387-172397, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3025210.

[23] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot," in *2020 IEEE 14th international conference on big data science and engineering (BigDataSE)*, Guangzhou, China, 2020: IEEE, pp. 88-95, doi: https://doi.org/10.1109/BigDataSE50710.2020.00020 .

[24] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. Abd El-Latif, "A secure federated learning framework for 5G networks," *IEEE Wireless Communications,* vol. 27, no. 4, pp. 24-31, 2020, doi: https://doi.org/10.1109/MWC.01.1900525.

[25] J. Zeng *et al.*, "Deep transfer learning for 5G massive MIMO downlink CSI feedback," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, 2021: IEEE, pp. 1-5, doi: https://doi.org/10.1109/WCNC49053.2021.9417349 .

[26] J. Guan ,J. Cai, H. Bai, and I. You, "Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems," *International Journal of Machine Learning and Cybernetics,* vol. 12, pp. 3351-3365, 2021, doi: https://doi.org/10.1007/s13042-02.4-01415-1

[27] B. Yang *et al.*, "A joint energy and latency framework for transfer learning over 5G industrial edge networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 531-541, 2021, doi: https://doi.org/10.1109/TII.2021.3075444.

[28] T. Hu, Q. Liao, Q. Liu, and G. Carle, "Network Slicing via Transfer Learning aided Distributed Deep Reinforcement Learning," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil, 2022: IEEE, pp. 2909-2914, doi: https://doi.org/10.1109/GLOBECOM48099.2022.10000763 .

[29] M. Attaran, "The impact of 5G on the evolution of intelligent automation and industry digitization," *Journal of ambient intelligence and humanized computing*, vol. 14, pp. 5977-5993, 2023, doi: https://doi.org/10.1007/s12652-020-02521-x.

[30] Y. Tang, S. Dananjayan, C. Hou, Q. Guo, S. Luo, and Y. He, "A survey on the 5G network and its impact on agriculture: Challenges and opportunities," *Computers and Electronics in Agriculture*, vol. 180, p. 105895, 2021 ,doi: https://doi.org/10.1016/j.compag.2020.105895.

[31] A. Agarwal, G. Misra, and K. Agarwal, "The 5th generation mobile wireless networks-key concepts, network architecture and challenges," *American Journal of Electrical and Electronic Engineering*, vol ,3 .no. 2, pp. 22-28, 2015, doi: https://doi.org/10.12691/ajeee-3-2-1.

[32] W. H. A. Al-Zubaedi, "Planning a C-RAN deployment for the next generation cellular networks," PhD Thesis, Brunel University London, College of Engineering, Design and Physical Sciences, Deprtment of Electronic and Electrical Engineering, 2019 .

[33] A. Checko *et al.*, "Cloud RAN for mobile networks—A technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405-426, 2014, doi: https://doi.org/10.1109/COMST.2.014.2355255

[34] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE network*, vol. 29, no. 1, pp. 35-41, 2015, doi: https://doi.org/10.1109/MNET.2015.7018201.

[35] K. M. S. Huq and J. Rodriguez, *Backhauling/Fronthauling for Future Wireless Systems*. John Wiley & Sons, 2016, p. 232.

[36] R. Chhatani and A. Cheeran, "Survey on cloud radio access network," *EAI Endorsed Transactions on Wireless Spectrum*, vol. 16, no. 7, p. e5, 2016, doi: http://dx.doi.org/10.4108/eai.19.2016.151001-1-

[37] S. Papavassiliou, "Software defined networking (SDN) and network function virtualization (NFV)," *Future Internet* vol. 12, no. 1, p. 7, 2020, doi: https://www.mdpi.com/1999-5903/12/1/7#.

[38] D. Scano, L. Valcarenghi, K. Kondepu, P. Castoldi, and A. Giorgetti, "Network slicing in SDN networks," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, Bari, Italy, 2020: IEEE, pp. 1-4, doi: https://doi.org/10.1109/ICTON51198.2020.9203184 .

[39] S. Sridharan, "5G Cloud Network Resource Slicing–A Literature Review," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 11, pp. 1-5, 2020.

[40] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead," *Computer Networks*, vol. 182, p. 107516, 2020, doi: https://doi.org/10.1016/j.comnet.2020.107516.

[41] A. Domeke, B. Cimoli, and I. T. Monroy, "Integration of Network Slicing and Machine Learning into Edge Networks for Low-Latency Services in 5G

and beyond Systems," *Applied Sciences*, vol. 12, no. 13, p. 6617, 2022, doi: https://doi.org/10.3390/app12136617.

[42] S. A. Kazmi *et al.*, "Network slicing: The concept," in *Network Slicing for 5G and Beyond Networks* :Springer Cham, 2019, pp. 13-24.

[43] H. Dujuan, "Mobile communication technology of sports events in 5G era," *Microprocessors and Microsystems*, vol. 80, p. 103331, 2021, doi: https://doi.org/10.1016/j.micpro.2020.103331.

[44] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE communications magazine*, vol. 55, no. 5, pp. 94-100, 2017, doi: https://doi.org/10.1109/MCOM.2017.1600951.

[45] N. Salhab, R. Langar, and R. Rahim, "5G network slices resource orchestration using Machine Learning techniques," *Computer Networks*, vol. 188, p. 107829, 2021, doi: https://doi.org/10.1016/j.comnet.2021.107829.

[46] E. N. Tominaga, H. Alves, O. L. A. López, R. D. Souza, J. L. Rebelatto, and M. Latva-Aho, "Network slicing for eMBB and mMTC with NOMA and space diversity reception," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Helsinki, Finland, 2021: IEEE, pp. 1-6, doi: https://doi.org/10.1109/VTC2021-Spring51267.2021.9448974 .

[47] S. A. AlQahtani, "Cooperative-Aware Radio Resource Allocation Scheme for 5G Network Slicing in Cloud Radio Access Networks," *Sensors*, vol. 23, no. 11, p. 5111, 2023, doi: https://www.mdpi.com/1424-8220/23/11/5111#.

[48] Ericsson. "Helping to shape a world of communication " https://www.ericsson.com/en (accessed.

[49] X. Li *et al.*, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Computing,* vol. 21, no. 5, pp. 20-27, 2017, doi: https://doi.org/10.1109/MIC.2017.3481355.

[50] V. Deepak, N. Alwala, and P. Swathy, "Data Preprocessing Toolkit: An Approach to automate data preprocessing," *International Journal of Scientific Research in Engineering and Management (IJSREM),* vol. 7, no. 2, pp. 1-5, 2023.

[51] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 6, pp. 10173-10196, 2023, doi: https://doi.org/10.1109/TPAMI.2023.3250241.

[52] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain," *Mathematics,* vol. 10, no. 11, p. 1942, 2022, doi: https://www.mdpi.com/2227-7.#1942/11/10/390

[53] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends,* vol. 2, no. 1, pp. 20-28, 2021, doi: https://doi.org/10.38094/jastt20165.

[54] H .Kamel and M. Z. Abdullah, "Distributed denial of service attacks detection for software defined networks based on evolutionary decision tree model," *Bulletin of Electrical Engineering and Informatics,* vol. 11, no. 4, pp. 2322-2330, 2022, doi: https://doi.org/10.11591/eei.v11i4.3835.

[55] L. Rokach and O. Maimon, "Decision trees," in *Data mining and knowledge discovery handbook*: Springer, Bosto, MA, 2005, pp. 165-192.

[56] B. S. Vardhan, M. Khedkar, I. Srivastava, P. Thakre, and N. D. Bokde, "A Comparative Analysis of Hyperparameter Tuned Stochastic Short Term Load Forecasting for Power System Operator," *Energies*, vol. 16, no. 3, p. 1243, 2023, doi: https://www.mdpi.com/1996-1073/16/3/1243#.

[57] S. Imtiaz, G. P. Koudouridis, H. Ghauch, and J. Gross, "Random forests for resource allocation in 5G cloud radio access networks based on position information," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 142, pp. 1-16, 2018, doi: https://doi.org/10.1186/s13638-018-1149-7.

[58] A. A .Jabbar and W. S. Bhaya, "Security of private cloud using machine learning and cryptography," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 561-569, 2023, doi: https://doi.org/10.11591/eei.v12i1.4383.

[59] J. H. Kim, M. L. Lee ,and C. Park, "A Data-Based Framework for Identifying a Source Location of a Contaminant Spill in a River System with Random Measurement Errors," *Sensors*, vol. 19, no. 15, p. 3378, 2019, doi: https://www.mdpi.com/1424-8220/19/15/3378#.

[60] C. Shah, Q. Du ,and Y. Xu, "Enhanced TabNet: Attentive interpretable tabular learning for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 3, p. 716, 2022, doi: https://www.mdpi.com/2072-4292/14/3/716#.

[61] J. Yan, T. Xu, Y. Yu, and H. Xu, "Rainfall forecast model based on the tabnet model," *Water*, vol. 13, no. 9, p. 1272, 2021, doi: https://doi.org/10.3390/w13091272.

[62] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, no. 8, pp. 6679-6687, doi: https://doi.org/10.1609/aaai.v35i8.16826 .

[63] S. Boughorbel, F. Jarray, and A. Kadri, "Fairness in tabnet model by disentangled representation for the prediction of hospital no-show," *arXiv:2103.04* ,048pp. 1-9, 2021, doi: https://doi.org/10.48550/arXiv.2103.04048.

[64] T. Xu, Y. Yu, J. Yan, and H. Xu, "Long-Term Rainfall Forecast Model Based on The TabNet and LightGbm Algorithm," ed, 2020, pp. 1-26.

[65] O. I. Abiodun, A. Jantan, A. E. Omolara, K .V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, p. e00938, 2018, doi: https://doi.org/10.1016/j.heliyon.2018.e00938.

[66] A. Dongare, R. Kharde, and A. D. Kachare, "Introduction to artificial neural network," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189-194, 2012.

[67] S. Haykin, *Neural networks and learning machines*, Third ed. Pearson Education India, 2009, p. 934.

[68] N. Gupta, "Network and Complex Systems Artificial Neural Network. ," *Netw. Complex Syst.*, vol. 3, pp. 24-28, 2013.

[69] R. McLean, B. Ombuki-Berman, and A. P. Engelbrecht, "Swarm Based Algorithms for Neural Network Training," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, 2020: IEEE, pp. 2585-2592, doi: https://doi.org/10.1109/SMC42975.2020.9283242 .

[70] A. Rakitianskaia and A. Engelbrecht, "Weight regularisation in particle swarm optimisation neural network training," in *2014 IEEE symposium on swarm intelligence*, Orlando, FL, USA, 2014: IEEE, pp. 1-8, doi: https://doi.org/10.1109/SIS.2014.7011773 .

[71] A. K. Gumar and F. Demir, "Solar photovoltaic power estimation using meta-optimized neural networks," *Energies*, vol. 15, no. 22, p. 8669, 2022, doi: https://www.mdpi.com/1996-1073/15/22/8669#.

[72] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, 2020, doi: https://doi.org/10.1109/JPROC.2020.3004555.

[73] C. T. Nguyen *et al.*, "Transfer learning for future wireless networks: A comprehensive survey," *arXiv:2102.07572*, pp. 1-39, 2021, doi: https://doi.org/10.48550/arXiv.2102.07572.

[74] L. Joffe, "Transfer Learning for Tabular Data," *TechRxiv* ,pp. 1-14, 2021, doi: https://doi.org/10.36227/techrxiv.16974124.v1.

[75] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, "Multi-class confusion matrix reduction method and its application on net promoter score classification problem," in *The 14th pervasive technologies related to assistive environments conference*, Corfu, Greece, 2021: Association for Computing Machinery, pp. 412-419, doi: https://doi.org/10.1145/3453892.3461323 .

[76] L. P. Joseph, E. A. Joseph, and R. Prasad, "Explainable diabetes classification using hybrid Bayesian-optimized TabNet architecture," *Computers in Biology and Medicine*, vol. 151, p. 106178, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.106178.

[77] L. Wan *et al.*, *5G System Design: An End to End Perspective*. Springer Cham, 2020, pp. XXIII, 393.

[78] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv:1710.05941*, pp. 1-13, 2017, doi: https://doi.org/10.48550/arXiv.1710.05941.

[79] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, vol. 107, pp. 3-11, 2018, doi: https://doi.org/10.1016/j.neunet.2017.12.012.

[80] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv:1511.07289*, pp. 1-14, 2015, doi: https://doi.org/10.48550/arXiv.1511.07289.

[81] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv:1505.00853*, p. 5, 2015, doi: https://doi.org/10.48550/arXiv.1505.00853.

الخلاصة

ان الانتشار السريع لتكنولوجيا الجيل الخامس (5G) قد أدخلنا في عصر من الاتصال الاستثنائي، حيث تستوعب مجموعة واسعة من المستخدمين وتنقل كميات هائلة من البيانات. وسط هذا النمو، ظهر مفهوم شرائح الشبكة في بنية 5G كحل محوري. تتيح تقنية شرائح الشبكة إنشاء شبكات منطقية معزولة مصممة لتلبية متطلبات خدمة محددة، مما يحدث ثورة في مجال الاتصالات. ومع ذلك، يواجه هذا التقدم تحدياته الخاصة، حيث تتمثل إحدى أكبر العقبات في محدودية توفر مجموعات البيانات المناسبة لبناء نموذج قادر على تصنيف شرائح الشبكة التي تلبي متطلبات المستخدمين. بالإضافة إلى ذلك، تبرز الحاجة إلى اختيار شرائح الشبكة بطريقة تضمن أداءً مثاليًا للشبكة، بما في ذلك زمن استجابة منخفض وموثوقية عالية لكل شريحة.

تقدم هذة الاطروحة منهجًا للتعامل مع تصنيف شرائح شبكة 5G. يشمل البحث حالتين مقترحتين:

الحالة ١ والتي تتضمن نموذجين: النموذج الاول وهو نموذج ذو مستويين يجمع بين decision tree, random forest و TabNet، والمصنف الهجين PSO-ANN، صُمّم هذا النموذج من أجل تصنيف شرائح الشبكة مع القدرة على التعامل مع عدم توازن مصادر البيانات. حيث يتمثل الإسهام الرئيسي لهذا النموذج في استخدام نموذج هجين لتقليل وقت التدريب. و اعطى هذا النموذج اعلى معايير الأداء الحالية في جميع مقاييس الأداء. اما النموذج الثاني فهو نموذج محسّن ذو مستويين يجمع بين decision tree, (ANN random forest)، ونوع محسّن من TabNet، صُمّم هذا النموذج لتصنيف شرائح الشبكة من خلال التعامل مع مصادر البيانات المجدولة .وقد أعطي نتائج جيدة مقارنة مع المعايير الحالية في جميع مقاييس الأداء. بالاضافة الى ما سبق يوفر كلا النموذجين إمكانية تقليل زمن الاستجابة وزيادة الموثوقية لتطبيقات 5G في الوقت الفعلي.

اما بالنسبة للحالة ٢ فانها تتضمن نموذج يتعامل مع حالات ندرة البيانات؛ يعتمد النموذج على مبدأ التعلم بالتحويل باستخدام TabNet ومصنف decision tree. حيث يتم تدريب TabNet على مجموعة بيانات ضخمة من المجال المصدر لاستخراج الميزات والأنماط، بينما تتكيف عملية ضبط النموذج الدقيق مع المجال المستهدف، مما يعزز قدرته على تحديد أنماط خاصة بالمجال وتحسين دقة التصنيف.

قد حقق النموذج الأول في الحالة الاولى دقةً تبلغ ٩٧٪، مقارنةً بنموذج المقارنة الذي سجل٩٦,٦٪. بينما أظهر النموذج الثاني أداءً مشابهاً مع دقةٍ تبلغ ٩٨٪، مقارنةً بنموذج المقارنة الذي سجل ٩٦,٦٪، علاوة على ذلك، أظهر نموذج الحالة الثانية وهو التعلم بالتحويل باستخدام (TabNet) تحسينات ملحوظة في الدقة، حيث تم استخدام ثلاث حالات من مجموعات البيانات المحدودة (١٠٠ و ٢٠٠ و ٣٠٠ عينة) تمثل سيناريوهات شبكة 5G، حيث وضحت نتائج التجارب أن منهج التعلم بالتحويل يحقق دقة تصنيف جيدة

(٧٥٪ و ٧٧٪ و ٧٨٪) على التوالي لثلاث حالات من مجموعات البيانات مقارنة بحالات قبل استخدام التعلم بالتحويل حيث تكون الدقة (٧٠٪ و ٧٥٪ و ٧٣٪).

أخيرًا، تشير الدقة العالية للنماذج المقترحة إلى إمكانية تقليل زمن الاستجابة وتحسين الموثوقية، وهو أمر حيوي للتطبيقات والخدمات في الوقت الفعلي.

جمهورية العراق

وزارة التعليم العالي و البحث العلمي

جامعة بابل

كلية تكنولوجيا المعلومات ـ قسم البرمجيات

تصنيف محسّن لشرائح شبكات الجيل الخامس بالاعتماد على طرق تعليم الآلة

اطروحة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات ـ جامعة بابل كجزء من متطلبات نيل درجة دكتوراه فلسفة في تكنولوجيا المعلومات / برمجيات

من قبل:

حوراء شريف حمزة حسين

باشراف:

أ. م . د. مهدي عبادي مانع

١٤٤٥هـ ٢٠٢٣م