A Thesis
For the Degree of Doctor of Philosophy

Ensemble-based Prediction Scheme for Resource Utilization in IBN-enabled Network Slice Lifecycle Management

Khizar Abbas

Department of Computer Engineering

GRADUATE SCHOOL
JEJU NATIONAL UNIVERSITY

February 2022

Ensemble-based Prediction Scheme for Resource Utilization in IBN-enabled Network Slice Lifecycle Management

Khizar Abbas
(Supervised by Professor Wang-Cheol Song)

A thesis submitted to the Department of Computer Engineering and the Faculty of Graduate School of Jeju National University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Engineering

2022. 02.
This thesis has been examined and approved.

.....
Thesis Committee Chair
Khi-Jung Ahn, Professor, Jeju National University

.....
Soo-Kyun Kim, Professor, Jeju National University

.....
Jin Hyun Ahn, Professor, Jeju National University

.....
Seong Baeg Kim, Professor, Jeju National University

.....
Thesis Supervisor,
Wang-Cheol Song, Professor, Jeju National University

Department of Computer Engineering
Graduate School
Jeju National University

Dedicated to
my dearest parents, loving brothers and sisters, Loving brothers in law, and Friends.

ACKNOWLEDGMENTS

First and foremost, my humble praise and gratitude to Allah Almighty, the most gracious, the most merciful, for showering His endless blessings on me throughout my life. Many blessings and salutations on Prophet Muhammad (PBUH), who taught and emphasized the importance of learning and seeking knowledge.

I am incredibly grateful to my supervisor Prof. Wang-Cheol Song for his continuous guidance, suggestions, and constant interaction throughout my Ph.D. studies at Jeju National University.

I offer my humble gratitude to Prof. Khi-Jung Ahn, Prof. Soo-Kyun Kim, Prof. Seong Baeg Kim (Department of Computer Education), and Prof. Jinhyun Ahn (Department of Management Information Systems) for their valuable suggestions and extremely important comments during the process of my thesis evaluation. I would like to extend my sincere gratitude to my Ph.D. course teachers, Prof. Yung-Cheol Byun and Prof. Do-Hyeun Kim, for their tireless hard work, dedication, devotion, and inspiration during my entire course work tenure of doctoral studies.

I would like to acknowledge the company, cooperation, and help of my NCL lab mates during the tenure of the study. I offer special thanks to Dr. Muhammad Afaq, Talha Ahmed Khan, Asif Mehmood, Dr. Adeel Rafiq, Javier Jose Diaz Rivera, Waleed Akbar, and Mir Muhammad Suleman Sarwar for their enormous help and support. I would also like to thank Mr. Waqar Kiyani, Khurram Javed, Dr. Ibrahim, Dr. Muqeet Ur Rehman, Dr. Rashid Ahmad, Dr. Ghayas Ud Din Siddiqi, Dr. Fasihullah Khan, Dr. Shabbir Ahmed, Dr. Qazi Saqib Ul Islam, Shenawar Ali Khan, Rayyan Ali, Umair Khan, and Farhan for their persistent source of encouragement and support.

I would like to thank my friends Dr. Shahbaz Raza, Saim Satti, Prince Waqas Khan, Muhammad Saqib Butt, Muttee Ur Rehman, Muddasir Liaq, Dr. Pyae Pyae Phyo, Muhammad Asim Raza, Waseem Raza, and Sham Satti, who have always supported me in my even and odds. I would like to thank my department fellows Dr. Faisal Jamil, Dr. Imran Jamal, Faisal Mehmood, Naeem Iqbal, Zeinab Shahbazi, Debapriya Hazra, Sedighe Jafari, and others.

I owe a debt of special thanks to Dr. Afaq Muhammad and Talha Ahmed Khan for their cordial cooperation during many occasional lights and dark moments of research that I shared with them. Their encouragement, determination in helping me, kindness, and care allowed me to finish this journey. I would also like to offer special thanks to Asif Mehmood and Shahbaz Raza, who have always been my well-wishers. This journey would have never been easy without their prayers and support.

An honorable mention goes to my dearest parents, loving brothers, sisters, uncle, brothers-in-law, cousins, and family members. I am forever indebted to my mother and father for their

unconditional love and endless prayers. No words can describe their everlasting love to me. I owe a lot to them; they encouraged and helped me at every walk of my life. Also, my distinctive credit to my grandparents, who have been a great source of sustenance and motivation for my studies, I am sure they would have been proud of my achievement if they were alive.

Khizar Abbas

February 2022

Ensemble-based Prediction Scheme for Resource Utilization in IBN-enabled Network Slice Lifecycle Management

Khizar Abbas

Supervisor: Prof. Wang-Cheol Song

ABSTRACT

5G networks come up with many innovative features compared to legacy networks, such as network slicing that envisioned a wide variety of services from different customers, network operators, and industrial verticals. Network slicing is the partitioning of a physical network into multiple logical isolated networks. It ensures dedicated and isolated resources to each of the services. The autonomous orchestration and management of end-to-end (e2e) network slicing is critical due to the complex network configuration for the underlying infrastructure. On the other side, data analytics seems promising to manage and control the underlying network resources proactively. So, Network Data Analytics Function (NWDAF) has been introduced in 5G service-based architecture (SBA), which enables network operators to use various Artificial Intelligence (AI) and Machine learning (ML) techniques. These ML models are trained on historical network data collected from multiple domains such as core, RAN, and edge. It allows network operators to implement their own or third-party ML mechanisms. More specifically, the proactive management of cloud resources is still a challenging task. Therefore, this thesis primarily focuses on e2e network slice lifecycle management and AI and ML-based network data analytics mechanisms for proactive management of network resources.

An Intent-based Networking (IBN) mechanism has been developed to automatically control, orchestrate, and manage e2e network slicing. It follows a closed-loop approach for the

network slice lifecycle management (LCM). The results achieved through the proposed mechanism show satisfactory performance. Moreover, motivated by NWDAF, a data analytics mechanism has been integrated with the IBN platform to achieve proactive resource updates and assurance. This network data analytics mechanism uses novel hybrid ensemble learning (EL) algorithms for network resource utilization prediction and anomaly detection and mitigation. With the help of results, it can be observed that the developed mechanism outperformed the considered algorithms. In addition, ML models assist the IBN platform in updating and managing the network resources proactively.

CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

The traditional networks do not support a wide variety of services and ensure just limited services in terms of messaging, voice, and internet access. These multi-services have different quality of services (QoS) requirements regarding bandwidth, latency, mobility, reliability, and capacity. The primary vision of Fifth Generation (5G) mobile networks is to fulfill the diverse service requirements for different consumers, mobile network operators (MNOs), industrial verticals, and businesses [1] [2]. The future generation mobile networks should accommodate various industrial verticals such as automotive, energy, healthcare, entertainment, media, and manufacturing industry 4.0. These industrial use cases have diverging QoS requirements, and service-oriented architecture

is needed to handle these various use cases. However, the 5G networks have been designed on the service-oriented pattern that efficiently entertains multi-services with desired bandwidth, reliability, and latency [3].

In recent years, software-defined networking (SDN) and network function virtualization (NFV) have emerged as innovative technologies to build virtualized, softwarized, cloudified, highly programmable, and flexible 5G mobile networks [4]. SDN decouples the user/data plane from the control plane. The network functions (NFs) in the control plane run as independent applications under the centralized network controllers [5]. On the other side, NFV enables the network operators (NOs) to deploy their NFs in a Whitebox or generic hardware instead of expansive dedicated hardware devices. So, NOs can quickly deploy their various virtual network functions (VNFs) over the general-purpose servers. Also, mobile edge computing (MEC) [6] emerges as a key technology to overcome the low latency issue, where the storage, computation, and network resources move from the central cloud to the edge closer to the users. It is also one of the critical use cases of the 5G network to ensure ultra-low latency communication for latency-oriented applications [7]. However, network slicing in 5G mobile networks enables the NOs to provide differentiated QoS requirements to each user category.

Network slicing is a primary use case of the mobile 5G networks, and it is possible due to the recent advancement in SDN and NFV computing technologies. Network slicing is partitioning the physical network into multiple logically isolated networks. Each of the logically isolated networks serves a specific group of consumers. It also enables the MNOs to share their infrastructure among other verticals to ensure service guarantees [8]. So, slicing the network is the best choice to accommodate the distinct tailored service requirements from different businesses, industrials verticals, and consumers over the same physical infrastructure. It facilitates

infrastructure providers to use generic hardware devices for implementing multiple VNFs for a slice rather than the legacy hardware devices. So, multiple VNFs are chained together flexibly to establish an end-to-end (E2E) network slice for a specific user group. The fully cloud-native nature of network slicing makes the 5G networks highly flexible and programmable that supports various services efficiently [9].

The International Telecommunication Union (ITU) and Third Generation Partnership Project (3GPP) have divided these diverse services into three major categories: enhanced mobile broadband (eMBB) service, ultra-reliable low latency (URLLC), and massive machine-type communication (mMTC). The eMBB slice category contains ultra-high definition (UHD) communication services, e.g., video streaming. Also, the URLLC slice type includes low latency applications, autonomous driving, and health care industry communication. Besides, the mMTC contains smart agriculture, smart factory, metering, billing, and logistics services type communication [10][11][12]. So, the automatic management and orchestration of these innovative services is required.

The management and orchestration of innovative network services is a very critical task. ETSI has introduced the NFV management and network orchestration (NFV-MANO) platform, which automates the deployment of VNFs in an efficient way. MANO comprises of NFV orchestration (NFVO) layer, VNF Managers (VNFMs), and Virtual Infrastructure Managers (VIMs) [11]. The NFVO is the orchestration entity that can manage the lifecycle of the network services with the cooperation of VNFM and VIM. It is responsible for the deployment of appropriate resources and establishes the connection. The MNOs input network service configurations through the Operation Support System /Business Support System (OSS/BSS), and NFVO deploys and activates the resource over the physical infrastructure with the help of VNFM

and VIM. NFVO has multiple VNMFS and VIMs for the automation and management of resources [10]. Figure 1.1 depicts the components of the NFV-MANO platform.

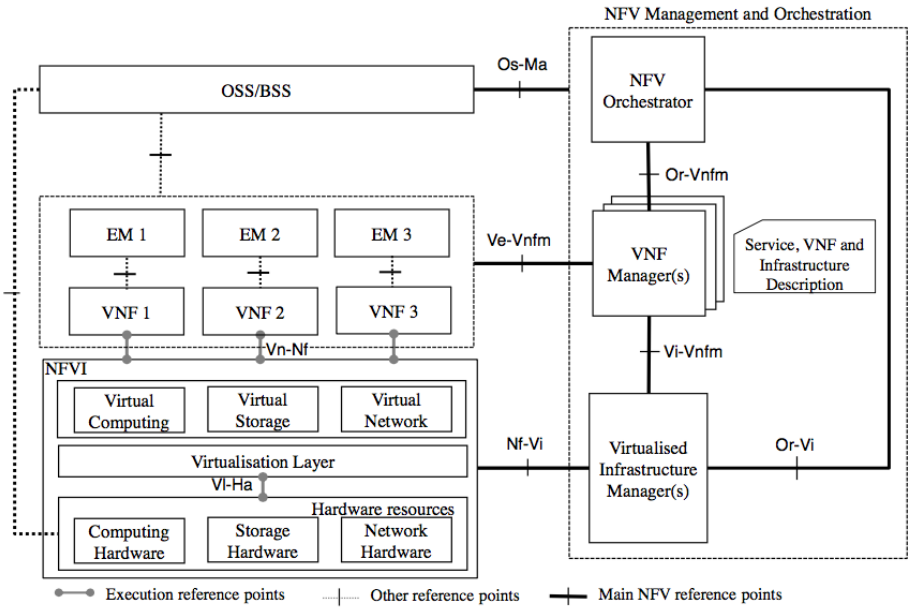

Figure 1.1: Architecture of ETSI NFV-MANO

The Open-Source MANO (OSM) orchestration platform has been developed based on ETSI NFV-MANO specifications that can efficiently orchestrate and manage the lifecycle of the network services. It has an integrated cloud platform, OpenStack as a VIM, and supports SDN controllers [13].

The e2e network slice lifecycle management (LCM) is a very crucial and still challenging task. 3GPP has divided the network slice LCM into four major phases: commissioning, activation, runtime monitoring and operation, and decommissioning of network slices as illustrated in Figure 1.2 [16] [17]. The design and preparation of the network slice is achieved in the network slice commissioning phase. In addition, the network slice template is prepared in the first phase, which contains all the information about services, resources, topology, configurations, resource's

location, etc. After that, the created slice template is implemented over the infrastructure to activate the resources. In the slice runtime operation and monitoring phase, the activated resources are monitored and ensure service guarantee [18]. On the other hand, the activated resources are deleted and released as per requirements specified in the slice template or service level agreement (SLA) in the slice decommissioning phase [14] [19] [17]. Hence, managing and orchestrating multi-domain network slices is vital, and a well-designed platform is needed to manage the e2e network slices in an automated fashion.

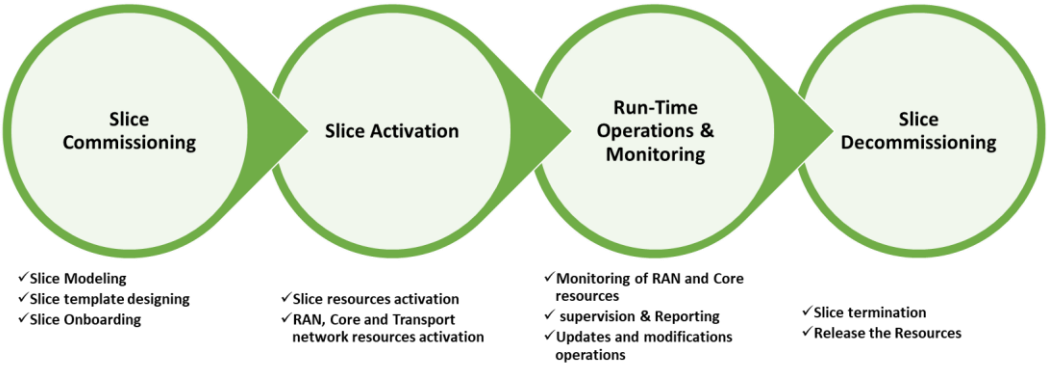

Figure 1.2: phases of network slice LCM

IETF has introduced another innovative technology for the automation and management of network resources named Intent-based networking (IBN) [20]. IBN works on the concept of intents, where users or NOs need to input higher-level abstract requirements, and the system itself translates them into policies and deploys them over the infrastructure [21]. It follows a closed-loop mechanism to perform service orchestration and management. It creates, translates, configures, deploys, and updates the resource in an automated fashion. Hence, this research has introduced an IBN platform for the automation and orchestration of network slice resources. It is a closed-loop

mechanism that can automatically design, commission, activate, monitor, and decommission the network slices.

3GPP has introduced an innovative network data analytics function (NWDAF) in 5G service-based architecture (SBA) to provide intelligence into the network. It collects the data from control and user plane NFs: access and mobility management function (AMF), network slice selection function (NSSF), user plane function (UPF), etc., and performs analytics by using various Artificial intelligence (AI) and Machine Learning (ML techniques. These AI, and ML algorithms are trained on historical network data and perform recommendation, prediction, and detection tasks [2].

Hence, the Intent-based networking and ML-assisted data analytics mechanism for autonomous e2e network slice lifecycle management have been developed in this research. It consists of an IBN platform, NWDAF, NFVO OSM, RAN controller, monitoring, and data collection mechanism. IBN platform can orchestrate, control, and manage the network slice instances automatically. It is a one-touch approach where the user needs to input QoS requirements into abstract form, and the system itself performs all the operations for deploying the resources. It receives the QoS requirements from users, automatically converts them into network policies, and deploys them over the infrastructure.

On the other side, we have integrated 3GPP NWDAF functionalities into the IBN platform for providing intelligent and proactive control and management of network slice resources. The implemented NWDAF is divided into three separate data analytics function (DAF) such as core-DAF (C-DAF), edge-DAF (E-DAF), and RAN-DAF (R-DAF) for each domain. So, for C-DAF, we have proposed a novel hybrid stacking ensemble learning (HSTEL) model for network resource utilization prediction by combining gradient boosting machine (GBM), Catboost, and XGBoost

ML models. On the other side, hybrid EL models have been developed by combining random forest (RF), Catboost, and XGBoost models for anomaly detection use cases. These hybrid models show promising performance in comparison to individual models and state-of-the-art models. Moreover, the prediction results of the NWDAF models will be used by the IBN intelligent decision engine to decide to scale-up/scale down the network resources, and attack detection and mitigation.

1.1. Research Problems and Objectives

The primary aim of the 5G network is to accommodate a wide variety of innovative services that have differentiated QoS requirements. So, it is a highly error-prone and time-consuming process to generate manual configurations for every service. Additionally, it also needed adequate human intervention, manual work, and expertise. Besides, the manual allocation of resources in a multi-domain environment for establishing an e2e network slice is not optimal. The automatic deployment of network resources over the RAN and core domain is still challenging because each domain requires specific configurations. So, a well-designed solution is required that generates multi-domain network configurations for the activation of e2e slices as per QoS requirements. On the other side, AI and ML approaches are needed for proactive management of network resources. In this aspect, dynamic scaling of the cloud resources is another critical issue while managing the cloud resources. It causes to degrade the QoS in case of resource overloading, and in under-utilization cases, it wastes the cloud resources. Hence, autoscaling of cloud resources is vital for reducing the cost and guaranteeing the customers' QoS requirements. So, an AI and ML-based accurate estimation of future network resource utilization to perform autoscaling is needed. However, it is tough to perform accurate resource utilization prediction due to the continuously changing nature of resource usage. By predicting accurate network usage can improve the

operational cost and efficiency of the cloud. Moreover, the attack detection and mitigation from the network to avoid performance degradation is still a challenging task.

To overcome the challenges associated with the above-mentioned research problems, following are the major contributions and objectives of this research:

- An IBN-based closed-loop mechanism is developed to automate the slice policy generation and slice resource orchestration procedure for a multi-domain environment.
- It follows a one-touch approach to design, activate, update, and delete e2e network slices and eliminates traditional manual practices.
- It follows 3GPP and IETF standards to accomplish network slice LCM.
- It has an integrated monitoring mechanism for the core, edge, and RAN domains to monitor network resources continuously.
- The IBN standards follow proactive resource automation, management, and control by using AI techniques. Hence, we integrate the 3GPP NWDAF mechanism with the IBN platform to perform proactive slice resource control and management.
- Hybrid ensemble learning-based NWDAF inside the IBN platform performs core resource utilization prediction, QoS assurance, and auto-scaling of cloud resources.
- It also has a hybrid ensemble learning model for anomaly detection and mitigation from the system.
- NWDAF makes the IBN platform an intelligent orchestration platform for managing the lifecycle of network slice resources proactively and dynamically. It can update slice resources in case of failure or requirements changes.

1.2. Thesis Organization

The main chapters of this dissertation are structured as depicted in Figure 1.3. The details are as follows:

- Chapter 2 explains the literature on e2e network slicing and lifecycle management, management and orchestration approaches, standardization bodies working towards network automation, and industrial solutions for automating the network. In addition, this section also discussed the existing ML approaches related to the NWDAF and network resource utilization prediction and anomaly detection and mitigation from the networks.
- Chapter 3 explains the proposed mechanism for the automation and management of e2e network slicing. This chapter presents the proposed IBN system for e2e network slice lifecycle management. Moreover, the proposed Hybrid ensemble learning-based network data analytics mechanism has been presented. In addition, the newly developed hybrid stacking ensemble learning-based model for network resource utilization prediction and a hybrid model for anomaly detection and mitigation from the system have been well discussed.
- Chapter 4 highlights the experimental results and discussion regarding the implemented system. Furthermore, the e2e network slicing and ML models results have been presented and discussed in this chapter. Also, the achieved results are compared with other relevant studies for the evaluation of the proposed mechanism.

- Chapter 5 concludes the thesis by summarizing this research and discussing future research directions.

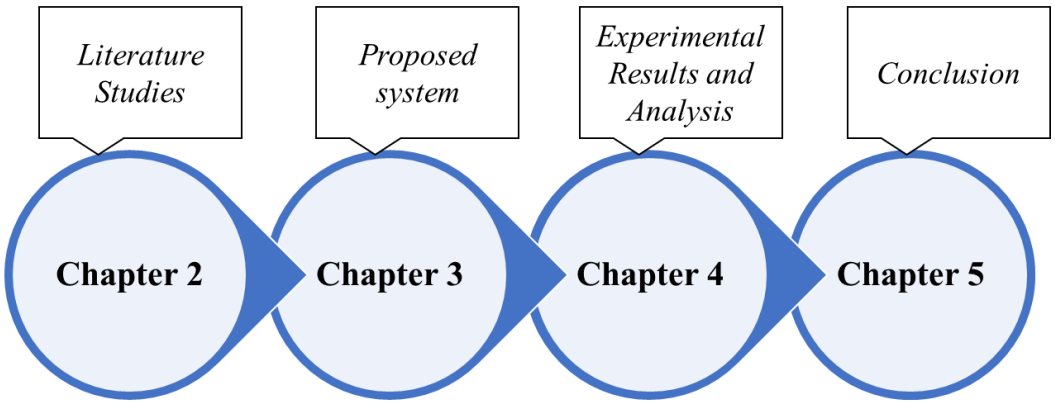

Figure 1.3: Thesis structure

Chapter 2

Related Work

This section explains the detail of previous works related to network slicing, 5G networks, service orchestration and management, slice LCM, and AI /ML approaches for the automation of future networks. It also includes the industrial solution for the automating the networks and highlights a direction on how to reduce or avoid the limitations of the existing mechanisms.

2.1. E2E Network Slice Orchestration and Management

To provide differentiated services to 5G consumers, the automation and management of the e2e network is an essential activity for mobile network operators (MNOs) [33]. Several mobile network standardization bodies have been defined the specifications for e2e network slice

automation and management, such as European Telecommunication standards Institute (ETSI), Internet Engineering Task Force (IETF), 3GPP, Fifth Generation Partnership Project (5GPP), ITU, Next Generation Mobile Network (NGMN), etc. As aforementioned, network slicing is divided into three major types: eMBB, URLLC, and mMTC. The eMBB type of service needs reliable broadband connectivity and high speed. On the other side, URLLC requires ultra-reliable low latency type of communication, and mMTC requires seamless connectivity for many devices and smart industries [1], [2], [10], [19], [34] [35]. Figure 2.1 illustrates main categories of network slicing in the 5G network.

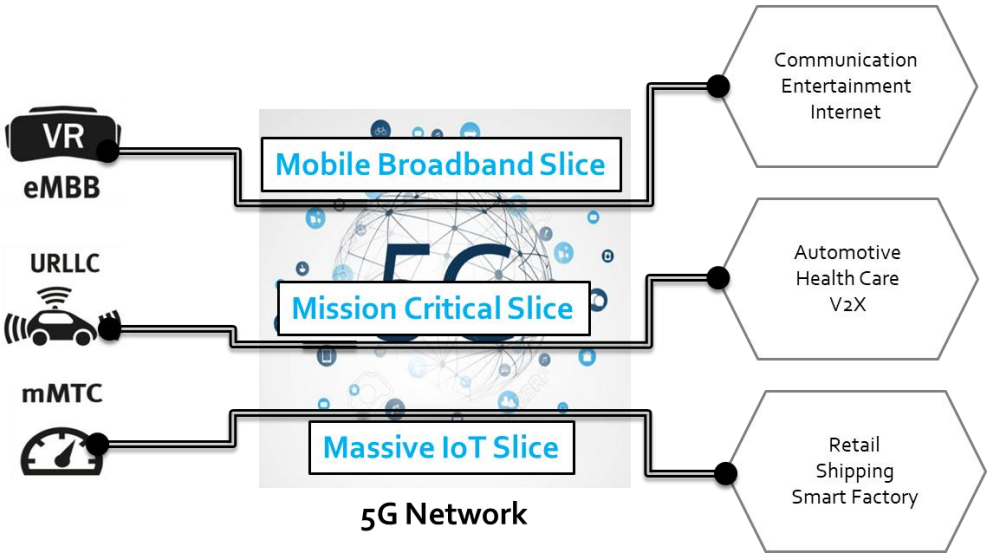

Figure 2.1: 5G network slice categories

Several open-source orchestration platforms were implemented based on the 3GPP and ETSI standards. These orchestration platform supports network slicing and automation. Some well-known orchestration platforms are Tacker, open network automation platform (ONAP), Open Network Foundation (ONF), COMEC, JOX, M-CORD, SONATA, OPNFV, 5G NORMA, Cloudify, OpenBaton, OpenStack HEAT, and Open-O [36]–[38] [39]. The primary aim of these

platforms is to enable programmability to automate the network resources deployment over the infrastructure [40] [33] . The network administrators define the policy configurations for the deployment of the resources. The OpenStack platforms is used to deploy the VNFs, and SDN-based controllers are used for chaining the VNFs. So, these existing orchestration platforms require specific and complex network configurations for the activation of resources.

Furthermore, ETSI has also introduced a Zero-Touch service management system (ZSM) for the complete automation of the network. It is an entirely closed-loop system that does not need any human involvement when in execution mode. It considers different Artificial Intelligence (AI), Machine Learning (ML), and big data approaches for proactively managing the network [41]. It uses ML models to learn the user traffic patterns from the network and performs future predictions. It performs data analytics and extracts the trends, patterns, behavior from the network. Based on the predictions of the ML models, ZSM can proactively prepare the resources and performs autoscaling of the VNFs resources. It can manage physical network functions (PNFs), VNFs, and physical infrastructure.

The 3GPP has also introduced an architecture for the management and orchestration of the service-oriented 5G networks. This system consists of Communication Service Management Function (CSMF), Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF) [14]. The CSMF is acts as a central management entity for the management and deployment of network slices. It is responsible for creating the network slices requests and sends them to NSMF for further operations. The MNOs used CSMF functions to plan, design, and activate the network slices. on the other hand, NSMF translates the slice requirements and generates domain-specific configurations. Further, those configurations are forwards to NSSMF for the deployment of the network slice instances. Each domain has separate NSMF, e.g.,

RAN, core, edge. Also, NSSMF ultimately manages each slice instance. Moreover, the CSMF receives network slice requests with QoS requirements from the customers and forwards those requests to NSMF. NSMF converts slice QoS into policies and activates the resources with the cooperation of NSSMF. The slice instances are appropriately monitored, and CSMF performs autoscaling of the network resources whenever needed [15]. Figure 2.2 depicts the management and orchestration framework to perform e2e network slicing.

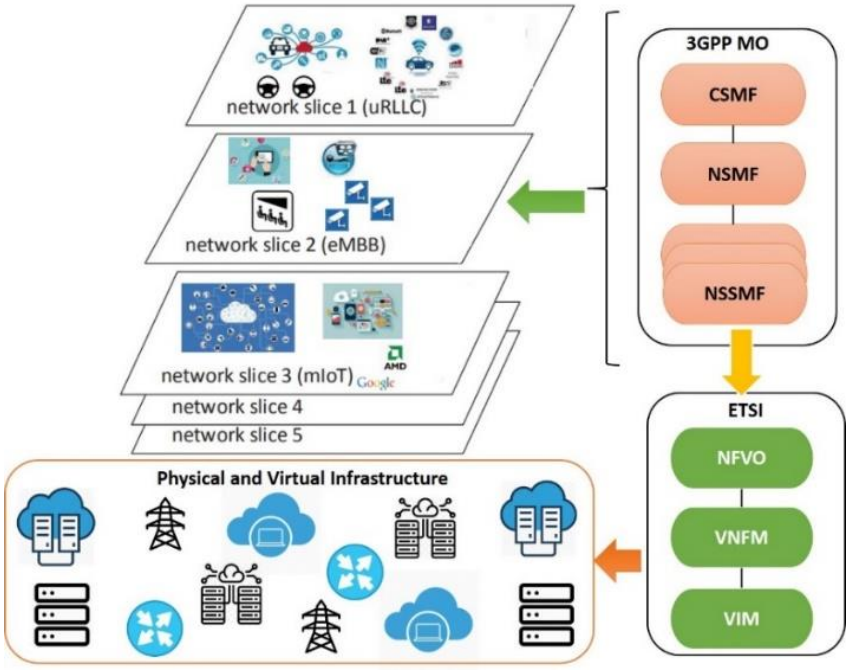

Figure 2.2: Integration of ETSI and 3GPP technologies for management and orchestration of e2e network slicing

The framework of slice management and orchestration was proposed to abstract the instantiation of end-to-end network slices. A chain of network functions in the framework is involved physically and virtually. The proposed plug-in-based SliMANO system appeals to network resources and interacts among network orchestration entities for an end-to-end slice performance. The range of these entities comes from MANO, SDN, and RAN controllers. The

implementation of a proof-of-concept prototype is evaluated. Results showed the delay increment concerning operations of instantiation and deletion by comparing the recent network slicing feature (NetSlice) of the OSM. Moreover, the delay in results corresponds chiefly to SliMANO, an entity external to the orchestrator itself. This SliMANO also drives beyond the MANO domain, and the interaction with SDN and RAN controllers islallowed [42].

The discussion of network slicing in transport and core networks was presented. Additionally, the RAN domain is extended, and all these domains are specified together as an end-to-end (E2E) NS system. Numerical simulations were conducted to prove the benefit of NS in RAN with a two-level resource allocation scheme. Afterward, the application of both hardware and software for the E2E NS system was established. The performance displayed good granularity, slice creation, deletion, and scheme adjustment in sub-minute time used in the network operation [43].

In this cited work, the three-layer technology-agnostic architecture for the slice service layer was firstly proposed in the life cycle management. The NAS was then defined to simplify the complex process of network slice. Later, the LTE network was applied to provide feasible architecture by filling the gap of existing technologies such as cloud, SDN, and NFV [9].

According to the goals and requirements of customers and industries, the E2E NS system was classified into Network Slice Design and Multi-Domain Orchestrator in terms of design time and runtime. These two components react differently based on different players and require interfaces and data structures in the system [44].

There is a high diversity in NGWNs by integrating communications with different scales, technologies, and network resources. Moreover, requirements are still left in using automatic

applications, e.g., machine-to-machine communications, factory automation, etc., concerning reliability and latency. Regarding these challenges, AI-based network slicing architecture was proposed for NGWNs by considering existing works and potential directions for the future [45].

For network operators, slicing in networks ensure that relevant services could be provided effectively to end-users. Considering improvement in OPEX savings, network efficiency, and time-to-market acceleration, ONAP platform was used to perform the slice management. Furthermore, model simulations, orchestration and E2E network slice enforcement were involved in RAN, core, and transportation networks by considering a private network as a case study [15].

In the field of Internet of Things, 5G networks is emerging era by connecting between several devices and heterogeneous sets that meet the requirements of network quality. There is big challenge for operators to overcome complex network services demanding from different customer areas (for e.g., healthcare, energy, automotive, etc.). Consequently, the conception and distribution parts of network automated management were mainly presented, especially highlighting two parts. The first part is to capture SLA based on the requirements of customer network by negotiating the refinement process. The second part is to deploy SLA instructions by forming network resource orchestration into sequence [46].

After revising data preprocessing and analytics of existing technologies, the complete framework for SLA was proposed to implement big-data-driven dynamic slicing resource. The framework was implemented by developing low-complexity slice traffic predictions, allocating resource models, and enforcing SLA through deep learning [47].

DRL was investigated to provide better performance and efficient coat services for network slicing. Using alternative and tendency actions in the DRL process could be a promising solution

in terms of interaction with the environment. Basic concept of DRL was described and applied it for typical resource management in the case of network slicing including radio resources and priority-based core networks. The generated results passing through extensive simulations of DRL usage over different schemes were demonstrated, and then provided possible challenges for various perceptive of DRL application [48].

2.2. Standardization and Industrial Progress towards Network Automation

Nowadays, network automation is a very hot topic for industrial as well as academic researchers. The procedure of planning, designing, activating, monitoring, and optimizing the network resources and services in an automated manner is named network automation. Primarily, the transition of manual work to automated way through softwarization for managing network resources starting from planning to the operational mood efficiently and reliably [49].

2.2.1. Standardized Bodies for Network Automation

The operational and management complexity of 5G and future networks shifted the industries toward the closed-loop network service automation and management mechanism. For this, ESTI has introduced a new ZSM framework for 100% automation and management of future networks. It aims to automate the complete operational and management tasks of the network. Moreover, using AI technologies enables ZSM to manage the network, reducing human errors, optimization, and operating costs. Figure 2.3 highlights the design and architecture of ETSI ZSM for automating networks.

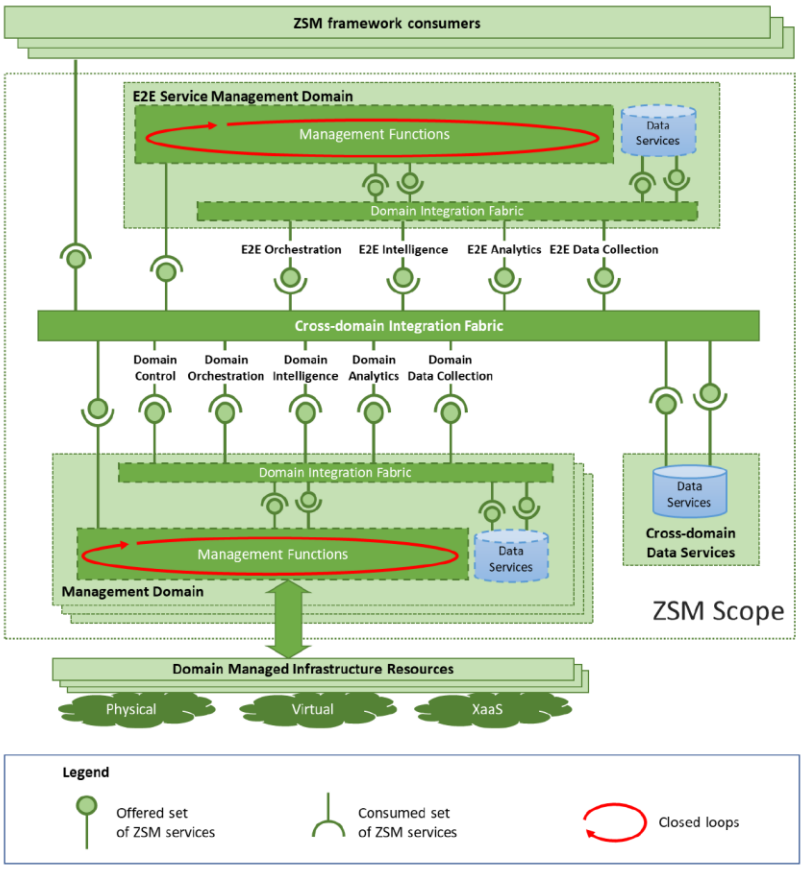

Figure 2.3: ETSI ZSM architecture for network automation [41]

The AI and ML technologies enabled advanced network automation tools to perform data analytics based on historical network data and automatically control several networks' operations. These operations are as follows: check network behaviours, perform prediction, classify the traffic, check user patterns, present recommendations, issue update policies, and many more [50]. Advanced tools such as ZSM, IBN, and ONAP can take proper actions based on AI algorithms analysis autonomously and handle that situation before occurring. So, due to network automation, NOs can reduce operational costs, increase service quality, provide better QoE, ensure better service availability, and reduce human errors.

Following tasks can be handled through network automation solutions.

- The designing, planning, and management operations are achieved through network automation.
- Network data collection from multiple devices such as smart devices, network resources, RAN data, Core cloud, transport data, management data, network topology information, real-time services, applications data, etc.
- The use of AI and ML to perform data analytics provides insights into current network behavior, resource utilization behavior, and future requirements for proactive management.
- Conformity of deployed configuration to validate the running operations of connected network devices
- Automatic reconfigurable in case of failure that includes troubleshooting and repairing of resources
- Should perform reporting, alarms, and alerts to indicate the network status
- It will be able to monitor multidomain network resources to guarantee QoS for the customers.
- Strong network security is also a concern for an automated solution.

As the current and future networks are ultimately moving towards cloud computing, the success of serval enterprises, their customers, and applications are highly dependent on the network. So, it is expected that the network should be reliable and automated for service providers. Currently, cloud network reliability, agility, and automation are major challenges for operators and vendors. The automation of the network can control the OPEX and CAPX of the network operators [49] [11] .

Following are the advantages of automating the networks:

- It can reduce the network issues and resolve them spontaneously through a closed-loop approach. The manual operation such as configuration generation errors and human errors are reduced.
- It can reduce the human effort for managing and handling network issues.
- Automation can cause to reduce the operational and management costs of the network to an extent such as fewer human resources are required to provision, configure, and manage the network and services. So, fewer human resources are needed to perform management and operational tasks.
- The service providers can design services for some special events across various geographical locations in advance and ensure high-quality services.
- It can reduce the down-time of the network because the network can detect errors, report failure, and overcome that failure automatically.
- Ensure more control and visibility to the network and proactively manage network

2.2.1.1. Intent-based Networking

Another standardization body IETF has been introduced the specifications for automating the network through Intent-based networking IBN [20]. IBN is an intelligent system that enables the future network to be self-configured, self-planned, self-assured, and self-healing. Many industrial organizations such as Cisco, Huawei, APSTRA, etc., also adopted IBN technology for the automation and orchestration of future networks [22] [23] [24] [25] [26]. Figure 2.4 illustrates the concept of IBN and how it works. Firstly, users need to input business intents through the system's dashboard; Secondly, received service requirements are translated into policies through the translation engine. Thirdly, translated policies are deployed over the physical and virtual

infrastructure. Finally, deployed services are appropriately monitored to assure the services and updates in case of failure.

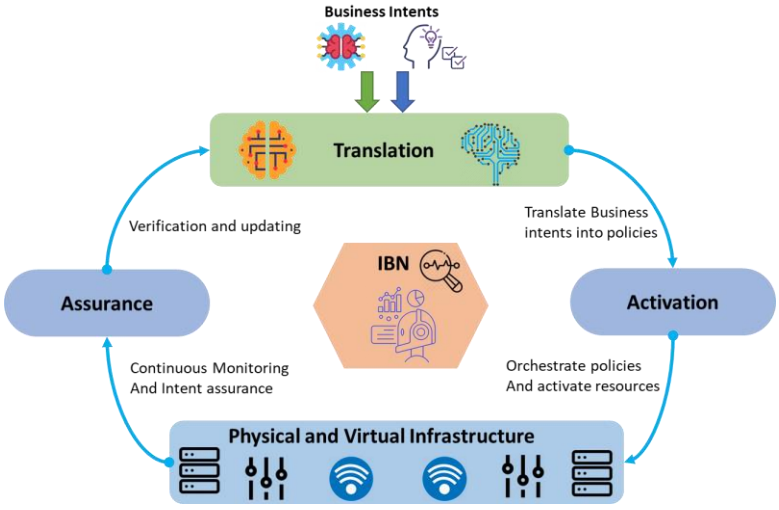

Figure 2.4: Intent-based Networking abstract design and architecture

According to the IETRF IBN for closed-loop automation, a proper IBN system (IBNS) contains two fundamental qualities that make it more than just a fancy configuration management platform: intent fulfillment and intent assurance. IBNS can recognize and generate user intents, translate them into policies, refine from NOs to validate intent, configure the resources, monitor the deployed resources, analyze network status, validate the QoS, and report to the operator for service assurance. Figure 2.5 depicts the process of intent lifecycle managed through IBN solution.

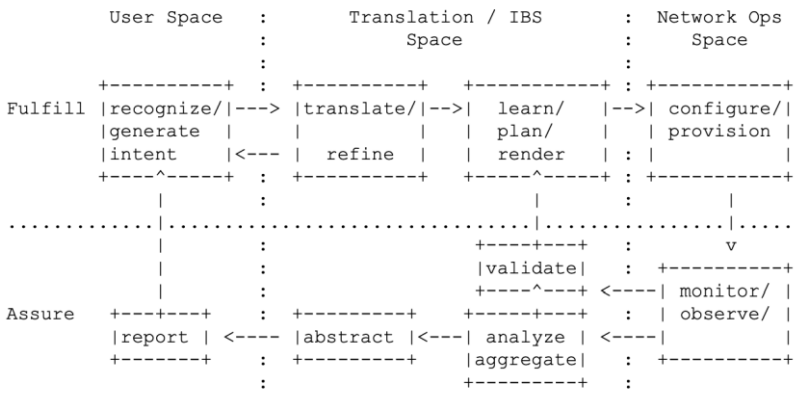

Figure 2.5: phases of IETF defined phased for intent lifecycle management

2.2.1.2. Network Data Analytics Function

3GPP has introduced NWDAF in 5G SBA architecture for providing intelligence into the future network. Using ML models, NWDAF can achieve network slice load prediction, resource utilization prediction, Anamoly detection and mitigation from the system, mobility prediction, user pattern prediction, and many more [27] [28]. So, it will help NOs to manage the network resources and assure service SLA proactively. So, NWDAF seems an excellent option for network resource utilization and Anamoly detection and mitigation from the system. So, in our approach, we have designed the NWDAF mechanism, which has two hybrid ensemble learning model for network resource utilization prediction and Anamoly detection from the system. Figure 2.6 depicts the design of NWDAF in 5G architecture.

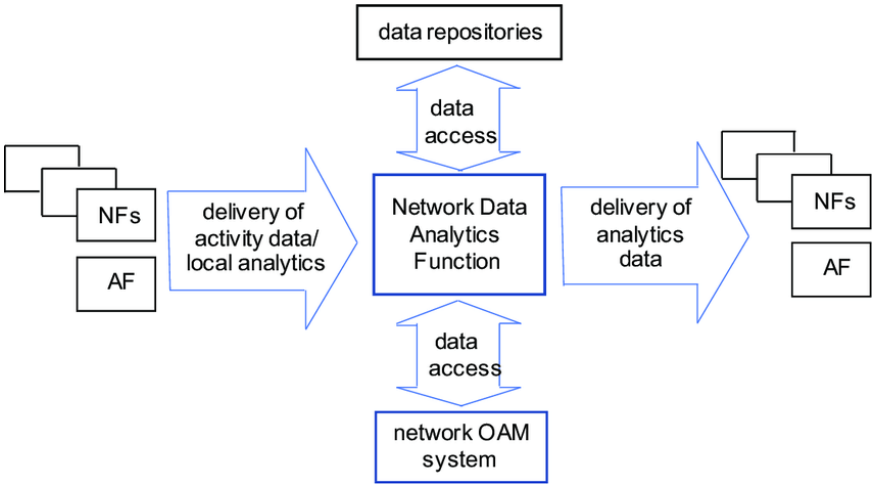

Figure 2.6: NWDAF workflow

2.2.2. Industrial Progress and Solutions for Automating the Network

Several well-known industrial organizations have developed enhanced approaches for the automation of networks including CISCO, Huawei, Apstra and Juniper etc.

Network automation is a foundational component of Experience First Networking, as it simplifies planning, design, and operations while improving customer experience. It eliminates the impact of human error and reduces customer churn. Staying ahead of your customers' business and personal requirements requires automation solutions that can support your entire network, from L2 to L7, through all stage of your services lifecycle, from day 0 through day 2. Juniper provides automation for their all products, and they offer a package of solutions for network engineering, operations, and DevOps teams to take advantages from the network automation. For achieving better QoE for customers, Juniper's have developed a closed-loop solution namely Paragon Automation to provide service assurance [51] [52].

Another well-known industrial organization Apstra have also developed a solution for automating the network. This solution can be achieved through following three technologies:

Firstly, they have developed IBN platform for service designing, translation of policies, self-reporting, and validation. Secondly, Single Source of Truth (SSOT), closed-loop network automation and data analytics mechanism achieved through a graph datastore. Thirdly, they provide complete openness and vendor independence. They have IBN analytics mechanism that allows to insight the status of complete infra and collects and store more important data. Apstra has Apstra operating system (AOS) that allows to design, build, deploy, and validate network on real-time. It does it by using IBN core concepts by having a central AOS. Figure 2.7 illustrates the architecture of AOS system [24] [53].

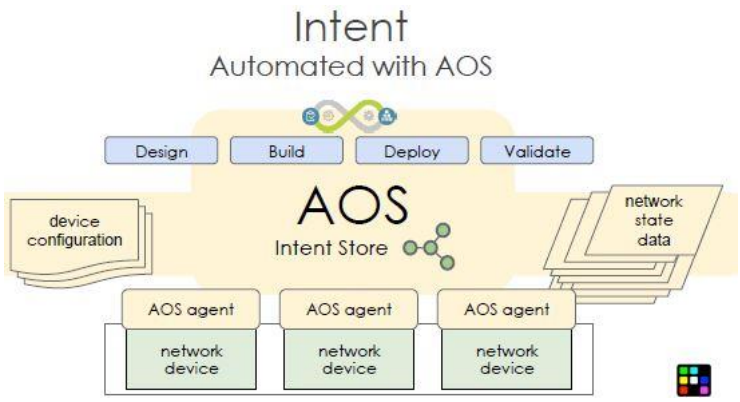

Figure 2.7: Architecture of APSTRA AOS

It may sound counterintuitive, but it's all made possible with intent. Intent means, essentially, what you are trying to achieve. Apstra Intent-Based Analytics works by allowing you to use quality information (patterns you know or expect), made available to you through intent (what you are trying to achieve) and reference design (how you are going to achieve it). Intent gives you the ability to gather the right knowledge, rather than all the knowledge, from user infrastructure. The right knowledge leads to actionable insights. Apstra's Intent-Based Analytics uses intent to find the right knowledge and data to identify conditions with significant semantic value (in other words,

insights). The insights you gain from Apstra's Intent-Based Analytics helps NOs to control infrastructure efficiently [53].

The Benefits of Apstra Intent-Based Analytics is different from traditional solutions:

- **Simple and clear insights**: identifies significant conditions and situations to watch and eliminates noise in real time using a "simple pane of glass."
- **Storage and processing cost savings:** enables storage and processing cost savings by extracting more information while collecting and storing less data.
- **Automated troubleshooting:** Empowers you to automate effective, complex, context-rich troubleshooting workflows, saving time and money.
- **Zero-Touch Maintenance:** Provides zero-cost and zero touch maintenance when changes occur.
- **No Integration Necessary:** Increased flexibility and cost savings by eliminating high-cost, fragile data-processing pipeline integration efforts.
- **Meticulous Accuracy:** More accurate than ML/AI approaches, making you more effective and agile while saving time and money.

RADCOM ACE (automated, containerized, e2e) has proposed a closed-loop and automated solution that works with Kubernetes to provide service assurance for 5G networks. It identifies each network slice instance, generates performance indicators, and forwards them to control plane NFs such as NSSF and PCF. RADCOM also has an ICON platform that provides service assurance to deliver Intelligent, Container-based, On-demand network analysis from the RAN to the Core domains. RADCOM ICON has implemented some capabilities of NWDAF. It allows consumers to subscribe to the NWDAF to receive performance indicators and exceed threshold notification whenever needed. The AI and ML algorithms inside the RADCOM system collect the data from

various domains, perform real-time data analytics, and forward results to NFs and NOs for the proactive management of the network. Also, network resources are continuously monitored to ensure service SLAs [54] .

Sandvine has also introduced the NWDAF framework for the core cloud that follows 3GPP standards. It is enriched with tested ML capabilities and performs intelligent traffic classification superior to other industrial solutions. Sandvine has not only built an NWDAF that meets this minimum standard but has also enriched it with proven traffic classification capabilities and granular, contextual, and precise KPI measurements that can be used to revolutionize the way mobile service providers see and manage their networks [55].

Another popular organization ERICSSON has also introduced a data-driven architecture for the management and automation of the network. It has ML-based data pipelines, exposure and probing mechanisms, and NWDF capabilities through AI and ML. They have the NWDAF for control plane and data plane network functions and MDAF for performing management domain analytics [56].

NWDAF is a critical element in providing automated assurance and analytics, allowing operators to manage elements such as network slicing in a cloud native NFV domain in real-time. A central foundation of VIAVI NITRO Mobile [57] is an open platform with northbound and southbound interfaces aligning with standards and technologies as they evolve. NITRO Mobile would leverage the NWDAF principles to provide granular insight on the 5GC network, services, and customers.

2.3. AI and ML Approaches for Network resource Utilization Prediction and Anomaly Detection and Mitigation

This paper proposed a convolutional neural network (CNN) and long short-term memory (LSTM) models to predict multivariate tasks such as central processing unit, memory, and network usage. Initially, the vector autoregression technique was used to filter linear interdependencies among the multivariate data. The residual data is then provided to the CNN layer to extract complex features of the virtual machine usage components after the LSTM network. The proposed hybrid model applied the scaled polynomial constant unit activation function and compared it with other predictive models. It improved the accuracy performance, showing around 3.8% to 10.9%. The error percentage rate also decreases to approximately 7% to 8.5% compared to the other models. Consequently, the proposed model could improve the central processing unit, memory, disk, and network usage by taking less computation time. We will implement this proposed model in VM energy and data prediction in the cloud data center in future research [58].

Support Vector Regression (SVR) model was considered to forecast the future usage of multi-attribute host resources and handle a non-linear cloud resource workload. Inside SVR, radial basis kernel function and Sequential Minimal Optimization Algorithm (SMOA) were hyper tuned to improve the forecasting accuracy. Moreover, we investigated the multi-attribute cloud resources over a single resource differently from the previous works. Our method employed eight sets of real-world data collected from BitBrain (BB), PlanetLab (PL), and Google Cluster Workload Traces (GCWT). This series of experiments showed one of the practical points of our approach. The generated accuracy increased about 4%-16%, together with the reduction in error percentage at around 8%-60% [59].

In this cited work, the structure and the protocols of NWDAF defined in the 3rd Generation Partnership Project (3GPP) standard documents were proposed. A cell-based synthetic dataset for 5G networks represented by the 3GPP specifications is generated, and some anomalies were added. The classification of these anomalies has been done within each cell, subscriber category, and user equipment. Linear regression (LR), LSTM, and recursive neural networks were then employed for the network load prediction capabilities of NWDAF in terms of minimizing mean absolute error (MAE). Two ML models, such as logistic regression and extreme gradient boosting (EGB), were implemented under the receiver operating characteristics curve in the classification module. The simulation results showed that neural network-based algorithms outperformed LR in network load prediction. However, the tree-based gradient boosting algorithm outperformed logistic regression in anomaly detection. Therefore, these estimations can improve the accuracy performance in the 5G network through NWDAF [27].

They proposed a novel method that can identify the most appropriate model adaptively and automatically and predict data center resource utilization. The classifier was trained based on statistical features of historical resource usage. The suitable prediction model was then decided for given resource utilization observations collected during a specific time interval. The evaluation of our approach was conducted using real datasets and comparing to multiple baseline methods. The results indicated that our model provided better performance than the state-of-the-art approaches, delivering 6% to 27% utilization estimation accuracy [60]. Moreover, an EL based hybrid model was developed for CPU usage prediction in this work [61]. The hybrid model is a combination of multiple lightGBM models. It shows 0.91 R_2 accuracy while performing CPU prediction.

Although artificial intelligence (AI) has been widely used in the area of 5G networks, there is still a lack of standard solutions to build an operational system. This article combined the relevant standard specifications and complemented them with additional building blocks. Their proposed framework used concrete AI-based algorithms serving different purposes toward developing a fully operational system. The results from applying our framework can control, manage, and orchestrate functions, showing the benefits that AI can bring to 5G systems [62].

The constant change in cloud consumption affect the accuracy of forecasting algorithms that become one of challenge in cloud computing [63]. Therefore, RNN was applied for predicting CPU utilization in terms of single time-step and multiple time-steps because it can predict more accurately than tradition approaches for time series problems.

Machine learning and slide window-based forecasting models were developed to scale proactive resource in the cloud [64] . Historical and current utilization data were used to forecast future utilization. The evaluated accuracy was improved by using prediction models and improved sliding window size. It shows 80% prediction accuracy.

The neural network along with self-adaptive differential evolution were implemented for predictions using cloud data [65]. This proposed approach provided better accuracy than usual back-propagation algorithm with respect to measuring RMSE.

Conventional time series models have no ability to handle long-term dependency which appears in cloud usage resource. Consequently, multivariate and bidirectional LSTM models were proposed for usage prediction. The generated results of both models were compared with different fractional-based techniques and outperformed existing models based on Google cluster trace [66].

Concerning challenges and issues in resource scaling and power consumption of cloud computing, LSTM predictive models were trained and tested using three baseline available data on web server logs [67].

An anomaly detection technique was developed by the hybrid model that includes the fuzzy K-means (FKM) algorithm, support vector machines (SVM), and Kalman filter (EKF). The proposed method was performed to verify universally accepted datasets, including DARPA'98 dataset and Knowledge Discovery and Data Mining 1999 (KDD'99) dataset [68] .

This work [69] revealed that the hidden Markov model outperforms all other anomaly detection techniques according to the average anomaly detection rate and FPR. Ghosh et al. [70] also proposed an artificial neural network-based anomaly detection method exploiting several variations in classifying system-call sequences. The performance of the detection rate using DARPA data has been improved.

Network disruptions are sometimes caused by the presence of anomalies in operational networks. As a result, the abnormal changes in network traffic have been analyzed by using several techniques. In the cited work of Jiang et al., [71] the network-wide traffic was analyzed to detect the abnormalities in the transform domain. The problem of anomaly detection was also examined for large-scale communication networks. This work performed the wavelet transform method in integration with the empirical mode decomposition approach to diagnose the anomalies involved in multimedia medical devices and capture the multiscale characteristics of anomalies in high-speed network traffic.

Deep learning (DL) based botnet detection system was proposed for network traffic flows [72]. In the botnet detection framework, the network traffic flows were collected and converted

into connection records and used the DL model to detect attacks from compromised IoT devices. Well-known and recently released benchmark datasets were applied to conduct the experiments to optimize the DL model. From the result comparison, the proposed model outperformed the conventional machine learning (ML) models.

This paper aims to detect the attack traffic by taking the centralized control aspect of SDN [73]. In today's world, various ML techniques are being deployed for detecting malicious traffic in the field of SDN. Despite benefits, there is an open question in selecting relevant features and accurate classifiers for attack detection. The combination of SVM with kernel principal component analysis (KPCA) reducing the dimension of feature vectors and a genetic algorithm (GA) optimizing different SVM parameters was proposed to get better detection accuracy. An improved kernel function (N-RBF) is also applied to reduce the noise caused by feature differences. The proposed model achieved more accurate classification with better generalization than the standalone SVM model. It can also embed within the controller to define security rules and prevent possible attacks by the attackers.

This article [74] presented and reviewed various false data injections by using attack detection methods for CPSs. The controllers of CPSs were grouped as centralized and distributed controllers based on the knowledge of control information. The discussion of existing centralized attack detection approaches revealed four domains: linear time-invariant systems, actuator and sensor attacks, non-linear systems, and systems with noise. Additionally, different decoupling methods were implemented for the distributed attack detection n. Some challenges and future research directions are also provided in the context of attack detection.

2.3.1. Ensemble Learning Approaches

According to existing research, the standalone forecasting model often could not improve accuracy performance if there is a complex structure of input features and non-linear relationship problems. As a result, many researchers have recently been conducted combining two or more algorithms as either a hybrid or an ensemble method to improve forecasting performance. In the case of the ensemble method, there is no limitations for the number of ensemble algorithms, and it can be classified into three learning methods: bagging, stacking, and boosting. The first bagging ensemble method fits many tree-based and machine learning algorithms trained on the same dataset. The final predictions of bagging are generally generated either averaging or voting among all predictions from ML algorithms [29] [30].

Similarly, various different type of ensemble members are trained on the same training data in the stacking ensemble method. In the stacking approach, several classifiers or regressor models are combined through meta-regressor or meta-classifier. The base learner models (level-0) results are used as input to train the meta-regressor model. The meta regressor model learns from the base learner how they make errors and resolve the error in the final prediction result. Nevertheless, unlike the bagging ensemble, stacking uses either a different algorithm or one of the ensemble members to combine final forecasting results. The last ensemble learning, so-called boosting, is significantly different from others because of not combining ensemble members instantaneously. Boosting trains ensemble members sequentially by correcting the predictions of the prior model and then executing the final predictions using the weighted average of models [31] [30][32].

However, taking advantage of the ensemble learning capabilities, we have proposed a novel stacking EL-based model for network resource utilization prediction by combining gradient boosting machine (GBM), Catboost, and XGBoost ML models. On the other side, hybrid EL

models have been developed by combining random forest (RF), Catboost, and XGBoost models for anomaly detection use cases. These hybrid models show promising performance in comparison to individual models and state-of-the-art models.

Chapter 3

Design and Architecture of Ensemble Learning-based Network Resource Utilization Prediction for IBN-enabled Slice LCM

3.1. Introduction

Intent-Based Networking (IBN) has been introduced by IETF, which provides proactive control and management to fulfill customers' differentiated QoS requirements. It receives higher-level user intentions and automatically deploys them over the infrastructure. It is a self-configure, self-assured, self-organized, and self-healing mechanism. Hence, the Intent-based networking and ML-assisted data analytics mechanism for autonomous e2e network slice lifecycle management have

been developed in this research. It consists of an IBN platform, NWDAF, NFVO OSM, RAN controller, monitoring, and data collection mechanism. IBN platform can orchestrate, control, and manage the network slice instances automatically. It is a one-touch approach where the user needs to input QoS requirements into abstract form, and the system itself performs all the operations for deploying the resources. It receives the QoS requirements from users, automatically converts them into network policies, and deploys them over the infrastructure.

On the other side, we have integrated 3GPP NWDAF functionalities into the IBN platform for providing intelligent and proactive control and management of network slice resources. The implemented NWDAF is divided into three separate data analytics function (DAF) such as core-DAF (C-DAF), edge-DAF (E-DAF), and RAN-DAF (R-DAF) for each domain. So, for C-DAF, we have used a hybrid STEL ML approach for future VNF resource utilization prediction and another ensemble ML model for anomaly detection. Hence, the prediction results of the NWDAF models will be used by the IBN intelligent decision engine to decide to scale-up/scale down the network resources, and attack detection and mitigation.

Figure 3.1 illustrates the abstract architecture and design of the proposed ensemble-Learning based network data analytics for e2e slice orchestration and management through IBN platform. It comprises an IBN platform, NWDAF, NFVO orchestrators, a RAN controller, and a monitoring mechanism. The step-by-step explanation of each component is illustrated below.

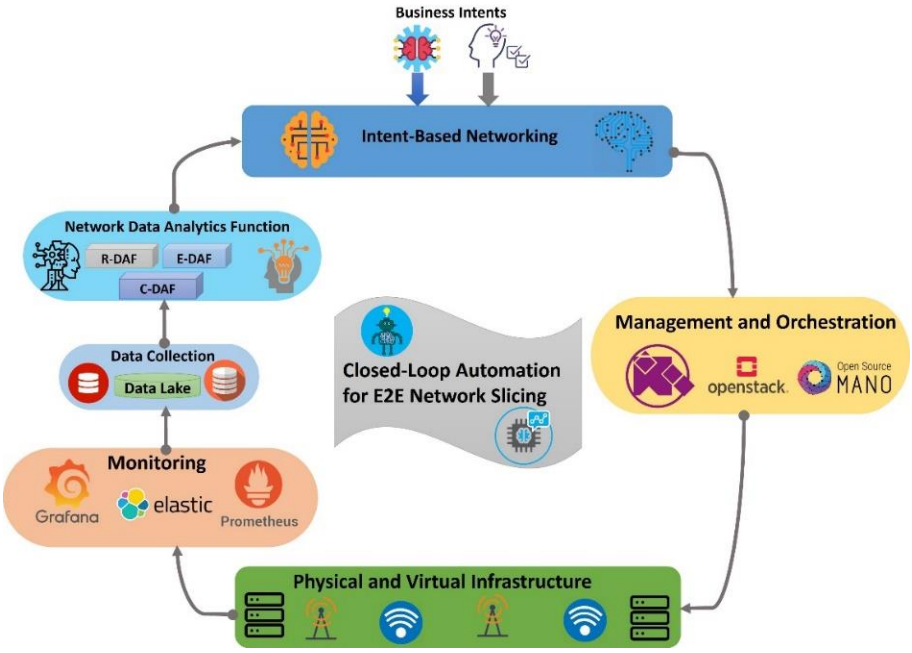

Figure 3.1: Abstract architecture of proposed system which contains IBN system, management and orchestration module, infrastructure, monitoring mechanism, data collection module, and NWDAF

3.2. Intent-based Networking (IBN) platform for e2e Network Slice lifecycle Management

The IBN platform is further divided into IBN dashboard, intent translation, e2e design and information repository, network policy generation module, ML-based NWDAF, and intelligent decision engine module. The IBN platform is a closed-loop mechanism for orchestrating, controlling, and managing e2e network slices in a multi-domain environment. It can commission, activate, monitor, and delete the network slice in an automated manner. The details of slice lifecycle management are explained below.

3.2.1. Slice Instantiation or Commissioning

The dashboard of the IBN platform is used to input the user intents or network slice requirements into abstract form. The intent translation module translates the abstract inputted intents into resource requirements. e2e design and information repository is a knowledge base or a rich database of IBN platform which has network topology information, underlying physical and virtual resource information, deployed policies information, and API's information. The intent translation engine can translate higher-level abstract requirements into resource information and designs a VNF forwarding graph (VNFFG) with e2e design and information repository correspondence. The created VNFFG is further forwarded to the policy generation module, where we have separate orchestrator-dependent policy generators because each orchestrator and network controller accepts policies in a different format. For example, OSM accepts the policies into JSON string format, M-CORD platform accepts into TOSCA format, and RAN controller accepts into JSON template. So, the RAN, edge, and core policy generator modules convert the received configurations into domain-specific policies and send those policies templates into the underlying NFVO and RAN controller.

Furthermore, OSM NFVO deploys those policies by activating the resources over the core and edge infrastructure with the help of VIM OpenStack. On the other side, the RAN controller deploys policies over the RAN domain. The RAN policy template contains activated core and edge VNFs such as dedicated vMME information for a specific slice. The detailed architecture of the proposed mechanism is depicted in Figure 3.2. The internal details of our implemented testbed for network slicing and working of each component is well explained in these references [4] [75]–[79]. Once the slice resources are orchestrated and connected, the monitoring mechanism

continuously monitors all domains' slice resources. It collects the logs from each domain and stores them into a data storage module.

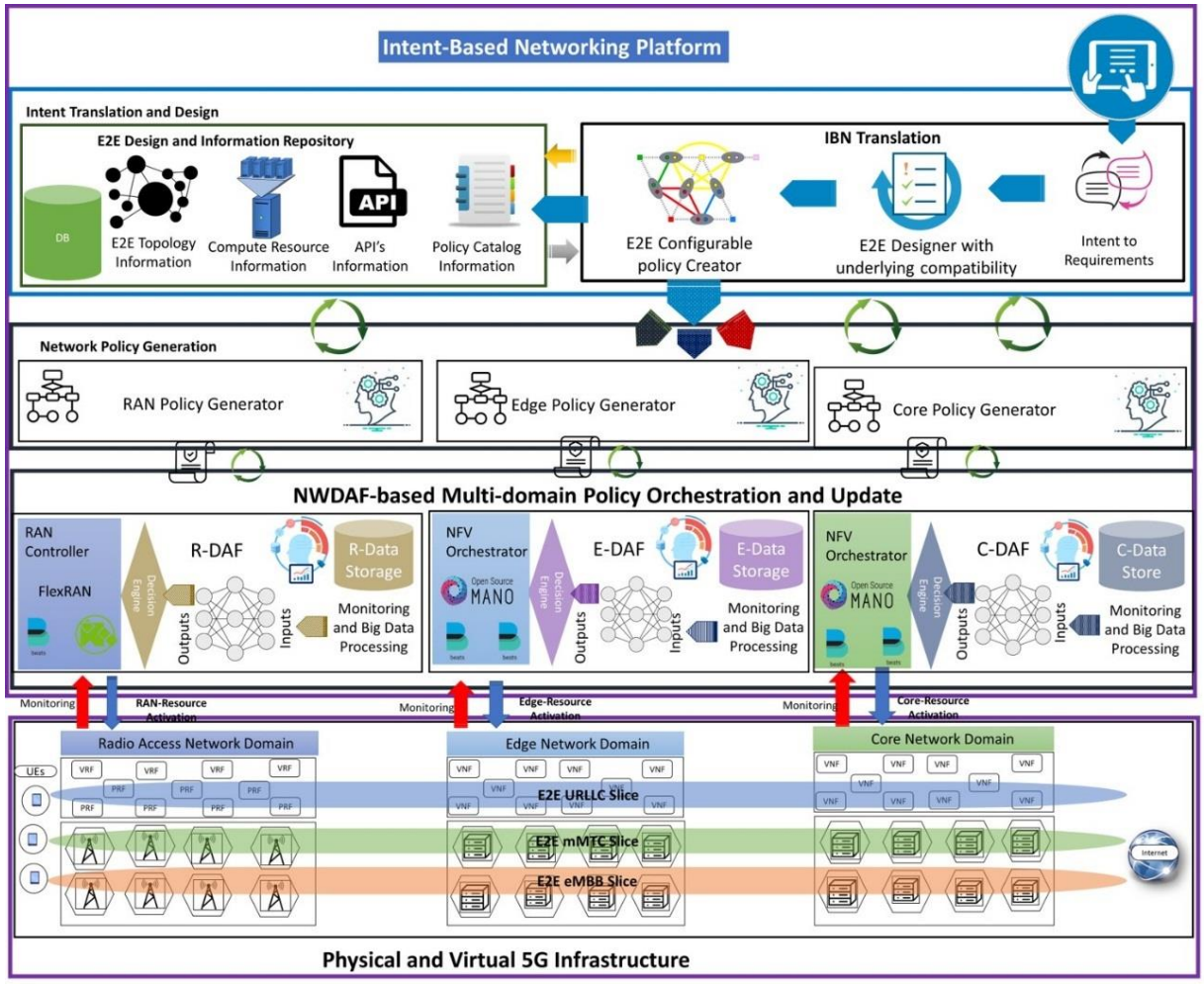

Figure 3.2: Detailed architecture of proposed mechanism of network data analytics for IBN enabled slice lifecycle management

3.2.2. Slice Activation

As mentioned earlier, the IBN platform converts the service requirements into network policies for domain-specific orchestrators and controllers. To activate the network slice over a multi-domain environment, we are using multiple NFVOs and network controllers: M-CORD and OSM

platform are used as NFVOs, SDN-based FlexRAN controller for RAN domain. The procedure of activating a slice over the infrastructure using various components is discussed below.

3.2.2.1. NFV- Orchestrator OSM for the Deployment of core VNFs

OSM is an open-source NFV-based orchestration platform developed by ETSI. It automates e2e service instantiation and management process. It not only supports the management and orchestration of VNF but also physical network functions PNF as well as hybrid functions. This is a widely adopted orchestration platform for VNFs deployment and management. It enables the NOs to orchestrate services over the RAN, transport, and core domain. OSM developed on ETSI MANO specifications has several modules: NFVO orchestrator, VNFM, VIM, and WIM (wide infrastructure manager). OSM used the OpenStack cloud platform as a VIM [80].

Figure 3.3 illustrates the OSM architecture for the management and orchestration of VNFs. It supports multiple VIMs to handle multi-domain network resources. In addition, OSM has a GUI dashboard for deploying, managing, and monitoring the e2e services over the infrastructure. It also supports the deployment of e2e network slicing in the core and RAN domains. So, we have used the OSM platform to deploy core NFs such as EPC components (vHSS, vMME, VSPGWC, vSPGWU). Besides, we have a policy configurator for OSM in the IBN platform, which converts the slice requirements into JSON string format or YAML format because OSM accepts the policies in this format. Thereby, the OSM policy configurator forwards the created policy template to the OSM through the Rest interface. Also, OSM deploys the NFs over the infrastructure with the help of OpenStack and VNFM. So, in this way, OSM deploys the core EPC NFs automatically.

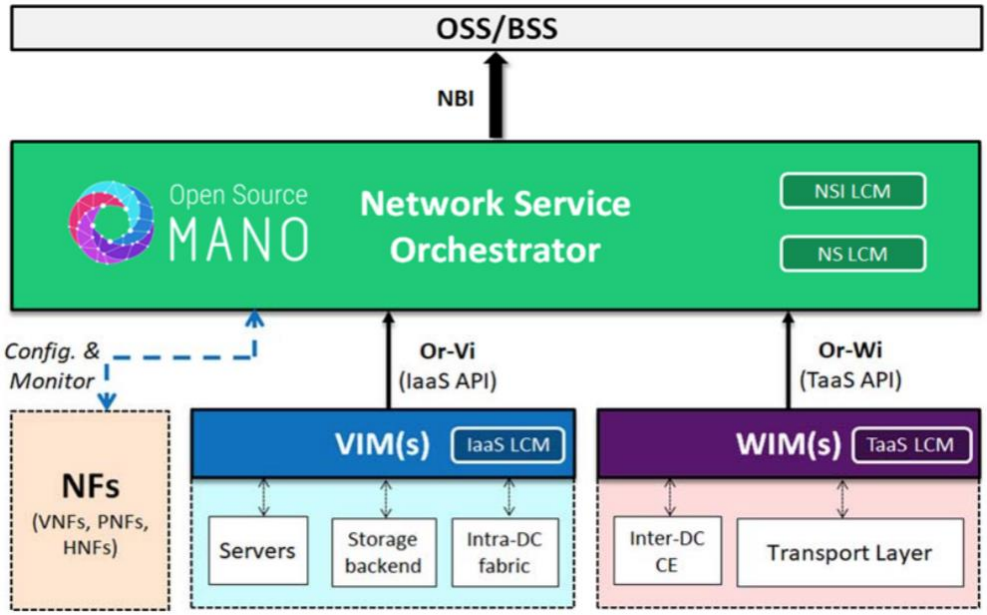

Figure 3.3: Architecture of OSM platform for the deployment of VNFs

3.2.2.2. RAN Controller

We have used the FlexRAN controller for slicing the RAN resources as per user requirements. FlexRAN is an SDN-enabled RAN controller that provides a highly programmable environment and decouples the control plane from the data plane. Its programmability support enables the users to develop the applications on top of it. It can also control and manage multiple base stations efficiently. It has two main components: FlexRAN control plane and agent API. The FlexRAN master controller resides inside the FlexRAN control plane, which interacts and controls the FlexRAN agents [81]. On the other side, the FlexRAN agent acts as a local controller when interacting with the other agents and the master controller. It can control one eNodeB at a time. The FlexRAN agent API is a southbound interface that can decouple the control plane of FlexRAN from the eNodeB data plane. The FlexRAN protocols are used for the communications between a FlexRAN master-controller and agents. Control and monitoring applications are developed on top

of the RAN controller and communicate with the master controller through the northbound interface. These applications are used for the automatic management, modifications, and control of the RAN resources. FlexRAN controller in our network slicing system is responsible for slicing the RAN resources [82]. Figure 3.4 highlights the components of FlexRAN controller. Due to this, our RAN slice template generator converts the higher-level slice configurations into the JSON format. Moreover, it receives the RAN slice template generated by the RAN slice template generator and deploys the slice over the eNB as per requirements specified in the slice template, e.g., 20 MB/s bandwidth. We made some modifications by developing management and control applications on top of controller to deploy QoS aware network slices. We have developed a radio resource management application for accommodating QoS aware RAN slicing. Besides, it can also monitor and control the RAN infrastructure.

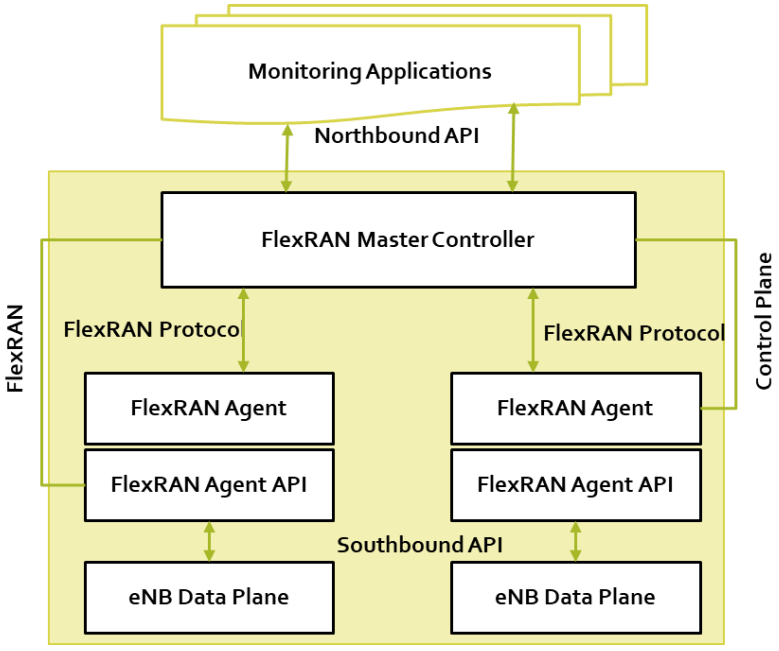

Figure 3.4: Abstract architectural view of FlexRAN controller for RAN domain slicing

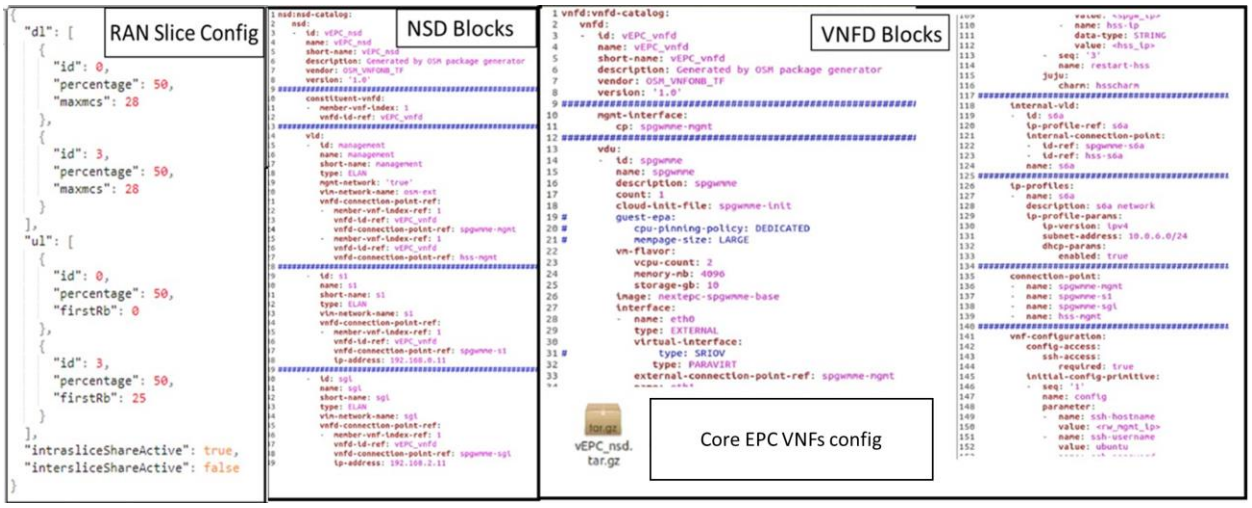

Figure 3.5: RAN and core network configuration templates for the deployment of resources through OSM and FlexRAN

Figure 3.5 illustrates the configurations sample generated through the IBN policy configurators for OSM and FlexRAN controller. These configurations are forwarded to the OSM platform to deploy core EPC VNFs. OSM receives configurations from the REST interface in the form of JSON-based NSD (Network Service Descriptors), and VNFD (Virtual Network Function Descriptor), and NST (Network Slice Template). NSD contains configurations for establishing the connection between VNFs to provide a service. On the other hand, VNFD contains VNFs configurations related to interfacing, networking, and resources. NST has connectivity information among VNFs and performs service function chaining (SFC).

3.2.3. Slice Run-time Monitoring

The run-time monitoring of slice resources is essential for efficient resource management. Figure 3.6 illustrates the monitoring module that works in two ways: RAN domain monitoring and Core VNFs resources monitoring. The eNodeB resources are monitored using the FlexRAN controller

and elasticMon tool [83]. The elasticMon is a monitoring tool specially designed for 5G mobile networks to monitor the enormous real-time data traffic to control and manage it.

FlexRAN controller and elasticMon tool can monitor and store the statistics of the eNodeB. Kibana and Grafana tools are used on the top of the FlexRAN controller and elasticMon to visualize the data traffic on runtime. On the other side, core network VNFs are monitored by the Openstack telemetry service or ceilometer.

On top of that, MON OSM monitoring service collects the VNFs data logs that are further forwarded to Prometheus [84], which is an open-source tool for monitoring the real-time data traffic and stored these matrices into a time-series database. So, Prometheus kept the VNFs data logs into the real-time database repository [85]. After that Grafana tool is integrated with the Prometheus database to visualize the VNFs resource stats such as CPU usage, RAM usage, etc. Using these open-source monitoring tools, we can monitor and collect the network slice resource usage on runtime. In case of failure, our system can reconfigure and update the slice resources dynamically.

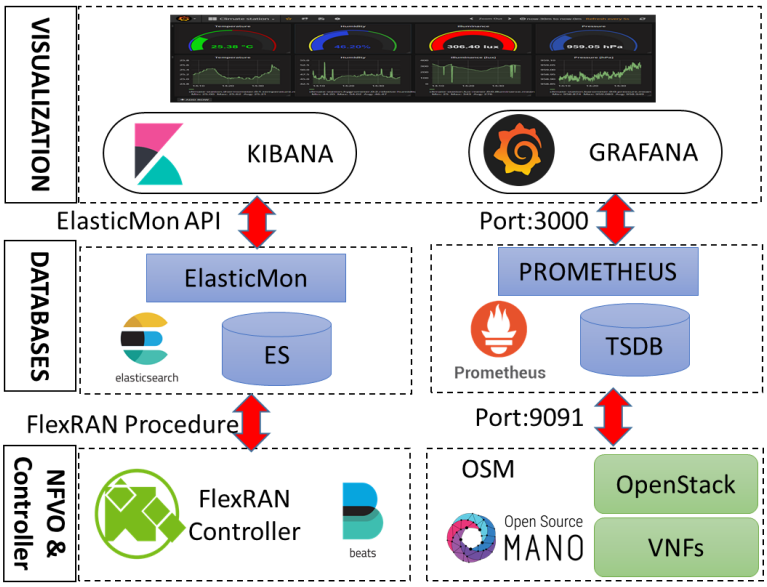

Figure 3.6: Deployed core and RAN resources monitoring mechanism

3.2.4. Slice Deactivation or Decommissioning

There are two folds for slice deletion using our IBN network slicing system. The first way is that the slice termination re-quest is generated through the IBN system web portal, which contains the slice SNSSAI (single network slice selection assistance information), SliceID. The slice deletion configurations are rendered the same as the slice creation process. The slice deletion template is forwarded to the NFVOs such as OSM and FlexRAN controller to delete the resources. The OSM and RAN controller delete the specified resources and release the resources for future usage. The second way is that slice starting and ending time should be inserted during the slice creation phase. When the slice ending time comes, the IBN system automatically generates the slice deletion request and automatically performs the same process. After that, deletion confirmation notification is sent back to the operator through the IBN platform.

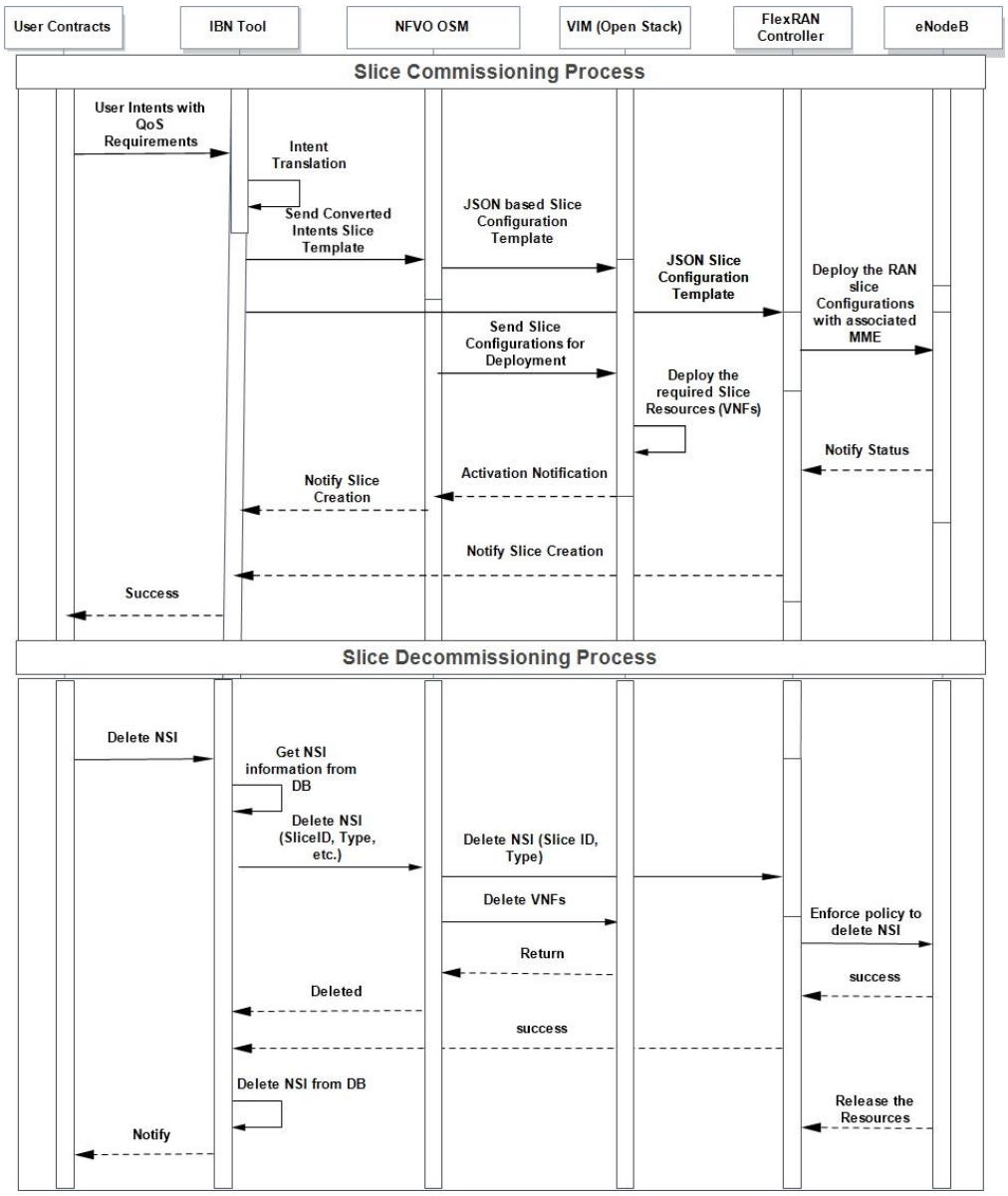

Figure 3.7: Procedure of network slice activation and deactivation through IBN

Figure 3.7 illustrates the step-by-step slice activation and deletion process through the IBN system. In the first step, the users define slice requirements in the form of intent by inputting QoS (uplink, downlink) requirements, sliceID, slice category information through the GUI of the IBN system. After that, the IBN policy configurators translates these higher-level requirements into policy configurations for each domain and forwards them to OSM and FlexRAN to deploy

resources. Furthermore, OSM prepares the core network EPC VNFs resources for the requested slice using OpenStack. On the other hand, the FlexRAN controller deploys the slice configurations at the RAN domain. These slice configurations include information about dedicated vMME, slice SNSSAI, and QoS requirements used to stitch RAN slice with core VNFs. Afterward, IBN notifies the user of successful slice activation. For the slice decommissioning process, the user needs to input the network sliceID information through the IBN portal. Then IBN manager checks the slice information from the catalog repository of the IBN system. IBN again translates the slice information into policy configurations and forwards them to OSM and FlexRAN controller to delete core and RAN domain resources associated with that slice. Subsequently, OSM with OpenStack deletes the core EPC VNFs, and FlexRAN deletes the slice at eNodeB. Finally, OSM and FlexRAN notify the IBN platform about slice deletion. IBN reports the user for successful slice deletion. In this way, the IBN system can automatically design, activate, and delete the slices where the user only provides higher-level slice requirements, eliminating manual effort.

3.2.5. Decision Engine

The decision engine is an integral part of the IBN system for automatic updates and resource assurance. The decision engine performs the decision based on a dynamic-thresholding oriented mathematical model to autoscale the core resources. It receives the HSTEL model prediction results and calculates whether the resource update is needed or not. It is entirely dependent on HSTEL model prediction results. In resource overloading cases, it performs scale-up decisions and notifies IBN for the deployment of more resources. Also, in the resource under-utilization case, it serves to scale in the resource with the help of IBN, OSM, and FlexRAN controller. On the other side, if there are no overloaded and under-loaded cases, the resource remains unchanged. It reports to the IBN platform about the decision, and IBN takes further action through the underlying

orchestrator and controller. The complete details of this module with the mathematical model is well presented in our paper [34]. Furthermore, the hybrid model traffic classification and detection results are continuous reports to the decision engine for further action for the second attack detection and mitigation scenario. If the malicious traffic is detected through the ML model, the decision engine notifies IBN for proper action to mitigate that attack. The IBN platform defines policies to stop the malicious flow from the system. So, IBN takes advantage of the ML models results to ensure service QoS and provides proper security. The decision engine issues smart alerts to notify the IBN platform. We have used Rest Interfaces to communicate all the components such as ML to the IBN platform. The use of ML models enhanced the capabilities of the IBN platform. So, IBN with ML intelligently performs management and orchestration of e2e network slice over the multidomain. Hence IBN can follow a closed loop to automate and manage the network resources and services.

3.3. Network Data Analytics Function (NWDAF) with IBN for Proactive Update and Assurance

This module is an innovative feature of the IBN platform that is the implementation of ML-based NWDAF. We have used a separate data analytics function (DAF) concept for each domain: RAN-DAF for radio access network domain, E-DAF for edge applications domain, and C-DAF for core network domain. Each DAF has pre-trained ML models on historical network data. These DAFs provide the results to the IBN platform to achieve specific use cases such as resource utilization prediction, attack detection, and prevention, mobility prediction, load balancing, etc.

Figure 3.8 illustrates the internal NWDAF ML pipeline mechanism with the IBN platform, where data is collected from the different source (SRC) nodes of the underlying data plane. We

have deployed a data exporter node inside each domain for data collection (DC) purposes using the KAFKA framework. The collected raw data has been stored in real-time NWDAF data storage separately according to each domain. After that, raw data is preprocessed through cleaning, transformation, and feature extraction ML operations. The extracted preprocessed data is stored in the data storage module. For data analytics and data storage, we are using the spark big data analytics framework. The stored preprocessed data is further used for training ML models. Once the model has been trained, it is evaluated based on performance measures such as accuracy and mean square error (MSE). If the model provides satisfying accuracy, then it will be deployed for execution. The results of various executed ML models have been forwarded to the IBN decision engine, which triggers update policy such as scale-up, scale down, or DDoS attack detection and prevention policy. After that, the issued policy will be deployed over the sink nodes through domain controllers and NFVO.

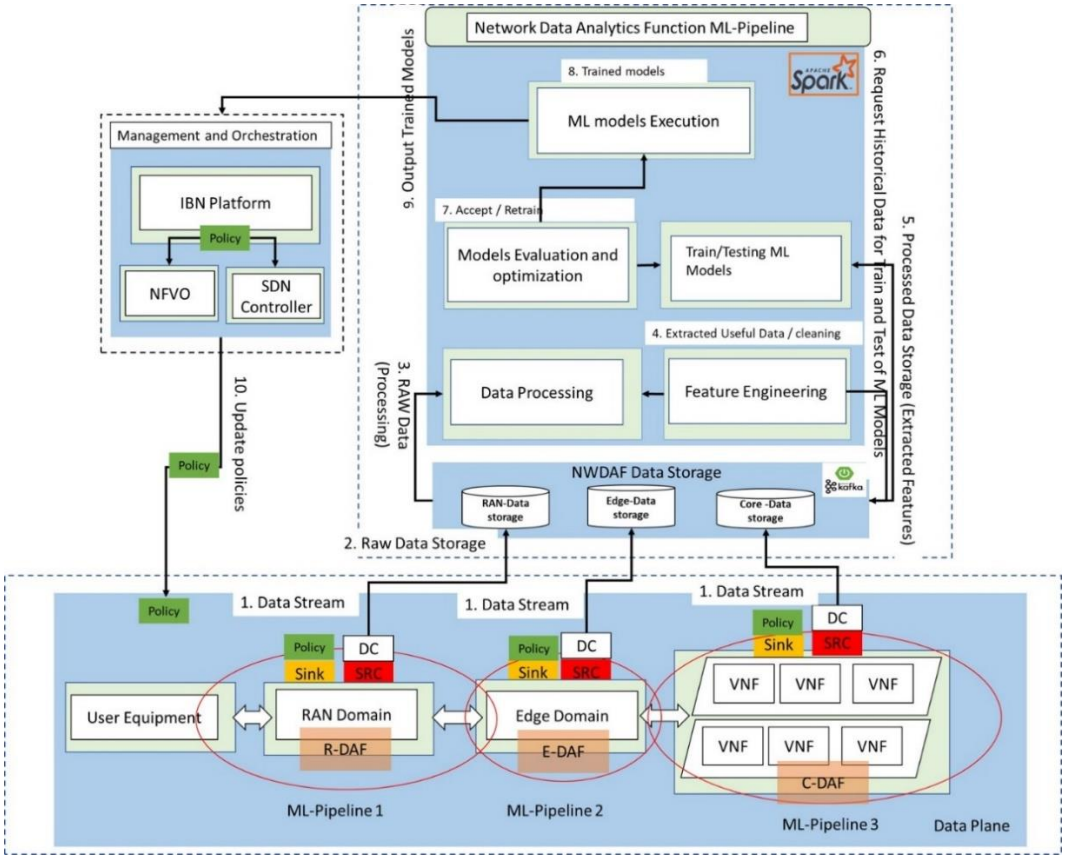

Figure 3.8: Network data analytics internal workflow

In our mechanism, we have used stacking EL-based model inside the C-DAF module to predict core VNF resource utilization or resource forecasts and DDoS attack detection and prevention. The other module decision engine of the IBN platform has a dynamic thresholding-based model that uses the NWDAF module's prediction results and performs the auto-scaling of the resources in case of performance degradation and notifies about the attack detection. Further, it reports the orchestrator and controller to block that malicious flow.

3.3.1. Dataset Preprocessing

For training the ML and DL models, a real-time dataset is required, so, we have used real-time cloud core resources dataset collected for the period of three months for training and testing of our

HSTEL model. Further details of the dataset are provided in this repository [86]. The collected dataset contains twelve performance matrices related to CPU utilization, RAM utilization, Disk write and received and transmitted throughput as highlighted in Table 1.

Table 1: Details of dataset features and their description

No#	Attribute Name	Description
0	Timestamp	Number of minutes
1	CPU cores	Number of used CPU cores
2	CPU capacity	Speed per CPU core
3	CPU usage	In MHz
4	CPU usage	Usage in percentage
5	Memory provisioned	Total memory of VM in KB
6	Memory usage	Used memory in KB
7	Memory usage	Usage in percentage
8	Disk write	Used in KB/s
9	Disk size	In GB/s
10	Network received throughput	In KB/s
11	Network transmitted throughput	In KB/s

Also, we have performed several preprocessing operations to transform the dataset, such as data cleaning, null value removal, and outlier removal:

1) There are separate CSV traces, and we have performed data preprocessing operations to merge them into a single file. 2) Null values are replaced with 0 because the model could not perform well if null values were in the dataset. The considered dataset is time-series, and it comes in the arena of regression problem. So, it is vital for a regression problem; the dataset should be cleaned and following the proper timestamp. 3) Using panda's library, we have removed several outliers from the dataset. 4) We have transformed the dataset and extracted important features for the

training of our hybrid HSTL model. 5) The original collected dataset conversion of minutely dataset into hourly dataset for mid-term prediction.

Figure 3.9 highlights the sample of CPU and memory utilization patterns in the dataset. It can be observed from the plot that, the actual CPU and memory are within the range of 10 to 70% usage but there are some spikes with 80% and 90% utilization. So, the ML model learns from these patterns and predicts future network resource utilization, which will be further used for autoscaling cloud resources. So, we are going to predict the future utilization of CPU and memory because these two are more critical factors for a VNF or VM. However, due to that, these two are our target variables.

Figure 3.9: A sample of target actual CPU and memory utilization in percentage

For testing the correlation of the features with other attributes, we have performed correlation analysis. Based on correlation analysis, the most important features are selected for training the ML model. Figure 3.10 highlights the heat-plot map, which shows the correlation of each attribute in the dataset. This correlation graph explains how each feature is correlated and dependent on

other attributes; for example, actual CPU utilization in percentage is more correlated to CPU usage attribute, and actual memory usage is more correlated to memory usage (KB).

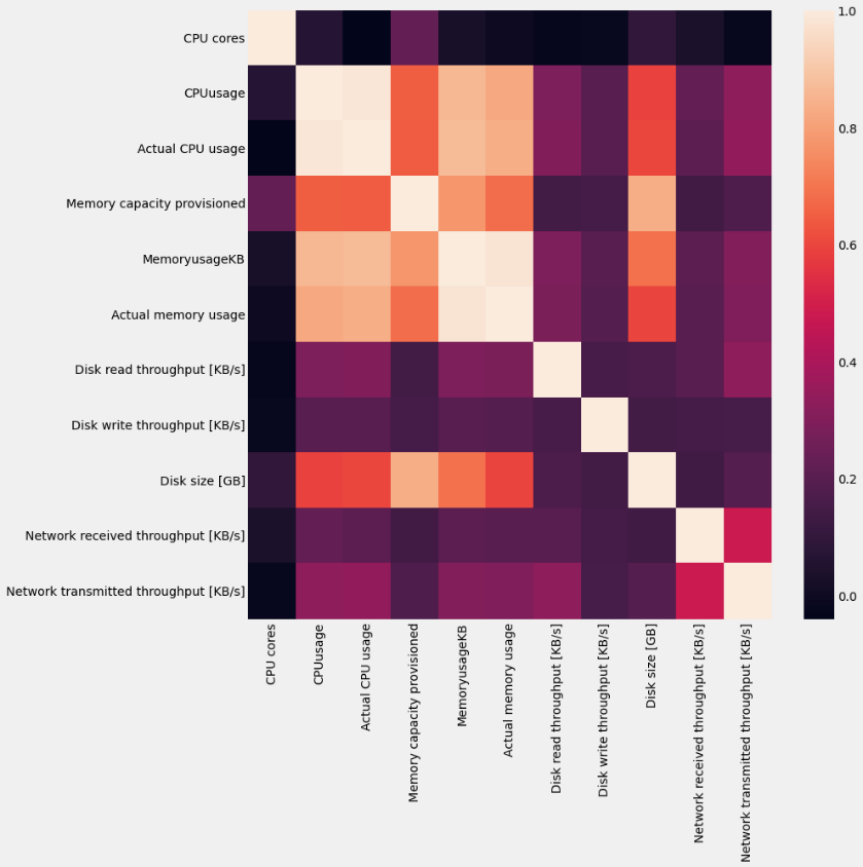

Figure 3.10: Heat-plot map for checking correlation among attributes

Furthermore, we have also performed feature importance analysis by using Random Forest (RF) [29] machine learning algorithm. We have separated timestamps by month, year, date, and time to predict better and according to the best and worst prediction results in peak hours and days. The importance of each feature considered in data preprocessing is illustrated in Figure 3.11.

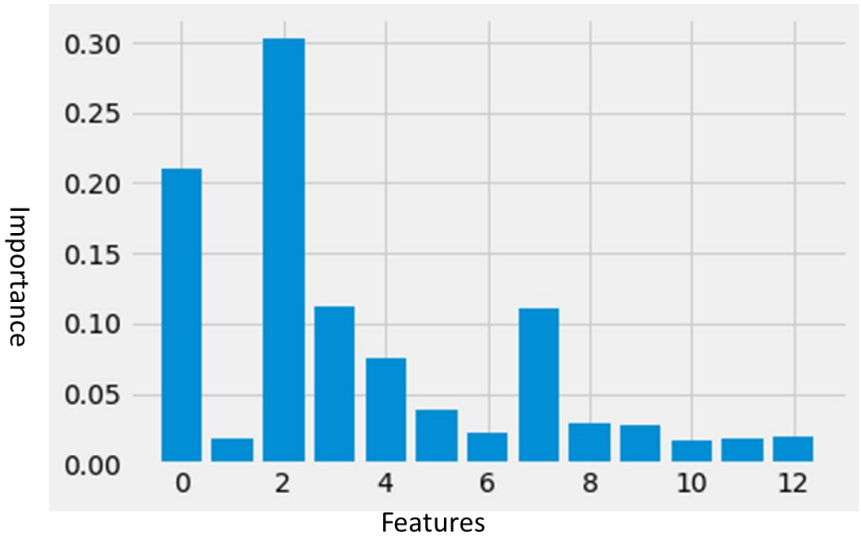

Figure 3.11: Feature important analysis through random forest model

Furthermore, we used 70% dataset for training the HSTEL and other ML models and 30%for testing purposes. We have used the Sklearn library to split the dataset into train and test set and used a manual approach. Evaluating the STEL model's performance shows satisfactory accuracy of almost 95% on CPU utilization prediction and 98% on memory utilization prediction. It predicts future CPU usage and RAM usage of the core VNFs. Currently, in C-DAF, we are performing short-term and mid-term forecasts such as minutes and hours. The further details of the proposed HSTEL model have been well-explained below.

3.3.2. Proposed Hybrid Stacking Ensemble Learning (HSTEL) Model for Network Resource Utilization Prediction

EL is a technique of ML in which various models are combined to achieve optimized and better accuracy. It categorizes into three types for implementing: averaging ensemble or voting ensemble, bagging (bootstrap ensemble), boosting, and stacking [32]. In this work, we have used stacking EL, also known as the stacked generalization approach, to better and optimize network resource

utilization prediction. Usually, the stacking EL technique's prediction results are better than individual models. In the stacking approach, several classifiers or regressor models are combined through meta-regressor or meta-classifier. The base learner models (level-0) results are used as input to train the meta-regressor model. The meta regressor model learns from the base learner how they make errors and resolve the error in the final prediction result [30] [87].

Furthermore, in our STEL model, we have used Gradient Boosting Machine (GBM) and Catboost regressor as base learner (Level-0) models and Extreme Gradient Boosting (XGBoost) as meta-learner (Level-1) model. The Architecture of proposed STEL model is illustrated in Figure 3.12. The details of implemented ML models are well explained below:

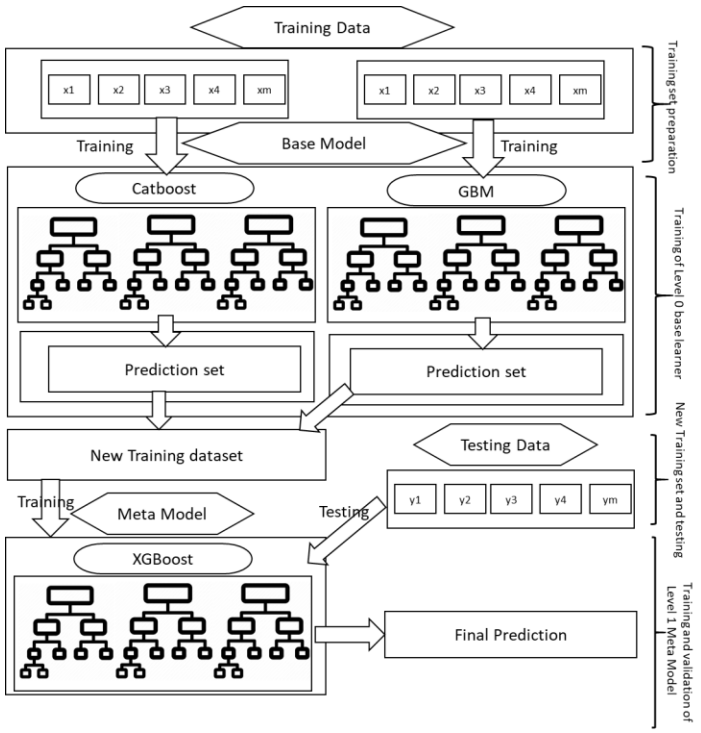

Figure 3.12: Design and Architecture of HSTEL model for network resource utilization prediction

3.3.2.1. Gradient Boosting Machine (GBM)

GBM is an extremely powerful ML algorithm introduced by Friedman in 2002 to perform classification and regression tasks accurately [88]. The basic behind GBM is to minimize the loss function by adding up a weak leaner in the model to accommodate the shortcomings of the existing weak learners. It reduces the variance and bias by including base learners (weighted sum) and refocusing the wrongly classified data. It uses multiple regression trees as base learners, and their results are combined to get the final output. The boosting approximation function is explained in equation 1 where $h(t;b_k)$ is a function for base learner model, B_k is expansion coefficients, t represents explanatory variables, and b_k illustrates the model parameters.

$$F(t_i) = \sum_{k=1}^{N} B_k h(t;b_k) \qquad 1)$$

The following hyperparameters were used to implement the GBM model: 1000 trees, 0.1 learning rate, evaluation criteria is Friedman MSE, 1.0 subsample, and 0.1 validation fraction. So, various recently developed boosting algorithms such as Catboost, LightGBM, and XGBoost used GBM as a base algorithm to boost scalability.

3.3.2.2. Gradient Boosting Model (XGBoost)

XGBoost is a widely adopted supervised learning algorithm proposed by Tianqi Chen and Carlos Guestrin in 2014 [89] . It works on boosting principle where an active learner can be created from the weak learners. It is a tree integration algorithm and efficient implantation of the GBM algorithm based on function approximation and regularization techniques. It is a scalable tree boosting mechanism that is more than ten times faster than other boosting algorithms. The most crucial factor in the XGBoost algorithm is parallel and distributed computation ability and innovative tree learning model for handling spare data. XGBoost uses the cumulative sum of the

predicted values of a sample in each tree as sample prediction in the model. Moreover, it reduced the overfitting problem due to the addition of regularization terms and loss functions optimization.

In our scenario, dataset D is given with n features $D= [(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ..., (X_n, Y_n)]$ to the model which has K number of trees for the target prediction $\hat{Y}_i$ as illustrated in Equation 2. Also, $\hat{F}_X = W_{Q(X)}$ $(W \in R^T)$ is Classification and regression Tree (CART) space, Q denotes structure of each tree, T represents number leaf nodes in the tree, W is leaf weights and F_K presents K_{th} tree.

$$\hat{Y}_i = \sum_{k=1}^{K} F_k(X_i),\ F_k \in F \qquad 2)$$

$$Obj\ (\theta) = \sum_i^I L\left(Y_i, \hat{Y}_i\right) + \sum_k^K \Omega(F_k) \qquad 3)$$

$$Obj\ (t) = \sum_i^n L\left(Y_i, \hat{Y}_i^{(t-1)} + F_t(X_i)\right) + \Omega\ (F_t) \qquad 4)$$

Equation 3 explains the objective function which has two parts loss function regularization terms, where Y_i is true label values, $\hat{Y}_i$ is the predicted values, $\sum_i^I L\left(Y_i, \hat{Y}_i\right)$ is the loss per sample and loss function L can be customized in several ways. The main goal is to minimize the objective function. On the other side, $\Omega\ (F_k) = \gamma T + \frac{1}{2}\ \lambda \sum_{j=1}^{T} W_j^2$ is regularization term where W is j_{th} tree leaf node weight, γ is regular leaf tree penalty term, and λ is regular leaf weight penalty term. These both penalty terms act as a smoothing factor and prevent the overfitting problem.

Equation 4 illustrates the objective function for t_{th} tree, whereas *t-1* is sum of the predicted values of the *t-1* tree, $F_t(X_i)$ is output of the t_{th} tree, and regularization values become the corresponding values of the latest tree. So, the final prediction is the sum of the output of numerous trees. The following setting parameters were used to tune XGBoost Model: 5000 total number of

trees, 50 early stopping rounds, maximum depth of the tree is 3, learning rate 0.1, linear regression objective function, and gbtree booster.

3.3.2.3. Catboost Model

Catboost is an innovative algorithm works based on symmetric decision trees proposed by Prokhorenkova et al. [90] in 2018. It processes categorical features efficiently with less information loss. Firstly, it uses modified GBM technique ordered boosting to control target leakage issues. Secondly, it is the best algorithm for handling small datasets. Thirdly, Catboost performs statistics operations on the categorical feature, computes the frequency of a specific feature category, and then includes several hyperparameters to generate numeral features. So, it preprocesses the categorical features by replacing them with numerical values. Moreover, it solves the issues of prediction shift, gradient bias, overfitting and improves model accuracy. Equation 5 explains the transformation of categorical features into numerical, where C_c is the class counter, average target T_{avg}, P_r is the initial numerator value, and C_t is the total counter.

$$T_{avg} = \frac{C_c + P_r}{C_t} \qquad 5)$$

$$F(t_i) = \sum_{k=1}^{N} C_k 1(t \in r_j) \qquad 6)$$

In equation 6, $F(t_i)$ is the tree function of t_i explanatory variable and r_j is the disjoint region in correspondence to the leaves of the decision tree. We have used the following hyperparameters to tune the Catboost model: 0.043 learning rate, the symmetric tree growing policy, plain boosting type, maximum leaves 64, 0.800 subsample, and 1000 iterations.

Algorithm 1: Hybrid stacking ensemble learning (HSTEL) model for network resource utilization prediction

Algorithm 1 Stacking Ensemble Learning (STEL) Model for Network Resource Utilization Prediction.

$D \leftarrow \{(x_1, y_1), (x_2, y_2)...(x_n, y_n)\}$ ▹ network resource utilization dataset
x_n = Feature Vector, y_n = Final Prediction => n = Total Observations
$Base\_Learners(level-0) \leftarrow \{B_1, B_2...B_z\}$
$Meta\_Learner(level-1) \leftarrow M_l$ ▹ Define meta-model
$Ensemble\_Total\_Size(level-1) \leftarrow N$
for n=1 to N **do**
    $B_z \leftarrow$ { creates *Base_Learners* from D } ▹ Level-0 models(base learners creation)
end for
→ creation of new dataset (D_{new}) for *Meta_Learner* ▹ New dataset preparation for meta-model
$D_{new} \leftarrow 0$
for m=1 to M **do**
    for n=1 to N **do**
        → make prediction with *Meta_Learner*
        $B_{mn} = B_z(x_m)$
    end for
    $D_{new} = D_{new} \cup \{((B_{m1}, B_{m2}...B_{mz}), y_m)\}$ ▹ combine different base regressor models
end for
→ training of *Meta_Learner* with new dataset D_{new}
$M_l^t = M_l, (D_{new})$ ▹ training of level-1 model
Prediction results → final prediction of *Meta_Learner* (M_l^t)
Model validation → test performance based on error measures: $RMSE, MSE, MAE, MAPE, R_2$

The network resource utilization prediction is a regression task in this work, where a model f takes the input feature vector onto the label values (target features). The training dataset is used to train model *f*, which falls within the supervised learning scope. As explained in equation 7, the regression problem is formulated as a minimization problem. The first part of the objective function is empirical risk outlined by a loss function that calculates the quality of the model *f*. The second part is the regularization term used to estimate the complexity of the model *f*, and λ is the regularization parameters.

$$\min(f_m) = \sum_{k=1}^{m} L(f_m((X_i), Y_i) + \lambda(f_m)) \quad 7)$$

$$P(STEL) = (P_{GBM} * \propto) + (P_{CB} * \beta) + \gamma \quad 8)$$

In our case, network resource utilization dataset D with n features $D= [(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots, (X_n, Y_n)]$ is divided into training and testing dataset for the training and validation of STEL model. Algorithm 1 explains the working mechanism of the hybrid STEL model. The training dataset has been used to train the base learner regressor models, and the validation set is used to test the prediction results. After that, prediction results of the level-0 models were used as input for training the XGBoost model. So, the XGBoost model was constructed based on two base-leaner models. XGBoost model trained on that dataset and tested on the test dataset. The prediction of the meta-learner regressor is the final prediction result. Equation 8 explains the final prediction of the STEL model, which is a linear combination of GBM and Catboost models. $P(STEL)$ is the final prediction of the STEL model, P_{GBM} and P_{CB} are the output of GBM and Catboost respectively, $\propto$ and β are weighting coefficients achieved through XGBoost model fitting, and γ is a correction constant. Furthermore, the STEL model predictions have been tested by calculating various error metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute percentage error (MAPE), and Regression-score R2 \cite{abdullah2020predicting}.

However, the prediction results of the HSTEL model will be used by the IBN decision engine. It has a dynamic thresholding-based model which decides the autoscaling of resources to assure QoS requirements. If there is a sudden increase in traffic, it will trigger a policy to deploy more VNFs to avoid performance degradation. Hence, NWDAF predictions results are used to perform the network's autoscaling of resource and anomaly detection and mitigation. So, the IBN platform with NWDAF performs closed-loop orchestration, control, and management of the e2e slice resources.

3.3.3. Hybrid Model for Anomaly Detection

Similar to network resource prediction, anomaly detection and mitigation is also crucial aspect of future networks. It is a critical use case of NWDAF defined by 3GPP. Network security has been a big challenge due to the continuous increase in smart devices in the last years. Therefore, anomaly detection and mitigation have gained a lot of attention from researchers. So, an automated ML mechanism is required to detect the anomalies from the system and take proper action to mitigate them from the network.

In the last years, ensemble learning techniques have been used in many areas of classification and regression. More specifically, these techniques have been widely used in the cybersecurity field anomaly classification. The main concept behind EL is to improve the performance of ML by combining several classifiers. The hybrid EL model shows better performance compared to the individual model in most cases.

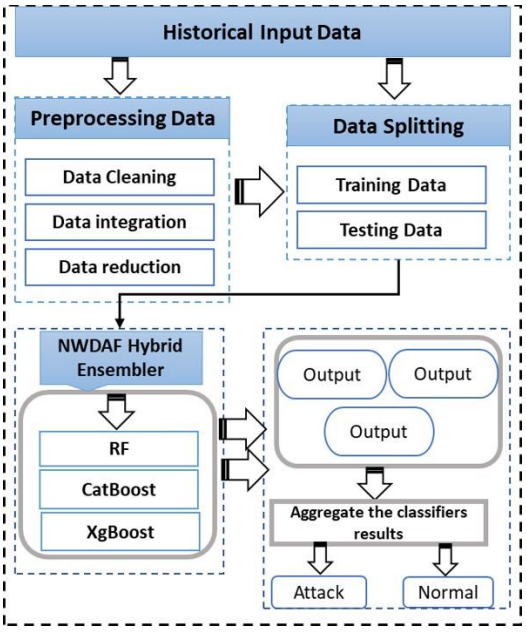

Figure 3.13: Hybrid ensemble learning model for anomaly detection from the system

Due to the many advantages of EL approaches, we have developed a hybrid model for anomaly classification from the network. The Random Forest (RF), Catboost, and XGBoost ML classifiers are combined using the EL voting method to perform better and optimize classification. The RF, Catboost, and XGBoost models are used as base models and combined through a voting ensemble model to achieve final classification output [68] . Figure 3.13 illustrates the workflow of proposed hybrid model for attack classification. The working functionalities of Catboost and XGBoost are well explained in the last section. On the other side, the working of the RF algorithm is discussed below.

3.3.3.1. Random Forest (RF)

RF is a viral ML algorithm that is mostly used for regression and classification problems. It comes under the umbrella of ensemble-learning approaches. The RF classifier completely works based on a decision tree (DT). Additionally, it is easy, fast, and simple to implement and highly successful while performing classification and regression [91], [92]. RF is a collection of DTs corresponding with a set of bootstrapping samples created from the actual dataset. The samples generated through the bootstrapping method are the same size as the actual dataset. The splitting of nodes are based on the Gini index (entropy) of a chosen sample of features. The main principle of the RF algorithm is the construction of multiple simple DTs in the training phase and performing classification based on a majority voting mechanism. Besides, due to the voting mechanism, it overcomes the DT overfitting issue on training data. During training, RF used EL bagging mechanism for each tree. In the bagging method, the random samples with replacement are selected from the training dataset and associated with DTs. Also, the DTs in the ensemble automatically learned through out-of-bag (OOB) errors. The samples that remain unselected through bootstrapping method are used to evaluate the DT [93].

Moreover, Bootstrapping helps to develop RF with the required number of DTs to improve classification accuracy and reduce overfitting issues. On the other hand, bagging is used to select the best DTs through the voting method. After developing the forest, a new data sample is given to that forest for classification; each DT in the forest casts a vote for a class that indicates tree decision. So, the RF chooses the class which gains most of the votes from the forest.

To test the performance, we have trained our hybrid models on two benchmark datasets: a well-known KDD-CUP DDoS attack dataset [94] and a synthetic attack dataset. The trained model detects anomalies from the network and reports to the IBN platform to execute mitigation policy and stop that flow.

3.3.3.2. Dataset Information

Table 2 highlights the details of the used dataset with its attributes. The considered dataset was labeled as normal and attacked traffic. The hybrid model learned the feature patterns and performed classification This dataset is used to train our hybrid model to perform attack classification that achieved 97% accuracy in the testing phase. The experimental results show the superiority of the proposed model compared to existing models.

Besides, another benchmark attack dataset was used to validate the system performance. We have performed data preprocessing by removing null values, outliers, data cleaning and data transformation.

Table 2: Details of first dataset

Column-Name	Data Type
frame.encap_type (int64)	tcp.srcport (int64)
frame.len (int64)	tcp.dstport (int64)
frame.protocols (int64)	tcp.len (int64)
ip.hdr_len (int64)	tcp.ack (int64)
ip.len (int64)	tcp.flags.res (int64)
ip.flags.rb (int64)	tcp.flags.ns (int64)
ip.flags.df (int64)	tcp.flags.cwr (int64)
p.flags.mf (int64)	tcp.flags.ecn (int64)
ip.frag_offset (int64)	tcp.flags.urg (int64)
ip.ttl (int64)	tcp.flags.ack (int64)
ip.proto (object)	tcp.flags.push (int64)
ip.src (object)	tcp.flags.reset (int64)
ip.dst (int64)	tcp.flags.syn (int64)
tcp.window_size (float64)	tcp.flags.fin (int64)
tcp.time_delta (object)	

Figure 3.14 shows the correlation matrix or heat-plot, which indicates the importance of each feature and dependencies. After performing correlation analysis, we have also checked feature importance through the RF model. In addition, the features are selected based on correlation analysis through heat-plot map and RF model. The selected features are used to train the hybrid model. On the other side, categorical distribution is performed on this dataset, as depicted in Figure 3.15, which shows the attack classes and their distribution.

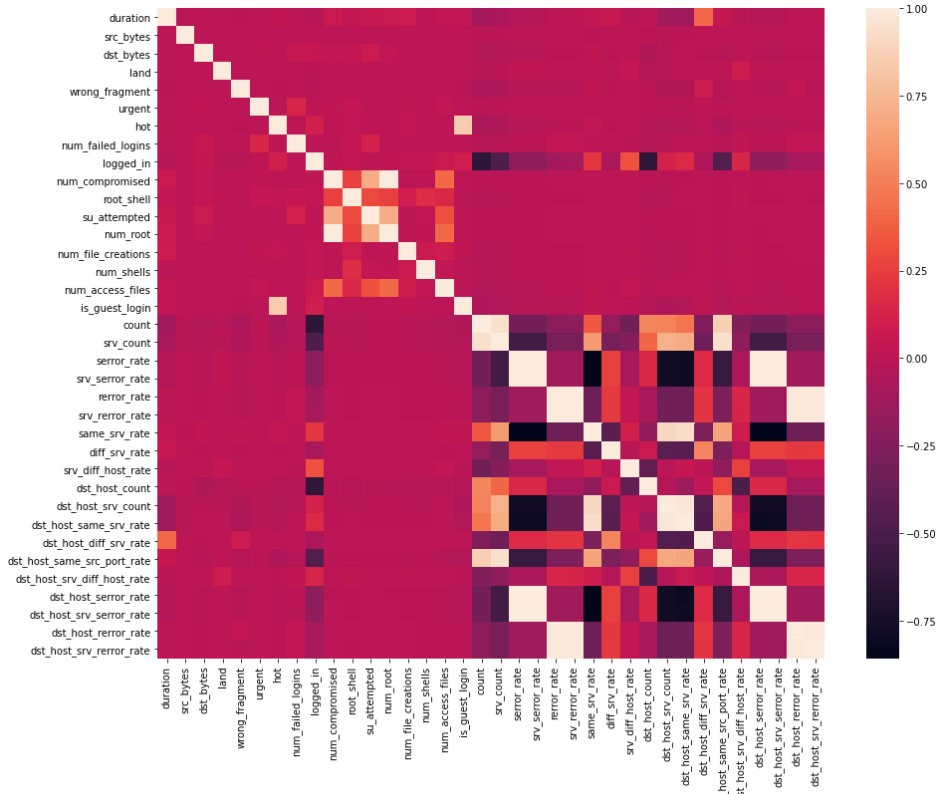

Figure 3.14: Correlation analysis of the dataset which shows the feature dependances and importance

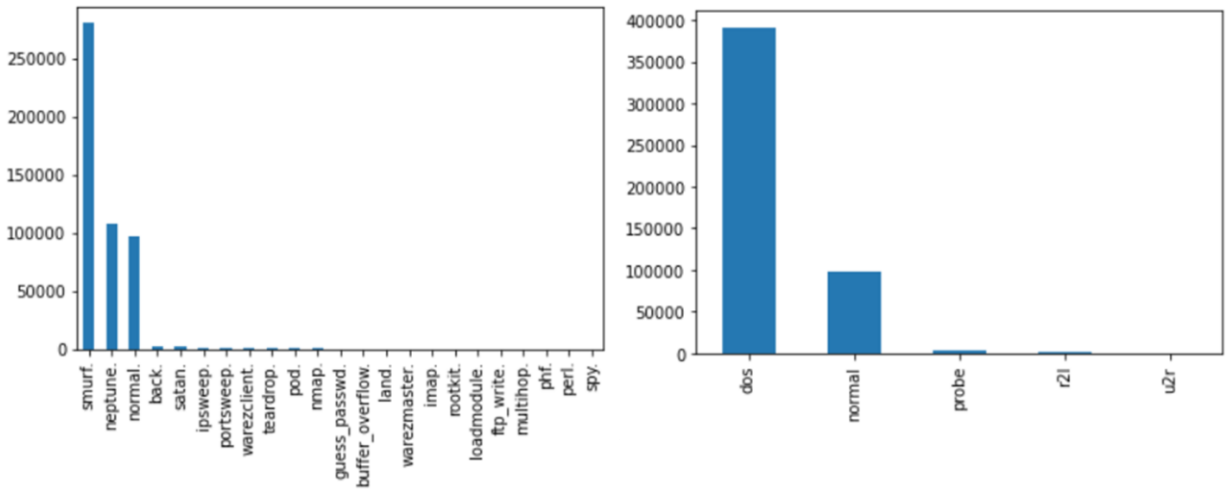

Figure 3.15: Attack types and their distribution in the dataset a) shows the explicitly attack type and their weight b) highlights five broad classes of the dataset

After performing the categorical distribution, the data splitting was done for the model training. It was divided by a 70:30 ratio, 70% of the data was used to train the model, and 30% to test the mode's performance. The hybrid model shows satisfactory results, and it achieves 98.3% accuracy during the training phase and 97% in the validation phase. So, in this way, the Hybrid model performs anomaly detection from the system, and the IBN decision engine checks the detection results of the model. If anomalous traffic enters the system, the IBN platform enforces a mitigation policy to NFVO OSM and FlexRAN controller to stop that malicious flow.

Chapter 4

Experimental Results and Discussion

4.1. Results of E2E Network Slicing through IBN System

We have designed and implemented an IBN platform for e2e network slicing to automatically create, update, monitor, and delete network slice instances. It has a dashboard portal for the users to define the QoS requirements for a network slice. The requested slice requirements are translated into a domain-specific policy template through the policy translator module. After that, the network orchestrator deploys those policies over the infrastructure and activates the resources for operation. The IBN platform communicates with the underlying domain orchestrator through the REST interface. The testbed of our e2e network slicing mechanism comprises of IBN system,

OSM orchestrator, FlexRAN controller, resource monitoring mechanism, EL-based NWDAF module, and OAI NFs. Figure 4.1 illustrates the components of implemented testbed for network slicing. The IBN system can manage and handle the complex configuration generation tasks and prepares the slice template according to the underlying platforms. IBN is the core entity of our proposed mechanism for performing e2e network slicing. It acts as the main orchestrator and handles the slice designing, admission, and control mechanism.

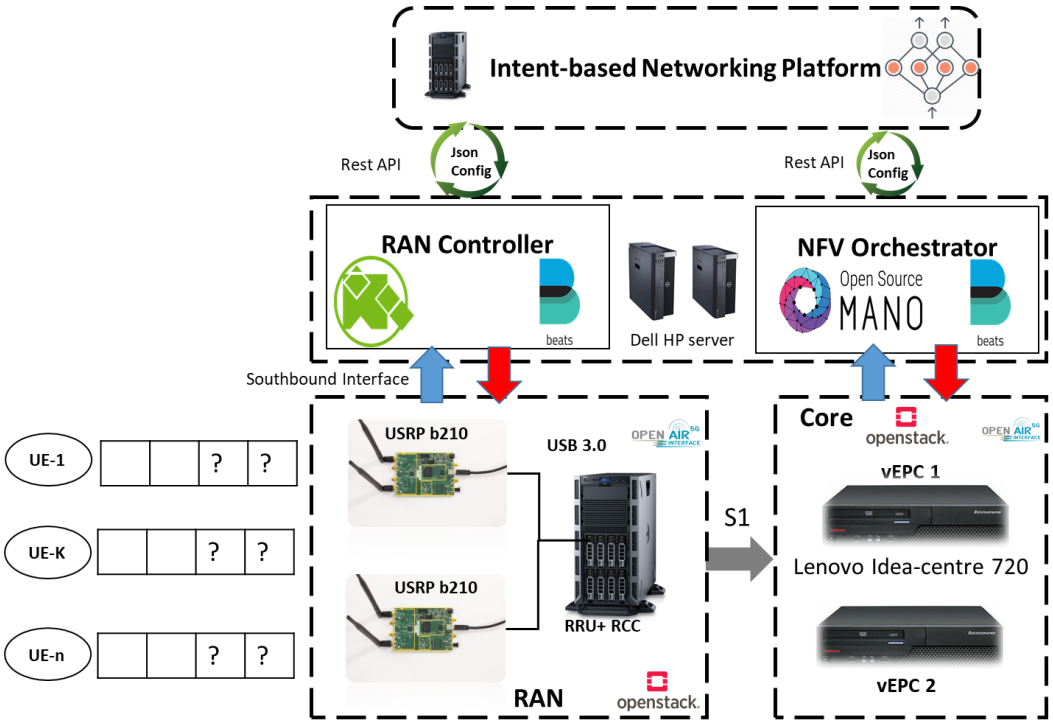

Figure 4.1: Testbed for e2e network slicing

4.1.1. Experimental Testbed Details

The OSM platform is used for deploying the core network VNFs, and the FlexRAN controller handles the RAN slicing mechanism in our work. In addition, the IBN system prepares the policy

configurations template for OSM and FlexRAN controller. Currently, in this testbed, we have tested the core and RAN slicing mechanism. Due to the lack of 5G resources, we have implemented our network slicing system using LTE OpenAirInterface (OAI) components. The OAI [95] is an open-source community that provides the LTE mobile network implementation. OAI provides EPC VNFs (vHSS, vMME, VSPGW), eNodeB, and UE. So, we have used OAI components for implementing e2e core and RAN domain network slicing. The OSM and OpenStack have been used for deploying dedicated core VNFs for a specific slice.

On the other side, multiple eNodeBs are deployed with software-defined radio (SDR) USRP (Universal Software Radio Peripheral) B210 for realizing LTE RAN capabilities. The FlexRAN controller is deployed in a separate machine that handles multiple eNodeBs. The RAN slice and core EPC VNFs are stitched by providing a dedicated vMME configuration in the template for establishing an e2e connection. Each slice is admitted using a unique public land mobile network (PLMN) id, and users can access a specific slice. The FlexRAN controller slice the RAN resource following the policy configurations received through the IBN platform. In addition, the FlexRAN controller can handle and manage the RAN domain by creating and deleting the slice instances over the infrastructure.

So, the IBN system can create a policy template in JSON string for FlexRAN and implement the received configurations over the RAN domain. On the other side, dedicated vMME and other NFs are assigned to the created RAN slice. Hence, the IBN system automates the policy template generation mechanism for multi-domain and manages the core and RAN resources through OSM and FlexRAN. The implemented monitoring tools can continuously monitor the

Core and RAN resources and collects the logs on runtime. The testbed components and details are presented in Table 3.

Table 3: Details of system components and their configurations

System Components	Specifications
IBN tool	OS: Window 10 RAM: 16 GB CPU: Core I5 3.0 GHZ H/D: 1 TB Programming languages: Python, PHP, JAVA Front-End languages: Bootstrap, jQuery, HTML 5, CSS 3, JavaScript Database: MYSQL
Open-Source MANO (OSM)	OS: UBUNTU 18 LTS RAM: 252 GB CPU: 32 Cores 2.10 GHZ H/D: 2 TB OSM Version: 7 Openstack Version: stein
SDN Controller FlexRAN	OS: UBUNTU 16.04 RAM: 16 GB CPU: CORE-i5 3.0 GHZ SSD: 500 GB
SDR USRP B210	Frequency Range: 70 MHz-6 GHz Channels: 2TX * 2RX

IBN platform web portal is presented in Figure 4.2, where users input slice requirements in user intents form. Received intents are translated into policies/slice templates with the help of domain policy configurators for underlying OSM and FlexRAN controller. Figure 4.3 shows the core EPC VNFs deployment status through OSM. It highlights the topology information, the relationship between VNFs, and interfacing. Moreover, OSM and FlexRAN controller enforces those policies over the Core and RAN infrastructure to activate network slice.

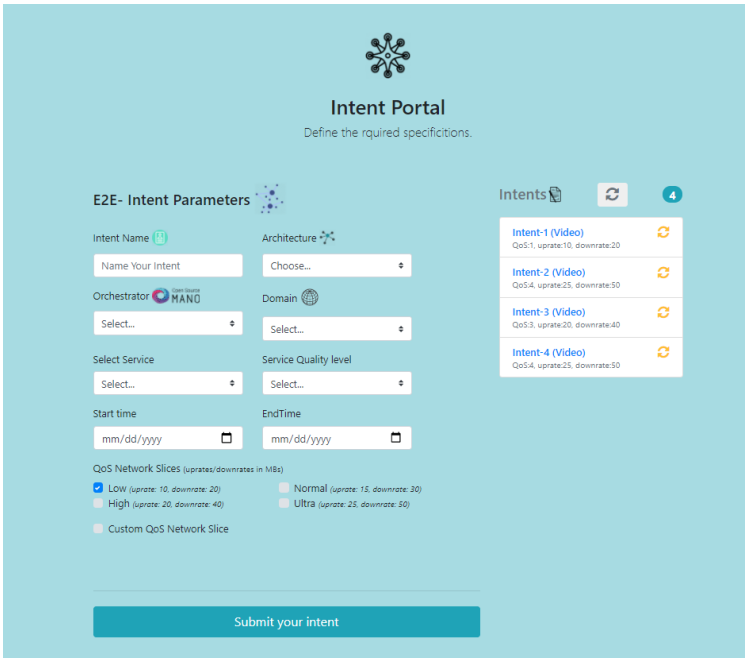

Figure 4.2: Web portal of the IBN platform for inputting service requirements in intent form

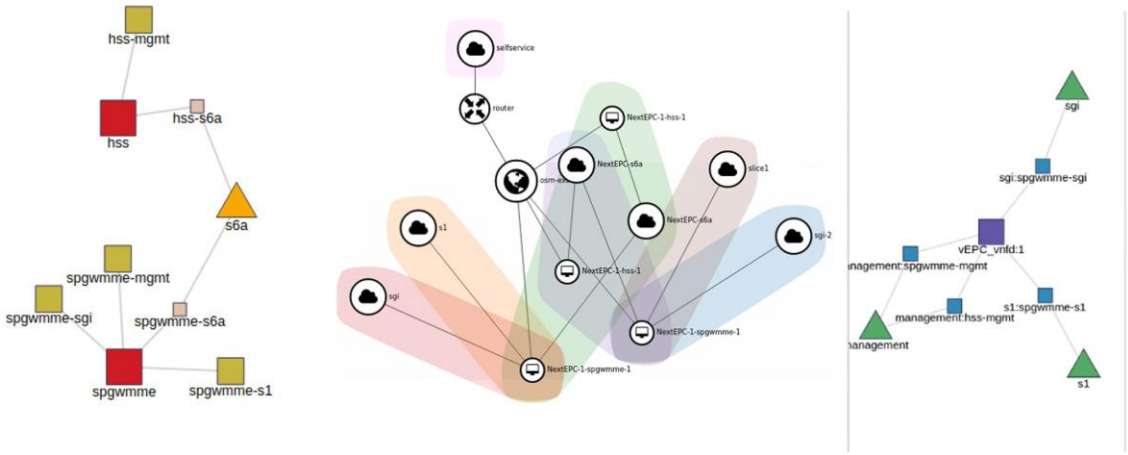

Figure 4.3: Status of EPC VNFs deployment using OSM

4.1.2. Results and Discussion of Network Slicing

The policy/slice template contains all the information related to the QoS requirements, such as bandwidth, memory, CPU, instance (VNF) images information, etc. For example, the RAN policy

template generated by the RAN policy configurator contains public land mobile network (PLMN), uplink, downlink, slice SNSSAI-ID, and type of service. These QoS requirements for a slice are further converted into the required number of radio resource blocks through the RAN slicing application inside the FlexRAN controller. Besides, FlexRAN controller deploys the resource through static approach such as 50% 40% of RAN resources. So, in our mechanism, we have used both the static and dynamic polices for slicing the RAN resources.

On the other side, the dedicated EPC VNFs are activated similar to the specified resources in the user contract and assigned to the RAN slice. The RAN slice template contains dedicated MME and other EPC VNF configurations for establishing an e2e connection. So, in this way our IBN system instantiates and activates core and RAN network slices over the infrastructure.

For testing the stability of our system, we have performed iPerf tests. We have created multiple slices with different QoS requirements of three types: eMBB, URLLC, and IoT slice. In the first case, we have deployed two static slices, eMBB, and IoT, with 50% of RAN resources, by inserting QoS requirements through the IBN portal, and the system automatically activates the e2e network slice with the cooperation of other components. Figure 4.4 shows the downlink speed recorded through the deployed e2e eMBB and IoT slice. The eMBB and IoT slice show almost similar throughput maximum of 23MB/s because of the equal resources requested through the SLA. On the other side, Figure 4.5 illustrates the downlink throughput of three slices recorded in multiple time slots with different QoS requirements. The eMBB, URLLC, and IoT slices are deployed with 30%, 20%, and 50% of the resource capacity, respectively. However, the IoT slice has achieved a maximum of 25 MB/s downlink throughput and an eMBB slice with a maximum of 17 MB/s throughput. Moreover, the maximum of 9 MB/s throughput speed was recorded with the URLLC slice.

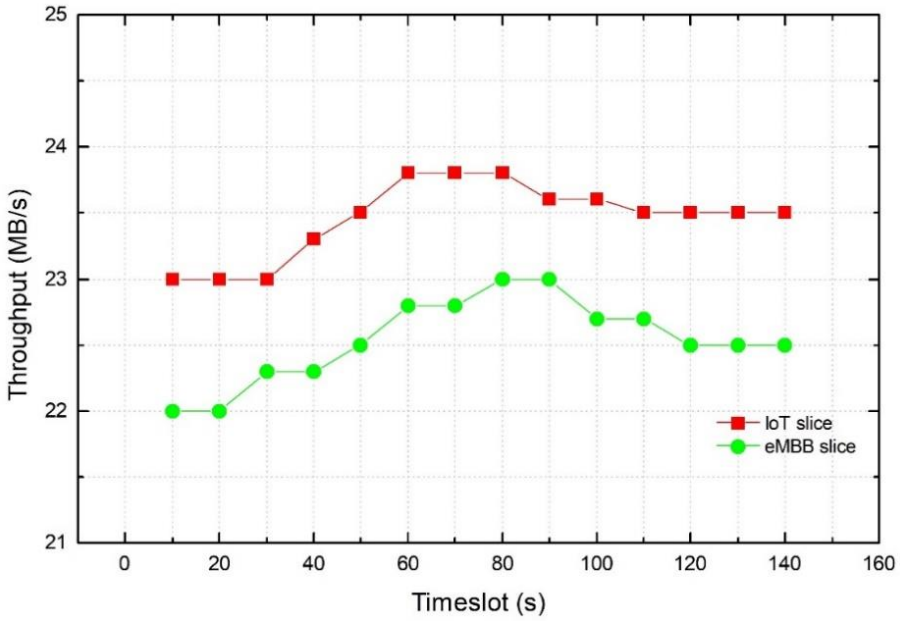

Figure 4.4: Average throughput testt of two slices instantiated through IBN system

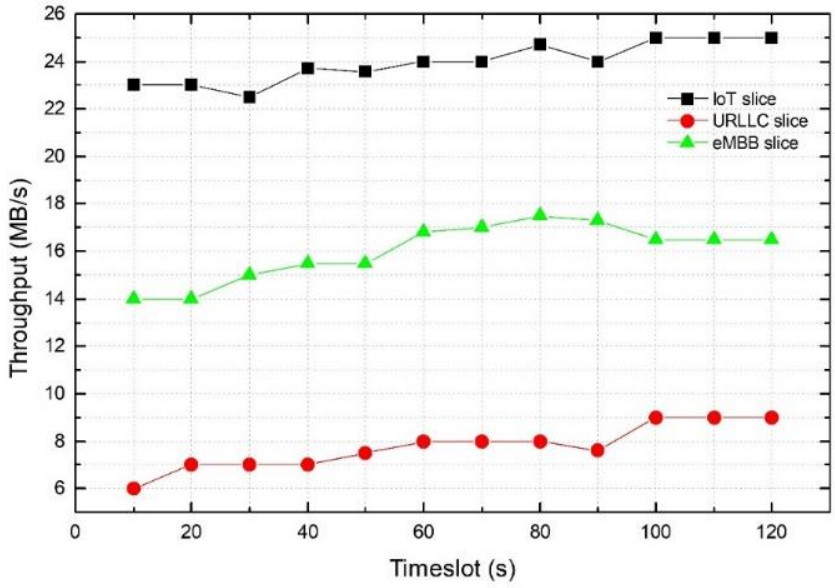

Figure 4.5: Average throughput test of three slices instantiated through IBN

Figure 4.6 illustrates the achieved downlink throughput results of four slices activated with different QoS requirements through the IBN system: eMBB slice#1 with 30% of RAN resources,

IoT slice with 30% of the resources, eMBB slice#2, and URLLC slice with 20% RAN resources for each. For the eMBB slice#1 slice, we have recorded a maximum of 17 MB/s throughput by performing several tests.

In addition, the IoT slice achieved a maximum of 15 MB/s throughput. On the other side, eMBB slice#2 got a maximum of 10.5 MB/s, and the URLLC slice achieved 9.7 MB/s by performing multiple tests. The first two slices show almost equal throughput because both are deployed with similar QoS requirements. Identical to the first two-slice, the last two were also activated with 20% resources for each.

Moreover, we have used four simulated OAI UEs to validate the deployed slice performance. The UEs are connected to the eMBB slice, which is activated with 20% of RAN resources. The achieved downlink throughput results for each UE are depicted in Figure 4.7, where UE1, UE2, UE3, and UE4 accomplish the maximum of 2900 KB/s, 2645 KB/s, 2500 KB/s, and 2305 KB/s throughput speed by performing several tests. Besides, we allocated 20% of RAN resources in user intent through the IBN intent portal, and FlexRAN converted those requirements and enforced them over the RAN resources.

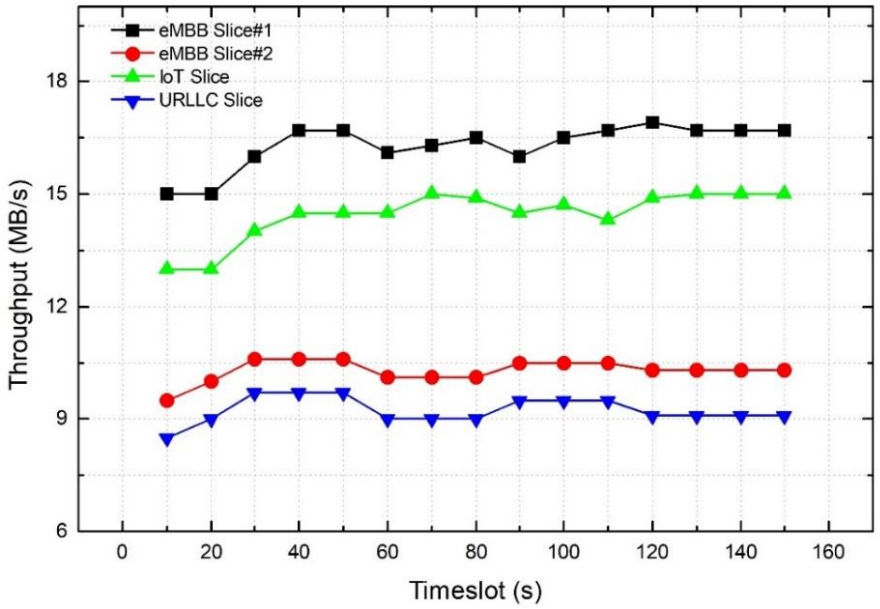

Figure 4.6: Average throughput test of four slices instantiated through IBN mechanism

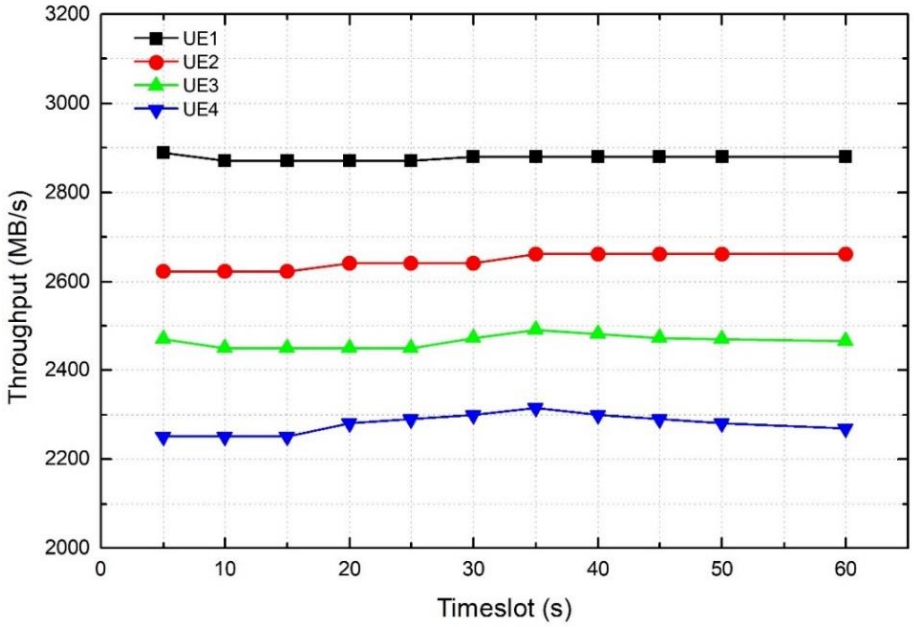

Figure 4.7: Average throughput test of four connected UEs with eMBB slice

The presented results show that the IBN platform can automatically create, activate, deactivate the network slices. It shows stable and satisfactory performance while performing e2e slice

lifecycle management. So, in our mechanism, we use SDR USRP B210 devices for LTE RAN, which is not a standardized solution. The primary purpose of testing the throughput is not to test the performance of our system compared to the standards solution but to show the stability of the created slices through the IBN platform. Despite many advantages of the implemented mechanism, there are few limitations. The dynamic allocation of RAN resources is still a challenging task. Further, the mobility management of the users is not handled by our system. The open-source implementation of 5G components is also a challenge for academic researchers.

The multiple VNFs are deployed through our system for creating core and RAN network slicing. Hence, our implemented system shows satisfactory performance while creating multiple e2e network slicing. it is a closed-loop mechanism that can orchestrate and manage the complete lifecycle of e2e network slices. Moreover, the ML-based NWDAF module is an interesting feature of our system, which predicts future resource utilization through the HSTEL model. In addition, it also has a hybrid model for Anamoly detection and mitigation from the network. Those predictions are used by the IBN system for efficient and proactive management of the core and RAN resources. So, our system can proactively prepare the resources whenever the traffic demands increase. It performs autoscaling by checking HSTEL prediction and efficiently placing the VNFs on the edge or core cloud location. The results of NWDAF for future resource utilization predictions are presented in the next section.

4.2. Results of HSTEL Model for Network Resource Utilization Prediction

4.2.1. Performance Metrics for Model Evaluation

We have used five widely applied performance measures to evaluate the prediction results of our proposed hybrid HSTEL model. These performance measures are RMSE, MSE, MAE, MAPE,

and R2. According to the literature, high regression-score R2 and low RMSE, MSE, MAE, MAPE is considered a criterion to validate the outcomes of the regression model. Presented below are the mathematical equations for each of the performance measures. Where $\widehat{P_k}$, $\overline{P_k}$, and A_k, are the predicted, average, and actual values of the sample k_{th}. N represents the total size of network resource utilization samples.

As illustrated in equation 1, MSE measures the average squared error of model prediction results. It is an average of the squared difference between the predicted and target values. RMSE shows the standard deviation of the sample and is calculated by just taking the squared root of MSE. On the other side, MAE is calculated by taking the average difference between actual and predicted values. MAPE calculates the accuracy of actual values and predicted values in percentage. Moreover, R2 is a statistical evaluation metric for measuring the prediction model's accuracy, also known as the coefficient of determination. It shows how a prediction model fits the actual network resource utilization values and how the predicted values are close to actual values.

$$MSE = \frac{1}{N}\sum_k^N \left(A_k - \widehat{P_k}\right)^2 \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_k^N \left(A_k - \widehat{P_k}\right)^2} \quad (2)$$

$$MAE = \frac{1}{N}\sum_k^N \left|A_k - \widehat{P_k}\right| \quad (3)$$

$$MAPE = \frac{1}{N}\sum_k^N \left|\frac{A_k - \widehat{P_k}}{P_k}\right| \quad (4)$$

$$R_2 = 1 - \frac{\sum_k^N (A_k - \widehat{P_k})^2}{\sum_k^N (A_k - \overline{P_k})^2} \quad (5)$$

4.2.2. HSTEL Model Prediction Results

In this section, the performance of the HSTEL model has been evaluated and compared with individual base models such as GBM, Catboost, and XGBoost. We have performed short-term and mid-term network resource utilization prediction in which the short-term is based on 5 minutes time intervals, and the mid-term is hourly resource usage prediction. The prediction results of the proposed model have been validated on the test dataset through above mentioned performance measures. Table 4 highlights the achieved performance metrics on the test dataset for each model while predicting CPU and memory attributes for short-term. These results indicate that the hybrid HSTEL model outperformed all the individual base learner models. Overall, all the models in our study are properly tuned by adjusting parameters and performing better than the state-of-art models. The proposed model shows less RMSE, MSE, MAE, and MAPE as compared to the base models on both memory and CPU utilization prediction. On the other side, it shows R2 accuracy of 0.95 for CPU utilization prediction and 0.98 for memory utilization prediction.

Table 4: Evaluation metrics of short-term multi-attribute network resource utilization prediction by proposed HSTEL model

	CPU utilization Prediction				Memory utilization prediction			
	GBM	XGBoost	Catboost	HSTEL model	GBM	XGBoost	Catboost	HSTEL model
RMSE	0.736	0.63	0.692	0.586	0.397	0.31	0.705	0.153
MSE	0.541	0.397	0.478	0.343	0.157	0.096	0.496	0.023
MAE	0.177	0.105	0.113	0.087	0.223	0.137	0.26	0.041
MAPE	1.869	0.81	1.94	0.53	0.989	0.505	0.743	0.159
R2	0.92	0.94	0.91	0.95	0.96	0.97	0.95	0.98

Figure 4.8 illustrates the actual (blue line) versus predicted CPU utilization (red line) achieved by the hybrid HSTEL model. Timestamp in minutes and CPU utilization in percentage are mapped on the X-Axis and Y-Axis, respectively. The prediction results of the HSTEL model

are nearly close to the actual CPU utilization, which shows the satisfactory performance of the proposed model. The predicted results are dependent on the dynamics of resource usage. More specifically, the CPU utilization prediction results achieved through the hybrid HSTEL model have been tested through RMSE, MSE, MAE, and MAPE that are recorded as 0.58, 0.34, 0.087, 0.43, respectively. Moreover, it shows R2 accuracy of 0.95 on CPU utilization prediction. For instance, it can be visible from the points 600th, 1800th, and 2900th in the plot, where we can observe that there are high error spikes. These error spikes are uncertain and a result of sudden changes in CPU utilization. So, the model did not predict these points well because these values are outliers and do not follow usage patterns in the dataset.

In addition to, Figure 4.9 depicts the comparison between actual and predicted CPU utilization for one-day prediction. It can be seen the correspondence of predicted values to the actual value that enlighten the accuracy of hybrid HSTEL model.

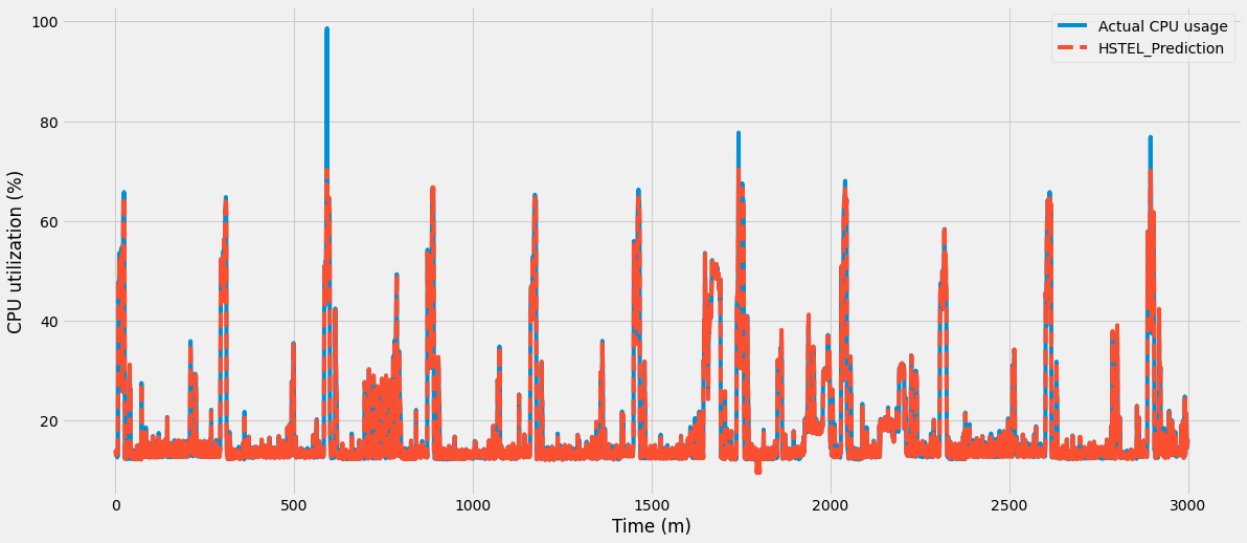

Figure 4.8: Comparison of actual and predicted CPU utilization through HSTEL model

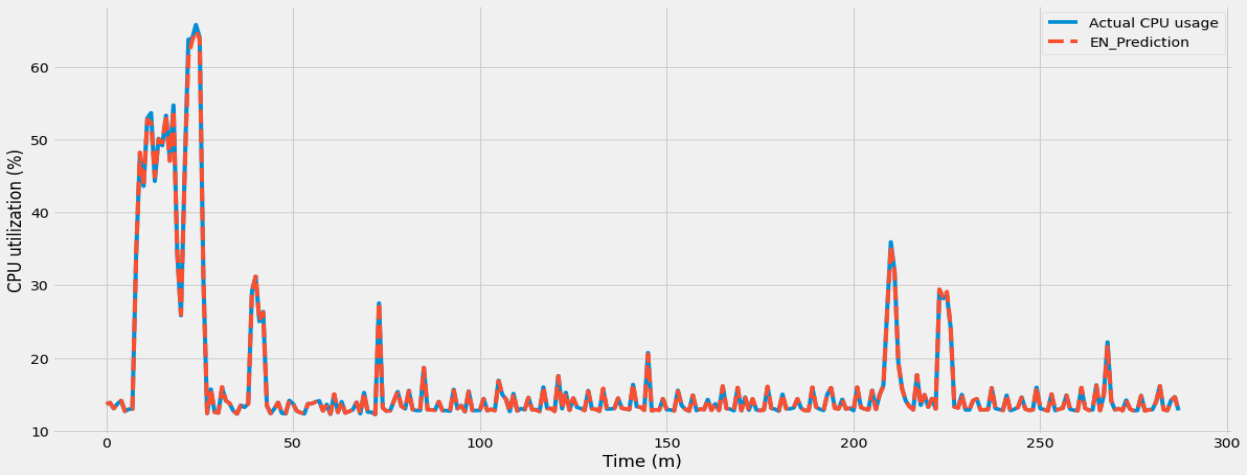

Figure 4.9: Comparison of actual and predicted CPU utilization for one-day prediction through HSTEL model

On the other side, Figure 4.10 depicts the MAPE error rate of CPU utilization for 400 predicted values, which proclaim the MAPE values are significantly less and within the range of 0 to 23. These error measures are minimal as compared to the base models.

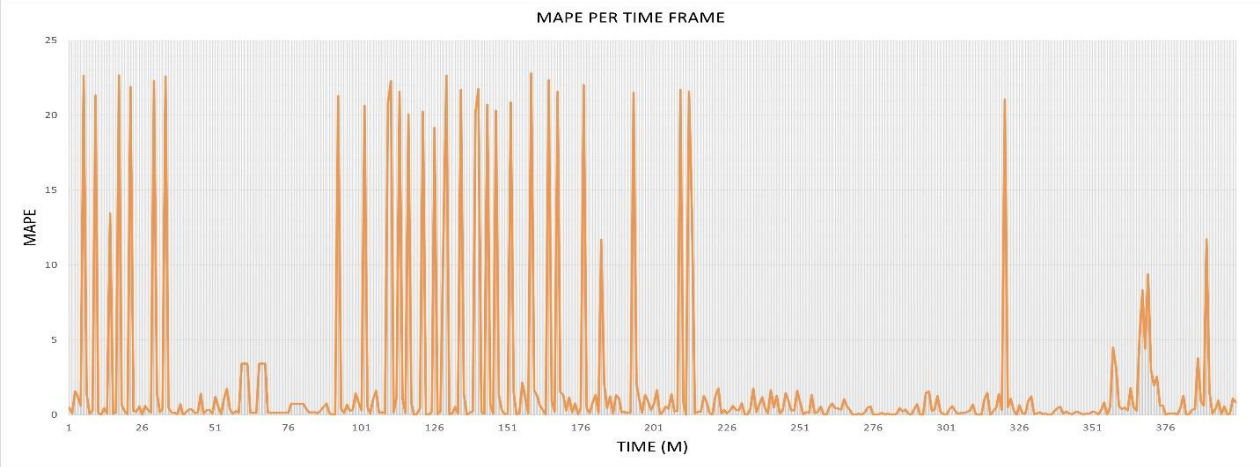

Figure 4.10: Calculated MAPE on CPU utilization prediction attribute through HSTEL model

Figure 4.11 highlights the correspondence of actual and predicted memory utilization, it presents the accuracy of hybrid model. The prediction results of memory utilization are more accurate than CPU prediction because CPU utilization has more fluctuation and more dynamics than memory utilization. The hybrid HSTEL model shows very less RMSE, MSE, MAE, and MAPE that are 0.15,0.023, 0.041, and 0.159, respectively. Moreover, it achieved almost 0.98 R2 accuracy for memory usage prediction.

Furthermore, Figure 4.12 illustrates the one-day prediction results of the model while plotting actual versus predicted memory utilization. Figure 4.13 depicts the MAPE for 400 timestamps, whereas the values are less and within the range of 0 to 14. From the achieved results, the hybrid HSTEL model shows satisfactory performance with higher accuracy and lower error while predicting the multi-attribute network resource utilization.

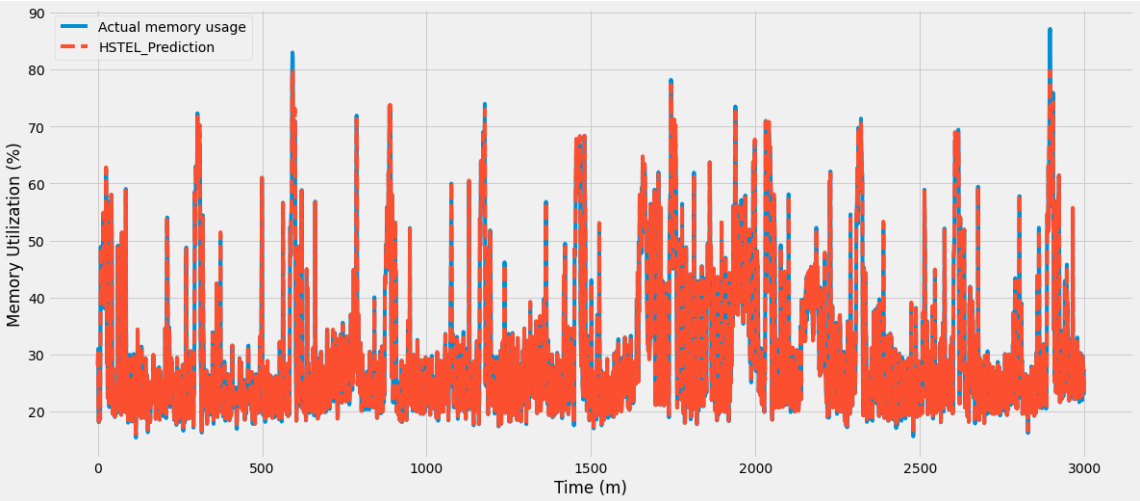

Figure 4.11: Comparison of actual and predicted memory utilization through HSTEL model

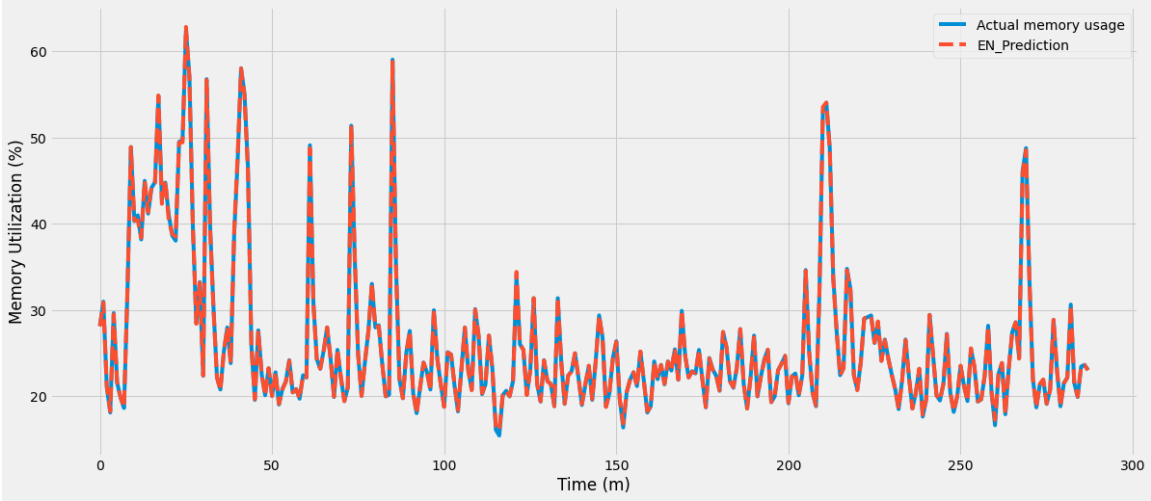

Figure 4.12: Comparison of actual and predicted memory utilization for one-day prediction through HSTEL model

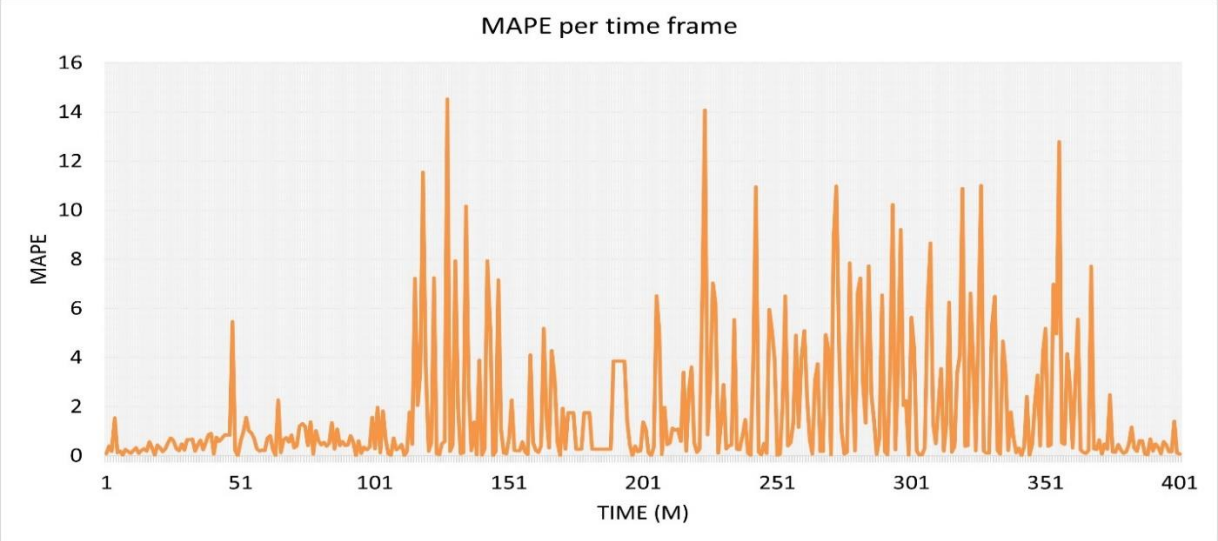

Figure 4.13: Calculated MAPE on memory utilization prediction attribute through HSTEL model

Table 5 describes the performance metrics obtained by each model during the validation, which are the results of mid-term prediction. It can be clearly seen that the proposed HSTEL model's performance is better compared to other individual models. While predicting CPU usage,

the STEL model has obtained 0.60, 0.38, 0.15, 1.10 error measures RMSE, MSE, MAE, and MAPE, respectively. On the other side, it achieved a 0.94 R2 accuracy score. The second-best performer in our study is the XGBoost model, which was the reason why we chose it for the meta learner model in our hybrid approach. XGBoost achieved 0.93 R2 accuracy and 0.84, 0.71, 0.82, 1.29 error measures RMSE, MSE, MAE and MAPE, respectively. Furthermore, the HSTEL model performance on memory utilization prediction is also adequate. It shows 0.96 accuracy and 0.60, 0.383, 01.5, and 0.73 RMSE, MSE, MAE and MAPE, correspondingly.

Table 5: Evaluation metrics of mid-term multi-attribute network resource utilization prediction by proposed and individual models

	CPU utilization Prediction				Memory utilization prediction			
	GBM	XGBoost	Catboost	HSTEL model	GBM	XGBoost	Catboost	HSTEL model
RMSE	0.98	0.84	0.914	0.732	0.723	0.62	0.69	0.60
MSE	0.96	0.71	0.83	0.535	0.523	0.43	0.45	0.383
MAE	0.95	0.825	0.875	0.715	0.188	0.1737	0.558	0.15
MAPE	2.157	1.293	1.387	1.102	1.01	0.972	1.713	0.73
R2	0.90	0.93	0.92	0.94	0.94	0.95	0.93	0.96

Figure 4.14 highlights the prediction results of all models versus actual CPU utilization historical data. The hybrid STEL model outperformed all the individual models. it is visible clearly in the plot, XGBoost performs quite better, Catboost perform little over-prediction and under prediction than actual. In some cases, GBM also predicts over than the actual CPU utilization. This is the beauty of ensemble learning techniques, where the following model can correct the errors of the previous models. The next model learned the experience of the prior ML models and corrected them accordingly.

Therefore, in our HSTEL approach, GBM and Catboost models were used as base-learners and XGBoost as a meta-model. So, XGBoost learned from the Catboost and GBM experience and corrected the wrongly predicted instances in the final prediction. Hence, the prediction results

achieved through HSTEL model are validated and shown satisfactory performance compared to others.

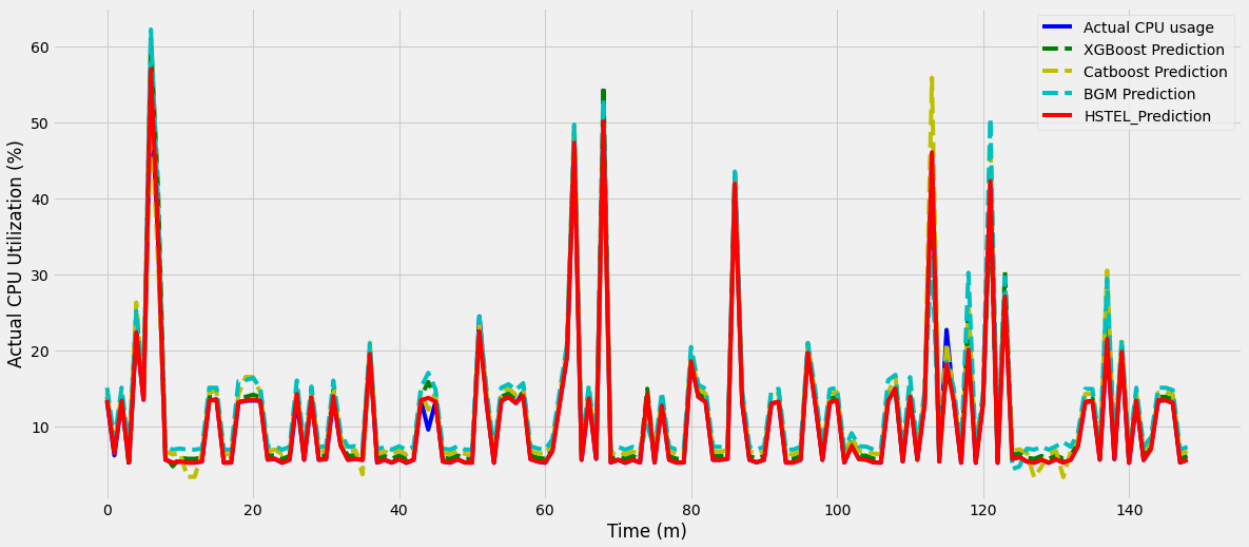

Figure 4.14: Comparison of CPU utilization prediction through GBM, XGBoost, Catboost and HSTEL model

Further, the memory utilization prediction results of all the models are depicted in Figure 4.15, Where the HSTEL model shows satisfactory performance compared to others. It offers almost 98% and 96% accuracy on short-term and mid-term memory utilization prediction, respectively. Moreover, all the models perform better on memory utilization prediction, but GBM and Catboost performs over-prediction in a few cases. In addition, it achieved higher accuracy and lower error metrics which are the critical factors while validating the performance of the regressor models.

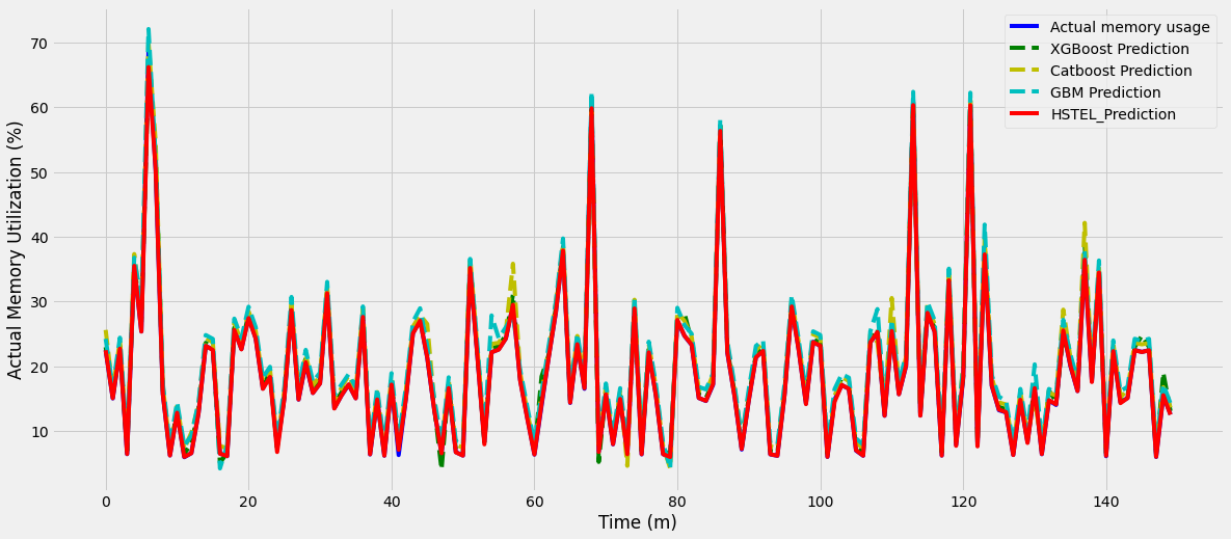

Figure 4.15: Comparison of memory utilization prediction through GBM, XGBoost, Catboost and HSTEL model

Table 6 highlights the comparison of the HSTEL model with various other state-of-art methods, which shows the superiority of our work. It offers 0.96 and 0.98 R2 accuracy and lower other performance metrics as compared to others. On the other hand, it performed multi-attribute prediction, but most of the literature models perform only one attribute prediction. Another advantage of our model is that it is faster and consumes fewer resources than LSTM, CNN, and deep learning approaches. Moreover, we have also performed hourly resource prediction. However, all these results demonstrate that the HSTEL model is a better choice for forecasting network utilization to manage core cloud resources.

Table 6: Comparison of Proposed HSTEL model with existing approaches

Paper#	Technique	Target-feature	MAPE	MSE	RMSE	R_2	Accuracy
[57]	CNN-LSTM hybrid model	CPU	0.733	1.116	0.83	-	-
[58]	LRT SVRT	CPU Memory	14.9 1.42	3.28 0.084	- -	0.73 0.91	71% 89%
[59]	Hybrid Adaptive approach	CPU	2.57	-	9.13	-	-
[60]	Integrated LightGBM	CPU	0.933	-	-	0.91	-
[62]	NN LR	CPU	0.683 0.363	-	9.6 10.41	0.93 0.94	80%
[63]	FFNN	CPU	-	-	0.80	-	-
[64]	LSTM	CPU	-	5.59	-	-	-
Proposed HSTEL model	Stacking Ensemble	CPU Memory	0.53 0.159	0.34 0.023	0.56 0.153	0.96 0.98	-

Table 7: Recorded training time of various ML and DL models

Models	Computation Time (s)
MLP	373.61
SVR	281.85
LSTM	5334.78
XGBoost	128.9
Catboost	181.34
GBM	194.16
HSTEL model	275.88

Several experiments have been conducted to verify that the DL model takes more training time than ML models. Table 7 illustrates the computation time comparison of different ML and DL models. It can be observed that the DL models take more computational time and resources due to their complex structure than ML models.

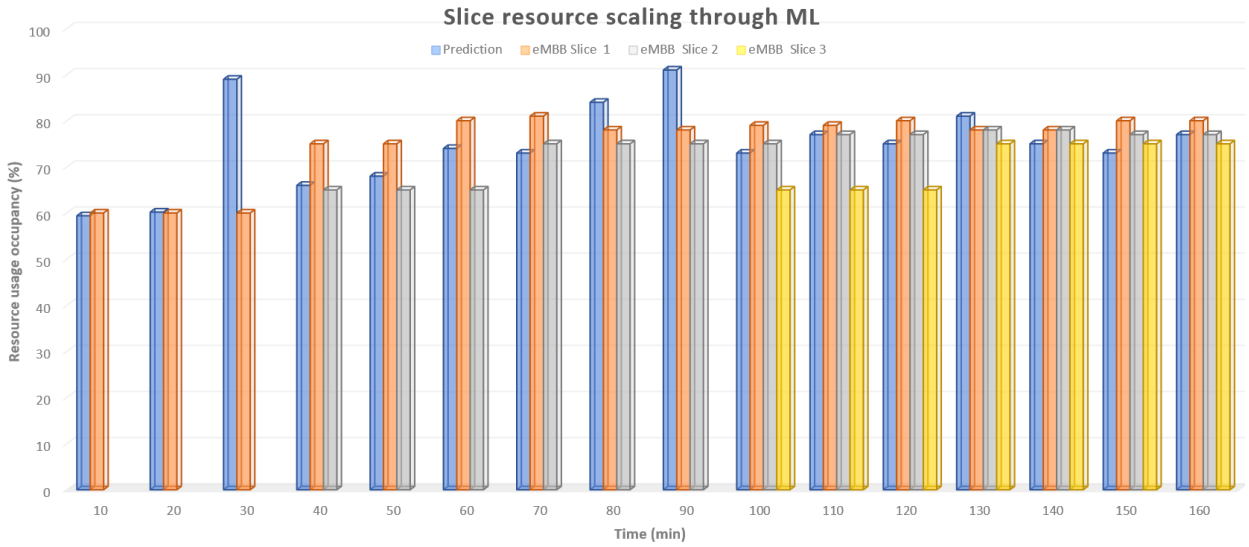

Figure 4.16: Network slice resource scaling through HSTEL model prediction results

Figure 4.16 shows the slice resource scaling based on ML HSTEL model resource utilization prediction. It can instantiate a new service whenever the resources are overloaded. The ML prediction results prevent resource overloading and enable the NOs to deploy more resources. Due to ML model predictions, cloud administrators can proactively manage resources to fulfill SLA.

The results mentioned above reveal that our HSTEL model outperformed the other models for network resource utilization prediction. We have performed short-term and mid-term forecasts for cloud VNFs resources based on minutely and hourly prediction. So, accurate prediction is essential to manage the cloud resources and ensure QoS automatically. The inaccurate prediction results lead to service degradation and compromise the QoE to the customers. The results of our

hybrid approach show that it is a good candidate for forecasting the network resource utilization process. Hence, the IBN platform can use the prediction results of the HSTEL model for autoscaling the core network resources. It guarantees QoS and proactively manages the cloud resources by using future prediction results.

4.3. Results of Anamoly Detection through Hybrid Model

This section explains the results generated through the proposed hybrid and other well-known methods for attack detection and classification from the networks. We have used 70% and 30% ratios for the training and testing of all models. Moreover, the hybrid model is compared with five ML classifiers such as MLP, SVM, RF, Catboost, and XGBoost based on classification accuracy on the same dataset. Besides, accuracy is used as an evaluation metric that shows the overall performance of the ML models. Equation 1 illustrates the mathematics behind the accuracy (A) metric, where T_p presents the number of correctly classified instances as N, F_p shows the number of cases wrongly classified as N, T_n highlights the number of correctly classified instances Not-N, and F_n denotes the number of wrongly classified instances Not-N.

$$A = \frac{T_p + T_n}{T_p + F_n + F_p + T_n} \qquad 1)$$

We have trained and tested our hybrid model on both explained datasets. Figure 4.17 highlights the training and testing results of the hybrid model on the first dataset. Figure 4.17 (a) denotes the training testing loss of the model while performing only 50 epochs in training and testing. The loss is relatively small, and accuracy is good, almost 97.5%. When we have increased the epochs to 200, it achieves 98.5% accuracy, which remains similar after that. Thereby, the model shows 97% accuracy in the final validation phase. Hence, the hybrid model claims outstanding performance and efficiency in performing network attack detection and classification. Furthermore,

the hybrid model with the same setting and parameters is also applied to another dataset. It outperformed individual models as well as SVM and MLP. A model that takes less time, consumes fewer resources, and shows better performance is considered suitable for real-time application.

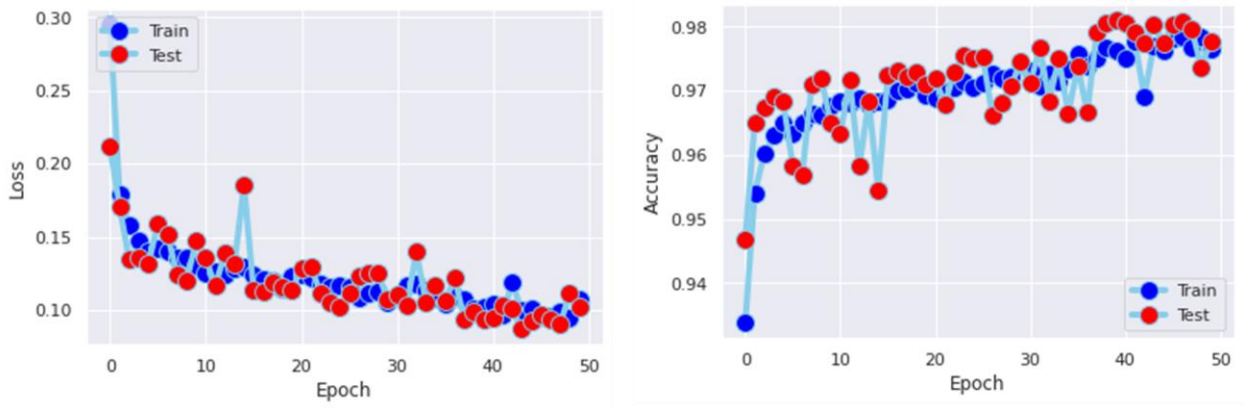

Figure 4.17: Hybrid EL model classification results on considered dataset a) shows the plot of model loss during training and testing phase b) presents accuracy of the hybrid model during training and testing

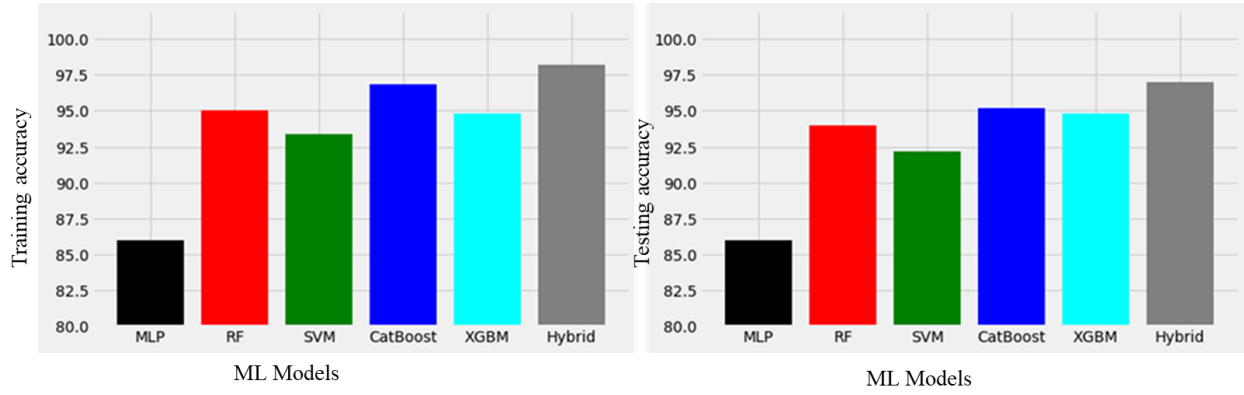

Figure 4.18: Comparison among hybrid model and other ML models while performing attack Classification

The results presented in Figure 4.18 illustrate the performance of the hybrid model compared to other state-of-the-art models. It can observe that the hybrid model achieved 98.35% and 97%

accuracy in the training and testing phase, respectively. Besides, the individual RF, Catboost, and XGBoost have 95%, 94.5%, and 97% accuracy during training. Also, these models achieved 93%, 94.2%, and 95% accuracy in the testing phase. To validate the proposed model, we have also trained multilayer perceptron (MLP) and support vector machine (SVM) on the studied dataset. SVM and MLP show 93% and 86% accuracy in the training and testing phase. Further, MLP is not efficient and takes more time and resources as compared to other models. However, the proposed model's superiority indicates the model's effectiveness while performing attack classification on two different datasets. It is incredibly efficient in terms of speed and resources consumption. Besides, the IBN decision engine uses the attack detection and classification results generated by the hybrid model, which mitigates the abnormal flow from the system.

Chapter 6

Conclusions

Future networks can accommodate an extensive range of innovative services with distinct QoS requirements regarding latency and bandwidth. The traditional networks cannot handle these diverse variety of services over the same infrastructure. NFV and SDN technologies allow NOs to virtualize and program their networks according to their needs. So, these technologies enable NOs to create multiple isolated networks over the same infrastructure and provide dedicated resources to each service; this concept is called network slicing. Network slicing is a primary use case of the 5G network that accommodates many innovative services. So, the orchestration and management

of multidomain network slices is still a challenging task. However, a well-designed automated platform is needed that can automatically handle e2e network slicing and ensure service guarantee. This work explained the IBN platform and a novel EL-based network data analytics mechanism for autonomous orchestration and management of network resources. IBN system follows a one-touch process where the user inputs abstract slice QoS requirements. In return, the system automatically translates them into policies and deploys them over the infrastructure with the help of various NFVOs and domain Controllers. Additionally, the OSM platform is responsible for deploying and managing the core domain VNFs, and the FlexRAN controller handles the slicing of the RAN domain resources. It automatically designs, activate, monitor, and deactivate the network slices. It can manage the slice LCM in an automated fashion. Moreover, it is also able to handle and manage the network slices over the multidomain infrastructure. In addition, multiple tests have been performed by creating several slices of core and RAN domain through our mechanism, showing satisfactory results in resource stability, automated resource provisioning, customization, and resource assurance.

On the other side, network data analytics with IBN has been used to solve two major network issues: network resource utilization prediction to perform autoscaling of resources and ensure service guarantee and anomaly detection and mitigation. A novel hybrid HSTEL model has been implemented for future VNFs load prediction. The Catboost, GBM, and XGBoost models are implemented through the stacking EL technique for getting better prediction results. The proposed HSTEL model outperformed other state-of-art algorithms while comparing the prediction results based on the error metrics. The decision engine of IBN uses these prediction results to guarantee QoS assurance and autoscaling of core VNFs. The anomaly detection mechanism is accomplished by combining ML models such as Random Forest, Catboost, and

XGBoost through the voting EL approach. The ML models result can be used by the decision engine of the IBN platform, which takes action to perform autoscaling of resources and mitigate the malicious flow from the network.

Bibliography

[1] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[2] 5GPPP, "View on 5G Architecture: white paper)." [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/ 5G-PPP-5G-Architecture-Whi%te-Paper-Jan-2018-v2.0.pdf.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, 2017.

[4] K. Abbas *et al.*, "An efficient SDN-based LTE-WiFi spectrum aggregation system for heterogeneous 5G networks," *Trans. Emerg. Telecommun. Technol.*, p. e3943, 2020.

[5] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Networks*, vol. 167, p. 106984, 2020.

[6] S. Kekki *et al.*, "MEC in 5G networks," *ETSI White Pap.*, no. 28, pp. 1–28, 2018.

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

[8] I. Afolabi, T. Taleb, P. A. Frangoudis, M. Bagaa, and A. Ksentini, "Network Slicing-Based Customization of 5G Mobile Services," *IEEE Netw.*, vol. 33, no. 5, pp. 134–141, 2019.

[9] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, 2017.

[10] T. S. 3GPP, "5G; Management and orchestration; Concepts, use cases and requirements." [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128530/15.00.00_60/t%s_128530v150000p.pdf.

[11] 3GPP, "Telecommunication management; Study on management and orchestration of network slicing for next generation network." [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3091.

[12] SAMSUNG, “Technical white paper: network slicing.” [Online]. Available: https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/network-slicing/200420_Samsung_Network_Slicing_Final.pdf.

[13] OSM, “Open Source Mano.” [Online]. Available: https://osm-download.etsi.org/ftp/Documentation/201902-osm-scopewhite-%paper/#!02-osm-scope-and-functionality.md

[14] ETSI, “Network Transformation; (Orchestration, Network and Service Management Framework).” Available: https://www.etsi.org/images/files/ETSIWhitePapers/ETSI_White_Paper_ Netw%ork_Transformation_2019_N32.pdf

[15] V. Q. Rodriguez, F. Guillemin, and A. Boubendir, “5G E2E Network Slicing Management with ONAP,” in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 87–94.

[16] H. Packard, “Network slice configuration and service slice lifecycle management.” [Online]. Available:https://h20195.www2.hpe.com/v2/Getdocument.aspx?docname=a00065972enw&skiphtml=1&&&.

[17] A. Boubendir, F. Guillemin, S. Kerboeuf, B. Orlandi, F. Faucheux, and J.-L. Lafragette, “Network slice life-cycle management towards automation,” in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 709–711.

[18] V. P. Kafle, P. Martinez-Julia, and T. Miyazawa, “Automation of 5G Network Slice Control Functions with Machine Learning,” *IEEE Commun. Stand. Mag.*, vol. 3, no. 3, pp. 54–62,

2019, doi: 10.1109/MCOMSTD.001.1900010.

[19] S. IETF, “Network Slicing Management and Orchestration.” [Online]. Available: https://tools.ietf.org/id/draft-flinckslicing-management-00.xml#rfc.se%ction.3.

[20] IETF, “Intent-based Networking.” [Online]. Available: https://tools.ietf.org/html/draft-irtf-nmrg-ibn-conceptsdefinitions-01%.

[21] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, “A survey on intent-driven networks,” *IEEE Access*, vol. 8, pp. 22862–22873, 2020.

[22] Cisco, “Intent-based Networking.” [Online]. Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprisenetwo%rks/digital-network-architecture/nb-09-intent-networking-wp-cteen.pdf.

[23] Huawei, “Intent-Driven Network.” [Online]. Available: https://carrier.huawei.com/~/media/CNBG/Downloads/Spotlight/all-cloud-n%etwork-towards-5g/idn-en.pdf.

[24] Apstra, “Intent-Based Networking: A Next-Gen Vision for the Next-Gen Network.” [Online]. Available: https://go.apstra.com/white-paper-apstra-intent-based-networking.

[25] Y. Wei, M. Peng, and Y. Liu, “Intent-based networks for 6G: Insights and challenges,” *Digit. Commun. Networks*, 2020.

[26] E. Zeydan and Y. Turk, “Recent Advances in Intent-Based Networking: A Survey,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[27] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, “Intelligent network

data analytics function in 5G cellular networks using machine learning," *J. Commun. Networks*, vol. 22, no. 3, pp. 269–280, 2020.

[28] E. Pateromichelakis *et al.*, "End-to-end data analytics framework for 5G architecture," *IEEE Access*, vol. 7, pp. 40295–40312, 2019.

[29] M. Fernández-Delgado, M. S. Sirsat, E. Cernadas, S. Alawadi, S. Barro, and M. Febrero-Bande, "An extensive experimental survey of regression methods," *Neural Networks*, vol. 111, pp. 11–34, 2019, doi: 10.1016/j.neunet.2018.12.010.

[30] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1249, 2018.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[32] R. Polikar, "Ensemble learning," *Ensemble machine learning*. Springer, pp. 1–34, 2012.

[33] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Stand. Mag.*, vol. 2, no. 1, pp. 60–65, 2018.

[34] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network Slice Lifecycle Management for 5G Mobile Networks: An Intent-based Networking Approach," *IEEE Access*, 2021.

[35] H. D. R. Albonda and J. Pérez-Romero, "An efficient RAN slicing strategy for a heterogeneous network with eMBB and V2X services," *IEEE access*, vol. 7, pp. 44771–44782, 2019.

[36] ONAP, "ONAP: open networking automation platform." [Online]. Available:

https://www.onap.org/.

[37] K. Katsalis, N. Nikaein, and A. Huang, "JOX: An event-driven orchestrator for 5G network slicing," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–9.

[38] Cloudify, "Cloudify: A open source Network Orchestrator." [Online]. Available: https://cloudify.co/.

[39] OpenBaton, "OpenBaton: NFV MANO-based framework." [Online]. Available: https://openbaton.github.io/.

[40] C. Rotsos *et al.*, "Network service orchestration standardization: A technology survey," *Comput. Stand. Interfaces*, vol. 54, pp. 203–215, 2017.

[41] ETSI, "ETSI Zero-touch Network & Service Management ," *OSIA Stand. Technol. Rev.*, vol. 31, no. 4, pp. 21–25, 2018.

[42] F. Meneses, M. Fernandes, D. Corujo, and R. L. Aguiar, "SliMANO: an expandable framework for the management and orchestration of end-to-end network slices," in *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, 2019, pp. 1–6.

[43] X. Li, R. Ni, J. Chen, Y. Lyu, Z. Rong, and R. Du, "End-to-End Network Slicing in Radio Access Network, Transport Network and Core Network Domains," *IEEE Access*, vol. 8, pp. 29525–29537, 2020.

[44] S. E. Elayoubi, S. Ben Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, 2019.

[45] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, 2020.

[46] T. A. Khan, A. Mehmood, J. J. D. Rivera, and W.-C. Song, "Machine Learning Approach for Automatic Configuration and Management of 5G Platforms," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–6.

[47] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34538–34551, 2020.

[48] H. Xiang, S. Yan, and M. Peng, "A realization of fog-RAN slicing via deep reinforcement learning," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 4, pp. 2515–2527, 2020.

[49] C. Benzaid and T. Taleb, "AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions," *IEEE Netw.*, vol. 34, no. 2, pp. 186–194, 2020.

[50] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, "Machine learning-based zero-touch network and service management: A survey," *Digit. Commun. Networks*, 2021.

[51] J. Networks, "Network Automation and Orchestration," pp. 1–8, 2015, [Online]. Available: http://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000541-en.pdf.

[52] J. Networks, "The Juniper Networks Platform for Agile Service Delivery," pp. 1–10.

[53] S. Ratkovic, "Apstra AOS Architecture Overview." [Online]. Available: https://go.apstra.com/hubfs/White_Papers/Apstra_White_Paper_-

_AOS_Architecture_Overview.pdf?__hssc=62805235.1.1601856000000&__hstc=62805235.2f3f33a24b44870ec4a577029c49e44b.1601856000000.1601856000000.1601856000000.1&hsCtaTracking=3de3a15c-91db-46d3-9cdb-0d1ddeab0e5f%7Ce04ef85b-3bbd-4ea0-b291-ab2ab7f074f5

[54] RADCOM, "Data analytics and containerized service assurance for 5G RADCOM ' s Network Data Analytics Function (NWDAF)." [Online]. Available: https://www.radcom.com/uploads/pdf/RADCOM's%20NWDAF%20Solution_ACE.pdf

[55] Sandvine, "An automated, containerized 5G assurance platform with AI-driven insights for standalone 5G network operations." [Online]. Available: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2020/Datasheets/5G/Sandvine_DS_5G%20Service%20Intelligence%20Engine%20AW%2020210713.pdf

[56] Ericssion, "Accelerating the adoption of AI in programmable 5G networks," no. July. pp. 1–27, 2021. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/accelerating-the-adoption-of-ai-in-programmable-5g-networks.

[57] C. A. Solution and S. I. Challenges, "NITRO Mobile." [Online]. Available:https://www.viavisolutions.com/en-us/products/virtualized-assurance-analytics

[58] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model," *Neural Comput. Appl.*, pp. 1–13, 2021.

[59] L. Abdullah, H. Li, S. Al-Jamali, A. Al-Badwi, and C. Ruan, "Predicting multi-attribute host resource utilization using support vector regression technique," *IEEE Access*, vol. 8,

pp. 66048–66067, 2020.

[60] W. Iqbal, J. L. Berral, A. Erradi, D. Carrera, and others, “Adaptive prediction models for data center resources utilization estimation,” *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 4, pp. 1681–1693, 2019.

[61] H. Xia, X. Wei, Y. Gao, and H. Lv, “Traffic prediction based on ensemble machine learning strategies with bagging and lightgbm,” in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.

[62] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, and X. Costa-Perez, “AI-Based Autonomous Control, Management, and Orchestration in 5G: From Standards to Algorithms,” *IEEE Netw.*, vol. 34, no. 6, pp. 14–20, 2020.

[63] M. Duggan, K. Mason, J. Duggan, E. Howley, and E. Barrett, “Predicting host CPU utilization in cloud computing using recurrent neural networks,” *2017 12th Int. Conf. Internet Technol. Secur. Trans. ICITST 2017*, pp. 67–72, 2018, doi: 10.23919/ICITST.2017.8356348.

[64] S. Islam, J. Keung, K. Lee, and A. Liu, “Empirical prediction models for adaptive resource provisioning in the cloud,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.

[65] J. Kumar and A. K. Singh, “Workload prediction in cloud using artificial neural network and adaptive differential evolution,” *Futur. Gener. Comput. Syst.*, vol. 81, pp. 41–52, 2018.

[66] S. Gupta and D. A. Dinesh, “Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks,” in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2017, pp. 1–6.

[67] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018.

[68] S. Garg and S. Batra, "A novel ensembled technique for anomaly detection," *Int. J. Commun. Syst.*, vol. 30, no. 11, p. e3248, 2017.

[69] J. Hu, X. Yu, D. Qiu, and H. H. Chen, "A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection," *IEEE Netw.*, vol. 23, no. 1, pp. 42–47, 2009, doi: 10.1109/MNET.2009.4804323.

[70] K. Ghosh, S. Neogy, P. K. Das, and M. Mehta, "Intrusion Detection at International Borders and Large Military Barracks with Multi-sink Wireless Sensor Networks: An Energy Efficient Solution," *Wirel. Pers. Commun.*, vol. 98, no. 1, pp. 1083–1101, 2018, doi: 10.1007/s11277-017-4909-5.

[71] D. Jiang, Z. Xu, P. Zhang, and T. Zhu, "A transform domain-based anomaly detection approach to network-wide traffic," *J. Netw. Comput. Appl.*, vol. 40, no. 1, pp. 292–306, 2014, doi: 10.1016/j.jnca.2013.09.014.

[72] S. Sriram, R. Vinayakumar, M. Alazab, and K. P. Soman, "Network flow based IoT botnet attack detection using deep learning," *IEEE INFOCOM 2020 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2020*, pp. 189–194, 2020, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

[73] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An efficient SDN-Based DDoS attack detection and rapid response platform in vehicular networks," *IEEE Access*, vol. 6, pp.

44570–44579, 2018, doi: 10.1109/ACCESS.2018.2854567.

[74] F. Pasqualetti, S. Member, F. Dör, S. Member, and F. Bullo, "2713 Attack Detection and Identi fi cation in Cyber-Physical Systems *[Bullo, IEEE TAC2013]Attack Detect. Identif. Cyber-Physical Syst.*, vol. 58, no. 11, pp. 2715–2729, 2013.

[75] 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)*, 2020, pp. 19–24.

[76] A. Rafiq, A. Mehmood, T. A. Khan, K. Abbas, M. Afaq, and S. W. Cheol, "Intent-Based End-to-End Network Service Orchestration System for Multi-Platforms," *Sustainability*, vol. 12, no. 7, p. 2782, 2020.

[77] K. Abbas, M. Afaq, T. A. Khan, A. Rafiq, and W.-C. Song, "Slicing the Core Network and Radio Access Network Domains through Intent-Based Networking for 5G Networks," *Electronics*, vol. 9, no. 10, p. 1710, 2020.

[78] T. A. Khan, A. Muhammad, K. Abbas, and W.-C. Song, "Intent-based networking platform: an automated approach for policy and configuration of next-generation networks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 1921–1930.

[79] T. A. Khan, A. Mehmood, J. J. D. Ravera, A. Muhammad, K. Abbas, and W.-C. Song, "Intent-Based Orchestration of Network Slices and Resource Assurance using Machine Learning," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–2.

[80] O. S. MANO, “OSM Performance Management.” [Online]. Available: https://osm.etsi.org/wikipub/index.php/OSM_ Performance Management.

[81] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, “FlexRAN: A flexible and programmable platform for software-defined radio access networks,” in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016, pp. 427–441.

[82] Mosaic5G, “Network sharing using FlexRAN.” [Online]. Available: https://gitlab.eurecom.fr/mosaic5g/mosaic5g/-/wikis/tutorials/slicing.

[83] Gitlab:MOSAIC5G, “elasticmon manual.” [Online]. Available: https://gitlab.eurecom.fr/mosaic5g/mosaic5g/-/wikis/ tutorials/elasticmo%n-manual.

[84] L. Chen, M. Xian, and J. Liu, “Monitoring System of OpenStack Cloud Platform Based on Prometheus,” *Proc. - 2020 Int. Conf. Comput. Vision, Image Deep Learn. CVIDL 2020*, no. Cvidl, pp. 206–209, 2020, doi: 10.1109/CVIDL51233.2020.0-100.

[85] M. Yang and M. Huang, “An microservices-based openstack monitoring tool,” *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2019-October, pp. 706–709, 2019, doi: 10.1109/ICSESS47205.2019.9040740.

[86] Materna, “Dataset.” [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-13-materna/.

[87] F. Divina, A. Gilson, F. Goméz-Vela, M. G. Torres, and J. F. Torres, “Stacking ensemble learning for short-term electricity consumption forecasting,” *Energies*, vol. 11, no. 4, pp. 1–31, 2018, doi: 10.3390/en11040949.

[88] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002.

[89] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R Packag. version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[90] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv Prepr. arXiv1810.11363*, 2018.

[91] A. Cutler, "Random Forest for Regression and Classification," *Books*, vol. 1, no. 1, p. 21, 2010, [Online]. Available: internal-pdf://semisupervised-3254828305/semisupervised.ppt.

[92] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[93] A. A. Q. Doan, "Intro To Random Forest." [Online]. Available: https://home.csulb.edu/~tebert/teaching/lectures/551/random_forest.pdf

[94] S. Hettich and S. D. Bay, "The UCI KDD Archive [http://kdd. ics. uci. edu]. Irvine, CA: University of California," *Dep. Inf. Comput. Sci.*, vol. 152, 1999.

[95] OpenAirInterface, "OAI: An open-source community." [Online]. Available: https://www.openairinterface.org.