Cape Peninsula
University of Technology

5G NEW RADIO AND FOG COMPUTING SCALABILITY AND QOS MANAGEMENT

By

Vuyo Pana

Thesis submitted in partial fulfilment of the requirements for the degree

Doctor of Engineering: Electrical, Electronic and Computer Engineering

In the Faculty of Engineering and Built Environment

At the Cape Peninsula University of Technology

Supervisor: Prof V Balyan

Co-supervisor: Dr OP Babalola

Bellville

January 2024

CPUT copyright information

The thesis may not be published either in part (in scholarly, scientific, or technical journals), or as a whole (as a monograph), unless permission has been obtained from the University

DECLARATION

I, Vuyo Pana, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

31 January 2025

Signed **Date**

ABSTRACT

This thesis investigates the advancements in fifth-generation (5G) mobile technology, with a particular focus on the 5G Fog Radio Access Network (5G-FRAN) architecture and resource optimization. The proliferation of Internet of Things (IoT) deployments has heightened the demand for latency-efficient applications positioned closer to end-users, necessitating a critical evaluation of resource management in 5G networks.

The research begins with a comprehensive literature review of Cloud Radio Access Network (CRAN) architectures and their variations. CRAN is pivotal for reducing operational and maintenance costs for mobile network operators. However, the exponential growth of IoT devices, surpassing 900 million by 2023, exposes the limitations of CRAN's centralized network structure, particularly its vulnerability to fronthaul congestion. To address these challenges, the 5G-FRAN architecture offers significant enhancements, including improved mobility, optimized radio resource allocation, superior service quality, and low latency. Despite these advancements, challenges such as resource allocation, end-to-end latency, and energy efficiency persist. Consequently, this thesis proposes machine learning-based resource management schemes to mitigate these challenges and meet industrial requirements effectively.

Thus, a novel cell association technique is developed to manage bandwidth in 5G-FRANs by accounting for subscriber mobility. The proposed approach employs probability distributions of time intervals and cell identifications to construct a Hidden Semi-Markov Model (HsMM) that predicts the optimal next cell for meeting downstream rate requirements. The model addresses critical issues, including ping-pong effects, missing data, and restrictive wireless environments, thereby enhancing user traffic prediction. Results demonstrate that the HsMM-based method achieves high accuracy with respect to observation time intervals and significantly improves user satisfaction compared to existing systems.

Furthermore, Eigen decomposition (ED)-based feature extraction techniques, specifically Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), which are applied to attributes derived from the dataset. The extracted features are then utilized in machine learning models to detect and classify service segments. Hence, in the thesis a detailed description of the dataset and the implementation of the ED-based feature extraction techniques is provided. The workflow of the machine learning models is illustrated using block diagrams, followed by an in-depth explanation of the experimental design. Experiments were conducted using a range of machine learning algorithms, including Decision Tree (DT),

Support Vector Machines (SVM), K-Nearest Neighbours (K-NN), Random Forest (RF), Artificial Neural Networks (ANN), and their combinations with PCA and LDA. These models were tested on fog computing application datasets featuring multi-class classification scenarios and noise-injected environments.

Thereafter, a thorough analysis of the experimental results, focusing on the impact of dimensionality reduction on model accuracy, efficiency, and resilience to noise in various machine learning configurations.

This thesis makes several notable contributions. It provides an in-depth analysis of the latest 5G network architectures and demonstrates the efficacy of HsMM in optimizing limited spectrum resources using machine learning techniques. Additionally, the study introduces and evaluates two ED-based feature extraction techniques (PCA and LDA) across multiple machine learning models, significantly reducing computational complexity while improving model performance. These techniques enable efficient processing of large datasets, enhancing the accuracy, reliability, and scalability of machine learning models in 5G-FRAN applications.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to:

- Dr Oluwaseyi P. Babalola for his invaluable guidance and insightful input throughout this study.
- Prof. Vipin Balyan for his unwavering encouragement and persistence, which kept me motivated.
- My family, friends, and colleagues for their constant support and understanding during this journey.
- The Ubuntu Post Graduate Forum played a crucial role in early part of doctoral studies I want say to my peer scholars thank you.
- The Centre for Postgraduate Studies (CPS) for their invaluable support and the wealth of information provided during several workshops.
- I want to take a moment to express my heartfelt gratitude to the Ubuntu Post Graduate Forum and my fellow peer scholars. Your support, collaboration, and shared insights played a pivotal role in the early stages of my doctoral journey. The sense of community and encouragement I found within this group was invaluable, and it has left a lasting impact on my academic and personal growth.
- A special thank you to my study mate, Mario Ligwa, for your unwavering support and camaraderie throughout this journey. Your dedication and encouragement made a significant difference.

Thank you all for your contributions and support, which have been instrumental in the completion of this work.

DEDICATION

To my dear wife, Somila Sabata, and my beloved daughters, Zimi, Mmangaliso, and Thalanda, for their unwavering love and support throughout the period of my studies. Your encouragement and sacrifices have been my greatest motivation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PUBLICATIONS

This study result in the following publications:

1. Pana, V.S., Babalola, O.P. and Balyan, V., 2022. 5G radio access networks: A survey. Array, 14, p.100170. https://doi.org/10.1016/j.array.2022.100170

2. Pana, V., Babalola, O.P., Balyan, V. (2024). HsMM-Based Mobility Aware Cell Association Method for Dynamic Bandwidth Management in 5G-FRANs. In: Lecture Notes in Network and System, vol 991. Springer, Singapore. https://doi.org/10.1007/978-981-97-2550-2 60

GLOSSARY

ABBREVIATIONS	DEFINITION
3GPP	3rd Generation Partnership Project
4IR	Fourth Industrial Revolution
5G	Fifth Generation
ACC	Accuracy
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Access Point
AR	Augmented Reality
ATM	Asynchronous Transfer Mode
AUC	Area Under Curve
BBB	Backtracking Branch-and-Bound
BBF	Broad-Band Forum
BBU	Baseband Unit
BE	Best Effort
BF	Beamforming
BS	Base Station
C2T	Cloud to Things
CAPEX	Capital Expenditure
CB	CPU-Bound
CELL-ID	Cell Identification
CN	Core Network

CO	Conversational
COM	Cell Outage Management
COMP	Coordinated Multi Point Transmission
COS	Class of Service
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CPUT	Cape Peninsula University of Technology
CRAN	Cloud Radio Access Network
CRRA	Cooperative Radio Resource Allocation
CRRM	Cooperative Radio Resource Management
CRSP	Collaboration Radio Signal Processing
CSI	Channel Status Information
CSP	Cloud Service Provider
D2D	Device-to-device
DB	Decibels
DHET	Department of Higher Education and Training
DIFFSERV	Differentiated Service
DRL	Deep Reinforcement Learning
DSP	Digital Signal Processing
DT	Decision Tree
DTMC	Discrete Time Markov Chain
ECOC	Error Correcting Output Code
ED	Eigen Decomposition
ED-FE	Eigen Decomposition Feature Extraction
EHS	Energy Harvesting Sensor

EICIC	Enhanced Inter-cell interference Coordinated
EMBB	Enhanced Mobile Broadband
ERR	Error Rate
ERRH	Enhanced Remote Radio Head
F-AP	Fog Access Point
FCA	Fog Computing Applications
FCN	Fog Computing Node
FDA	Fisher's Discriminant Analysis
FDR	False Discovery Rate
FE	Feature Extraction
FNe	Fog Node
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FRAN	Fog Radio Access Network
FTP	File Transfer Protocol
F-UE	Fog User Equipment
GA-GR	Greedy Algorithm Greedy Routing
GA-RR	Greedy Algorithm Randomized Rounding Routing
GBR	Guaranteed Bit Rate
GCFSA	Gini Coefficient Based Fog Computing Node Selection Algorithm
GNI	Gross National Income
H2H	Human to Human
HCRAN	Heterogeneous Cloud Radio Access Network
HD	High-Definition

HETNET	Heterogeneous Network
HETNETS	Heterogeneous Networks
HMM	Hidden Markov Model
HPN	High Power Nodes
HRRH	High Remote Radio Head
HSMM	Hidden semi–Markov Model
H-SVM	Hierarchical Support Vector Machines
IC	Interference Collaboration
ICA	Independent Component Analysis
ICT	Information and Communication Technology
IDC	International Data Corporation
IEEE	Institute of Electrical and Electronics Engineers
IIOT	Industrial Internet of Things
ILP	Integer Linear Programming
IN	Interactive
INTSERV	Integrated Service
IOT	Internet of Things
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union Standardization Sector
K-NN	K-Nearest Neighbours
LDA	Linear Discriminant Analysis
LPN	Low Power Node
LRRH	Low Remote Radio Head
LSTM	Long Short-Term Memory

LTE	Long Term Evolution
MAB	Multi-Armed Bandit
MAC	Medium Access Control
MATLAB	Matrix Laboratory
MBS	Macro Base Station
MC	Mission Critical
MDP	Markov Decision Process
METIS	Mobile and Wireless Communication Enablers for Twenty-twenty (2020) Information Society
MINLP	Mixed Integer Non-Linear Programming Problem
MIQCP	Mixed Integer Quadratic Constrained Programming
ML	Machine Learning
MMIMO	Massive Multiple in Multiple out
MMTC	Massive Machine-type communication
MMWAVE	Millimetre Wave
MNO	Mobile Network Operator
NFV	Network Function Virtualization
NOMA	Non-Orthogonal multiple access
NR	New Radio
OFDMA	Orthogonal Frequency Division Multiple Access
OPEX	Operational Expenditure
O-RAN	Open Radio Access Network
PC	Principal Component
PCA	Principal Component Analysis
POMDP	Partial Observable Markov Decision Process

PR	Precision Recall
PREC	Precision
PUEA	Primary User Emulation Attack
QOE	Quality of Experience
QOS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RATS	Radio Access Technologies
RF	Random Forest
RL	Reinforcement Learning
RO	Research Objectives
ROAGA	Resource Optimisation Approach Based on a Genetic Algorithm
ROC	Receiver Operating Characteristics Curve
RQ	Research Question
RRH	Remote Radio Head
RSRP	Reference Signal Received Power
RSSI	Receive Signal Strength Indicator
RT	Realtime
SARA	State Action Reward Next
SDG	Sustainable Development Goal
SDN	Software Define Network
SDO	Standard Development Organization
SE	Spectral Efficiency
SLA	Service Level Agreement
SMS	Short Message Service

SNR	Signal to Noise Ratio
SSA	Sub-Saharan Africa
ST	Streaming
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TVWS	Television White Space
UE	User Equipment
UN	United Nations
URLLC	Ultra-reliable and Low-Latency Communication
V2X	Vehicle-to-Everything
VCRAN	Virtual Cloud Radio Access Network
VOD	Video On Demand
VOIP	Voice Over Internet Protocol
VR	Virtual Reality
WMMSE	Weighted Minimal Mean Square Error
WP5D	Working Party 5D

CHAPTER:1 INTRODUCTION

1.1 5G Deployment in the Developing World

Mobile communication now in its fifth generation (5G) is playing a crucial role for the obtainment of the United Nation's (UN) sustainable development goals (SDGs)) [1]. This is clearly indicated in the report [2] report, where digital economy is shown to have benefited the developed world through the use digital data and online media. Whereas for the developing world the lack of infrastructure has a severe impact information and communication technology (ICT) sector. On the other hand, the lament from mobile network operators (MNO) has been that the lack of infrastructure expansion has been due to weak regulatory guidelines on radio spectrum allocation. Furthermore, as indicated in Figure 1.1 shows the data prices in the developing world, indicating that only two countries, Mauritius, and Gabon, have affordable data prices for large sections of their population as claimed in [3].

More so, the demand for more radio spectrum is on the rise for operators for legacy and new deployments. This has been further compounded by the slow progress of television spectrum digital migration (TV white spaces [TVWS]) necessary to free up some needed radio spectrum. In [4], it was reported that about 97% of TVWS spectrum rural areas are vacant. The number decreases in urban cities to between 70% and 50% in places like Pretoria and Cape Town, respectively. Stronger regulatory standards on radio frequency allocation, according to mobile network operators (MNO), will greatly help the telecommunications sector implement future network architecture.

1.2 Background to the research problem statement

The Sub-Saharan Africa (SSA) region had the highest mobile subscription growth rate in the world[4]. The projections are that close to 75 million IoT entities will be served by mobile internet in the SSA region by 2023[4]. This indicates that the demand for mobile internet resources will continue to rise. Therefore, the current traditional radio access network (RAN) designs for human-to-human (H2H) communication are not adequate, particularly, with additional use cases.

Furthermore, on a global scale, there are currently more than 5 billion mobile subscribers [5]. This translates to more than 70% of the global population with access to mobile technology. Thus, far-reaching penetration demands new radio access network designs, which not only give access but provide an improved service to a much wider audience. Therefore, a modified RAN architecture is required to fulfil the growing demand for radio access connectivity. The

new architecture is developed according to the fifth generation (5G) standards set by the International Telecommunication Union Standardization Sector (ITU-T), which forms part of the UN [6]. This architecture helps a great deal with the interoperability of information and communication technologies (ICT), which leads to a reduction in operating costs for voice, data, and video services.

The expected implementation of the 5G mobile technology would not only lead to the harmonization of the frequency spectrum across the telecommunications industry but would also lead to a shift in the design of the radio access network (RAN). Since, unlike previous mobile generations, 5G focuses on enhanced mobile broadband (eMBB) and two additional scenarios: massive machine-type connectivity (mMTC) and ultra-reliable low-latency communication (uRLLC).

Moreover, with the rise in the number of connected devices as well as mission-critical services, a computing model is seen as a possible solution since the model enables two important technologies: Fog and Cloud computing. These technologies support low latency, provide reduced Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) networks, and application in real-time. The latter leads to the development of Fog Radio Access Networks (FRAN) which, the architecture handles the limitation of Cloud Radio Access (CRAN) and Heterogeneous Cloud Radio Access Network (HCRAN) with distinctive features of high mobility, enhanced radio resource allocation, enhanced service quality, and low latency.

However, there are still challenges with implementing the FRAN architecture in 5G networks,

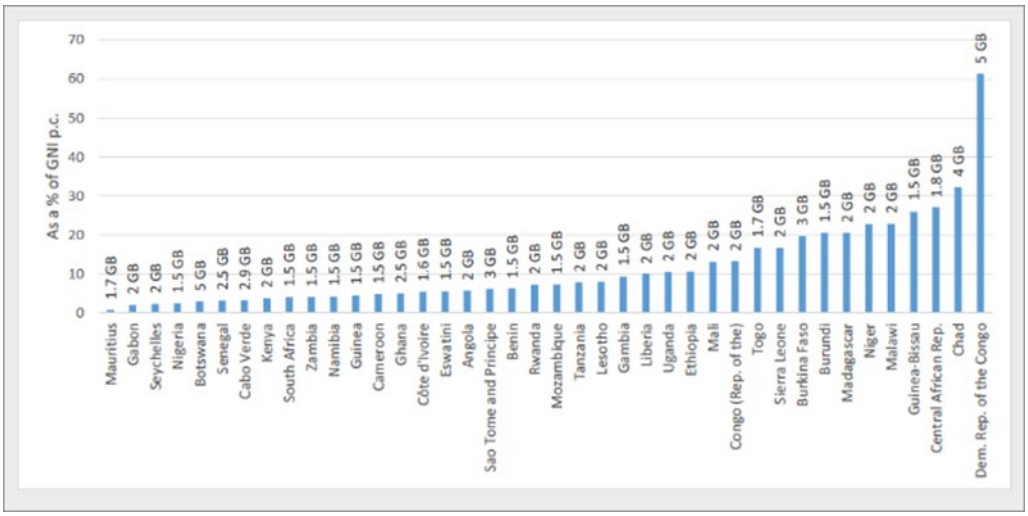

Figure 1.1 Mobile-data prices as a percentage of GNI monthly data [263]

such as optimal resource allocation, ultra-low-latency, and spectrum and energy efficiency, among others. Hence, a need to develop an enhanced resource management scheme based

on machine learning in Fog-RANs for 5G networks (5G-FRANs) to minimize the IoT device issues while also satisfying industrial requirements.

1.3 Statement of the Research Problem

The questions which this research will address are:

1. How does the increase in IoT devices impact resource management for 5G-FRAN? Research Question (**RQ1**)
2. What current machine learning algorithms are in use for resource management for 5G-FRAN? **RQ2**
3. In what way can resource management for Fog RANs in 5G (5G-FRANs) be enhanced using machine learning algorithms? **RQ3**
4. How is the use of machine learning algorithms helping in the implementation of an enhanced resource management scheme? **RQ4**

1.4 Research Hypotheses

The following three major hypotheses have been made in this research to begin studies and address the research question.

1. Implementing FRAN in a 5G system offers an enhanced network, such as low latency and spectral efficiency.
2. The FRAN offers scalable services as per user requirement, optimizing spectral efficiency for scarce resource.
3. Resource allocation and QoE of FRAN in 5G can be optimized using machine learning.

1.5 Delineation of the research

The analysis would be limited to 5G FRAN, with the key focus on resource allocation for IoT devices and their relationship with CRAN overall architecture. The algorithms that will be proposed will focus on radio resource allocation, system-level efficiency, and end-to-end latency.

1.6 Significance of the research

The 5G NR technology with Fog Computing will play a key role in the introduction of the 4IR required for networking and internet access networks. Besides, mobile internet will make it easier for the developing nations to invest and do business in the global economy. With each generation of mobile technologies, there has been an increase in the accessibility of services to the customer, be it voice quality, data throughput, and so on.

1.7 Expected original contribution

1. Develop a Fog computing resource allocation algorithm to maximize the usage of resources and optimize latency constraints.
2. Establish an equal usage policy for IoT resources that the algorithm aims to enhance latency limitations.

The above contributions will result in the publication of at least 2 journal articles and 1 conference paper in Department of Higher Education and Training (DHET) accredited journals related to FRAN QoS management.

1.8 Aim and research objectives

This research aims to develop an enhanced resource management technique for 5G-FRAN using a machine learning algorithm. The following objectives are considered to achieve the research aim:

1. To analyse and show that existing 5G-FRAN network performance for (IoT) and (IIoT) service delivery is sub-optimal. Research Objectives (**RO1)**
2. To develop an enhanced resource management scheme for 5G-FRAN based on machine learning. **RO2**
3. To model and simulate the proposed machine learning algorithm and analyse performances using Matrix Laboratory (MATLAB) software.**RO3**
4. To perform a comparative study of the results in step 2 with existing methods in the literature. **RO4**

1.9 Research design and methodology

The proposed method for the research work is machine learning (ML) using specific algorithms such as Hidden Markov Model (HMM), artificial Neural Network (ANN), Random Forest (RF), and Decision Tree (DT), which will be used to simulate the resource allocation scheme for FRAN. In this research, an experimental environment for 5G FRAN data collection will be designed using MATLAB R2020a package. The MATLAB licence is provided by CPUT, which contains the necessary 5G toolbox that can be adopted for different machine learning algorithms.

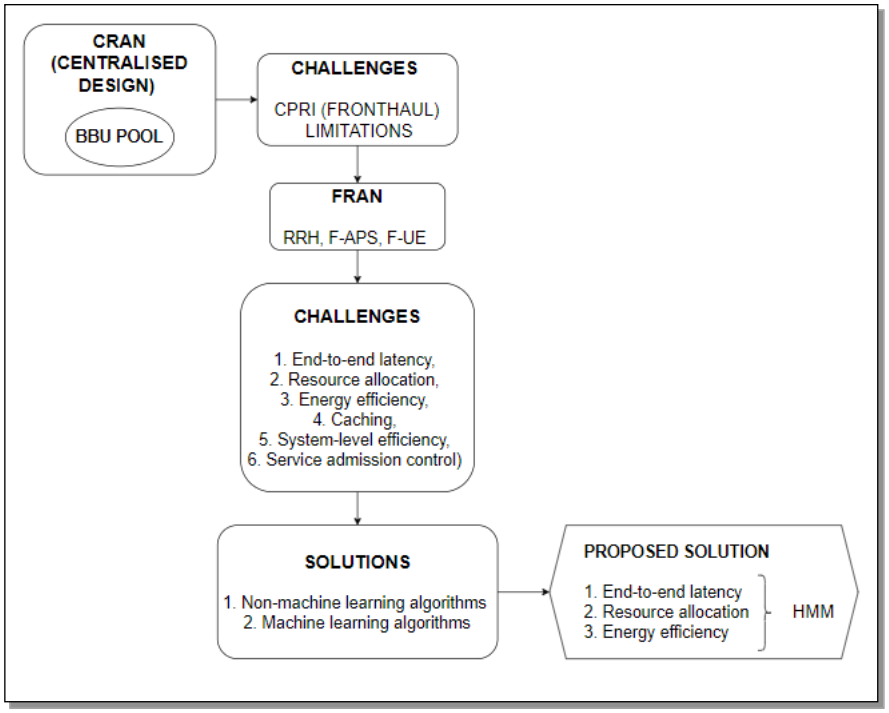

Figure 1.2: Framework for the proposed 5G-FRAN resource management scheme

Subsequently, the collected data will be analysed using digital signal processing (DSP) toolbox on MATLAB. Based on the theoretical DSP knowledge, feature extraction and reduction techniques will be developed to further remove the noise effects and reduce feature dimensionality of the raw data. The data pre-processing step is essential since to increase the accuracy of the proposed HMM while reducing the computational cost of the algorithm.

Figure 1.2 give a summary for both CRAN and FRAN architectures. Also, the challenges, existing solutions in the literature, and the proposed HMM machine learning algorithm. The suggested scheme will be implemented as a solution to the FRAN challenges: end-to-end latency, resource allocation, and energy efficiency. More so, the results of the designed HMM models will be compared to the existing machine learning algorithms, that is, K-means, Random Forest, SVM, and KNN.

Figure 1.2 outlines the major constraints linked to CRAN and FRAN architectures within the 5G system, emphasising significant concerns such fronthaul limitation, end-to-end latency, resource allocation, caching, system level efficiency, service level admission control and energy efficiency. This study reviews conventional techniques in the literature, encompassing both traditional algorithms and machine learning (ML) methodologies to tackle these difficulties. The suggested ML methods which included HMM, modified Decision Tree, Random Forest, ANN emerge as a viable alternative to enhance FRAN designs. The proposed strategies seek to alleviate the recognised issues while enhancing system performance. The outcomes of the developed ML models will be methodically compared with established

methods, such as K-means, Random Forest, SVM, and KNN, to assess their efficacy in 5G-FRAN.

1.10 Thesis Organization

This thesis has seven chapters Figure. 1.3 and below is a brief highlight for of each.

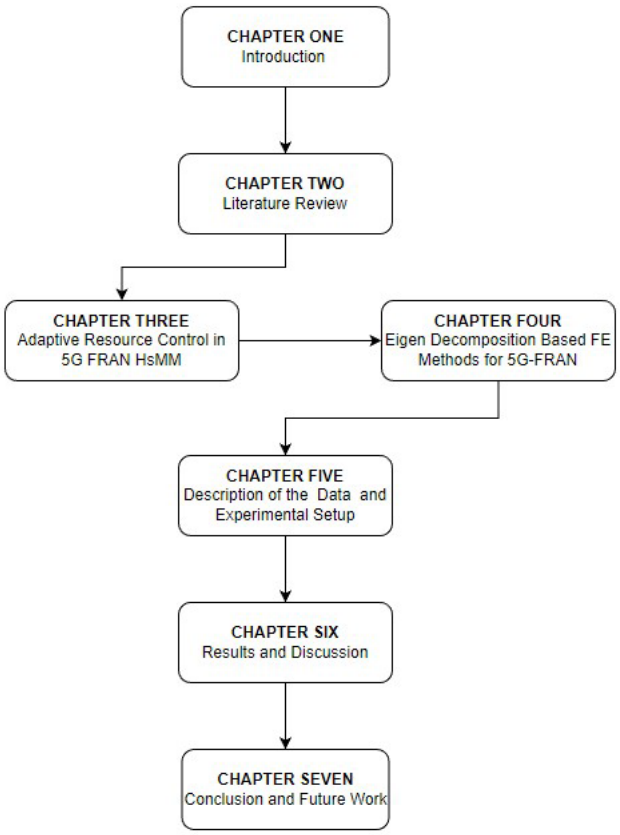

Figure 1.3: Thesis structure and content

Chapter 2: Literature Review

Meanwhile chapter 2 presents the details of 5G requirements, plus a comprehensive reconsideration of mobile network design, including the RAN architecture. The need for change in architecture is also due to several factors such as the exponential increase in the number of subscribers, the need for extensive connectivity, the handling of large amounts of data, and the demand for extremely low latency and increased data rates. Thus, a comprehensive assessment was conducted to analyse the existing RAN architectures for 5G mobile networks, with a particular focus on CRAN, HCRAN, and FRAN. The objective was to investigate the advantages of CRAN and HCRAN while acknowledging the difficulties that arise in each network architecture. Therefore, the researchers examined the existing CRAN and HCRAN

solutions. Detailed summaries were provided in a tabular format to address energy efficiency, fronthaul capacity, latency, and other relevant parameters. Regarding FRAN, a distinct approach was employed, prioritising resource categories within the design layers and thereafter doing an analysis of the resource management system.

Chapter 3: Adaptive resource control in 5G-FRANs using the HsMM-based mobility aware cell association technique

In this chapter is discussed a cell association technique for proactive bandwidth management in 5G fog radio access networks (5G-FRANs) by considering subscriber movement as a key factor. The proposed approach integrates probability distributions of time intervals with cell identifications (cell-IDs) to construct a hidden semi-Markov model (HsMM) that predicts the most suitable next cell to meet downstream rate requirements. Furthermore, the optimisation method is used to identify the cell that provides the highest rate. The suggested Hidden semi-Markov Model (HsMM) takes into consideration the ping-pong effects, missing values, and constrained context from wireless networks to understand user traffic situations. The results indicate that the HsMM-based method achieves a high level of accuracy when considering the number of observation time intervals. Furthermore, in comparison to the existing system, the suggested approach leads to a heightened level of user satisfaction.

Chapter 4: Eigen Decomposition Based Feature Extraction Methods for 5G-FRAN

This chapter introduces PCA and LDA as feature extraction algorithms. We specifically engineer the principal components from PCA and the discriminant components from LDA to extract attributes from datasets related to Fog Computing applications. We will feed the generated feature matrices into machine learning to detect and categorise service segments. PCA identifies a low-dimensional linear subspace that captures the majority of variance in a dataset, but it may not successfully construct a low-dimensional non-linear subspace if one exists inside the dataset. Additionally, when using LDA, non-linear attributes in the datasets may change how well the spatiotemporal trends in the basic system are shown. Nonetheless, the Fog Computing application dataset occasionally has nonlinear properties (Section x). Therefore, we propose advanced feature extraction strategies to enhance the use of PCA and LDA for feature extraction.

Chapter 5: Description of the Data and Experimental Setup

This chapter presents an in-depth discussion of the datasets used in this work and elaborates on the practical application of eigen decomposition and feature extraction (ED-FE) approaches, along with five machine learning (ML) models. We presented a block diagram that encapsulated the operational workflow of the ML models used in this investigation, followed by a comprehensive explanation of the experimental design. We used the deployed models,

which were divided into baseline ML models (MLs), PCA-enhanced machine learning models (PCA-MLs), and LDA-enhanced machine learning models (LDA-MLs), in a planned way on fog computing application datasets that had different types of classes of activity. This chapter outlines the settings and modifications made to each model, examining the impact of various feature extraction methods on the models' performance in noise-injected and multi-class classification scenarios. Chapter 6 fully analyses and discusses the results of these experiments. The experiments investigate how reducing the number of dimensions affects accuracy, efficiency, and noise resilience in different machine learning settings.

Chapter 6: Results and Discussion

For chapter 6, results and discussion chapter concludes with an observation that LDA generally performs better than PCA across the models in terms of stability and accuracy, particularly under conditions of higher noise. However, the LDA's ability to manage FDR and provide more consistent accuracy and sensitivity makes it a superior feature extraction technique for the FCA dataset in this context.

LDA's as FE technique strength lies in reducing false positives (better FDR) and maintaining higher accuracy and sensitivity, especially for models like Random Forest, k-NN, and Neural Networks. On the other hand, PCA tends to be more sensitive to noise and shows higher variability in its AUC performance. Thus, for FCAs, LDA is the better choice for ED-based FE compared to PCA.

Chapter 7: Conclusion and Future Work

The performance comparison reveals that different machine learning models exhibit varying levels of effectiveness when combined with PCA and LDA for feature extraction. Random Forest and Neural Networks generally perform better with PCA, showing higher accuracy and AUC mean values. However, noise levels significantly impact the performance of all models, with higher noise leading to increased FDR and decreased accuracy. These findings highlight the importance of selecting appropriate feature extraction techniques and considering noise levels in the development of robust machine learning models. Future research should focus on optimizing these techniques to enhance model performance under varying conditions.

Chapter 8:

The thesis delivers and conclusion remarks.

1.11 Delineation of the research

The analysis would be limited to 5G FRAN, with the key focus on resource allocation for IoT devices and their relationship with CRAN overall architecture. The algorithms that will be proposed will focus on radio resource allocation, system-level efficiency, and end-to-end latency.

1.12 The significance of the research

The 5G NR technology with Fog Computing will play a key role in the introduction of the 4IR required for networking and internet access networks. Besides, mobile internet will make it easier for the developed world to invest and do business in the global economy. With each generation of mobile technologies, there has been an increase in the accessibility of services to the customer, be it voice quality, data throughput, and so on.

1. Research output, results, and contributions
2. Develop a Fog computing resource allocation algorithm to maximize the usage of resources and optimize latency constraints.
3. Establish an equal usage policy for IoT resources that the algorithm aims to enhance latency limitations.

The above contributions will result in the publication of at least 2 journal articles and 1 conference paper in Department of Higher Education and Training (DHET) accredited journals related to FRAN QoS management.

1.13 Summary

In current research, it is forecasted the increase in the number of IoT devices is forecast to expand beyond 900 million by 2023. This exposes the limitation of the centralized network architecture of CRAN as it suffers from congested fronthaul. This research focuses on 5G-FRAN, whose architecture handles the congested fronthaul limitation of Cloud Radio Access (CRAN) and Heterogeneous Cloud Radio Access Network (HCRAN). Moreover, challenges such as resource allocation, end-to-end latency, and energy efficiency of implementing the FRAN architecture in 5G networks are investigated in the research. The aim is to develop an enhanced resource management technique for 5G-FRAN using a machine learning algorithm. Thus, a hidden Markov model (HMM), which is a robust machine learning technique is proposed as an enhanced resource management algorithm for the 5G-FRAN system. The proposed technique will optimize resource allocation and quality of experience (QoE) of FRAN

in 5G. Additionally, the HMM algorithm will provide reduced computational complexity compared to state-of-the-art ML algorithms.

-

CHAPTER:2 LITERATURE REVIEW

2.1 Introduction

The Covid-19 pandemic has profoundly affected the digital landscape globally. The social distancing strategies taken to halt the transmission of the pandemic have highlighted the importance of connectivity for social and economic well-being [7], [8]. Thus, emphasizing the need for a strong and inclusive digital economy, which is supported by universal access to fast, reliable and efficient internet. Fifth generation (5G) networks [9] are being deployed around the world to satisfy the need for several usage, which focuses on enhanced mobile broadband (eMBB) and two additional scenarios: ultra-reliable low-latency communication (uRLLC) and massive machine-type communication (mMTC). Consequently, the new enhanced 5G radio access network (RAN) architecture, is supposed to be spectrum efficient and utilises enhanced algorithms. Hence, the 5G RAN standards incorporate cloud-native architectural designs, which are cost efficient, flexible, and scalable [10] . Further allowing 5G to be used for a variety of applications, such as healthcare, automotive, entertainment, internet of things (IoT), and industrial internet of things (IIoT) applications.

The amount, variety, and speed of data generated by IoT, and future technologies are expected to propel the global market for internet of things (IoT) end-user devices at an astounding rate. Industry sales for IoT surged from 100 to 212 billion dollars between 2017 and 2019, and growth is anticipated to reach 1.6 trillion dollars by 2025 as indicated in [8]. Thus, around 50 billion of these advanced devices would be in use worldwide by 2030, forecasters International Data Corporation (IDC) [11], have predicted. This would create a vast network of linked devices that would include everything from cars to smartphones, home appliances, and more. Hence, the demand for mobile internet resources is expected to increase to support billions of users, causing challenges for resource management in 5G networks. System resource limitations and other challenges, among others, have an impact on the IoT's dynamic character [12],[13].

Numerous concepts and technological innovations have been proposed in the literature, particularly for 5G networks' Radio Access Network (RAN). Such concepts are aimed at satisfying growing capacity requirements, reduce CAPEX and OPEX networks, provide ultra-low latency and higher bandwidth to multiple end-users, and integrate existing technologies. The cloud radio access network (CRAN) is a network architecture that uses cloud computing to achieve 5G goals [10]. Furthermore, security, fronthaul, and a centralised baseband unit (BBU) pool constraint reduce CRAN capabilities. Fronthaul links with high bandwidth and ultra-

low latency are required in CRAN for centralised BBU processing. However, in real-time applications, the capacity of a link is usually limited and time delayed.

To address the CRAN fronthaul issues, [14] introduced a heterogeneous CRAN (HCRAN) that separates the user plane from the control plane. High power nodes (HPNs) provide uninterrupted coverage and other control plane duties and are linked to the BBU pool via backhaul to manage interference. Remote radio heads (RRHs), on the other hand, provide user plane high-speed data transmission. Nonetheless, real-time HCRAN implementation remains a challenge. The number of fronthaul links connecting RRHs and the centralised BBU pool grows, with several redundant traffic data due to the increased use of location-aware social applications, and thus the HCRAN fronthaul burden grows. Edge devices capable of data storage and processing, such as RRHs and intelligent user equipment (UEs), can be used to reduce the BBU pool and fronthaul link limitations. However, the edge devices are underutilised during low-traffic periods, resulting in an increase in CAPEX/OPEX costs.

Furthermore, with the proliferation of connected devices and the addition of mission-critical services, the fog computing model is regarded as one of the best solutions. The model supports two critical technologies: fog and cloud computing. These technologies enable low latency and lower network upgrade and maintenance costs. Thus, fog radio access networks (FRAN) are one of the most recent technologies to emerge in a short period of time. As mentioned in the article, the technology allows for low-latency service in 5G by bringing computing resources from the BBU pool to the network's edge [15]. The FRAN deals with the limitations of the Cloud radio access network (CRAN), service quality, and low latency [14]. Notwithstanding, there are still challenges to be addressed when implementing the FRAN architecture in 5G networks, such as optimal resource allocation, ultra-low latency, and spectrum and energy efficiency, among others. As a result, an improved resource management scheme based on machine learning for 5G Fog-RANs is required (5G FRAN). This is done to reduce the challenges of IoT devices while meeting industrial requirements.

The major contribution of this chapter was to undertake a comprehensive evaluation of resource management for 5G-FRAN. Thus, various resources within the four layers the FRAN system were categorised. The main objective is to identify the optimal resource management techniques for the 5G-FRAN system.

A full assessment of the 5G communication network, covering service classifications, preferred network topology, benefits, applications, and implementation concerns. In addition, the performance requirements, worldwide deployment of 5G, and enablement of numerous vertical sectors using network slicing and network virtualization were also considered.

-

The development of RAN architectures in the past and present is examined. Consequently, several RAN implementations which have been published in the literature are fully discussed. We also go over the present state of the art, ongoing standardisation initiatives, and why existing RAN systems need be rebuilt and remodelled to suit the demands of next-generation mobile networks.

A detailed comparative analysis of CRAN, HCRAN, and FRAN designs in this work is provided, with the goal of evaluating these RAN architectures from several perspective such as enhanced energy efficiency, enhanced energy efficiency, reduced CAPEX/OPEX, resource allocation and resource sharing, and so on.

In addition, we provide an in-depth analysis and overview of the essential enabling technologies for 5G RAN. We also look at 5G RAN improvements, which are projected to improve spectrum efficiency, lower CAPEX/OPEX, and meet end-user expectations as well as vertical industrial requirements.

This chapter is organized as follows. Section II introduces the basics of the 5G system, key milestones, and Standard Development Organisations (SDOs). Section III discusses the 5G radio access network (5G RAN) architectures and limitations. Section IV investigate the resource management categories for Fog RAN. Finally, section V concludes with the key contributions and future work.

2.2 5G Mobile Communication

The advent of 5G mobile communications has had a substantial effect on society and industry. When compared to previous cellular generations, 5G allows for massively better peak data rates that are accessible nearly everywhere and at any time. Even though mobile broadband services are readily accessible today, 5G essentially allows the next generation of human connectivity and human to everything interaction, such as widespread use of virtual or augmented reality, free viewpoint video, and telepresence [16]–[18]

The Mobile and Wireless Communications Enablers for Twenty-twenty (2020) Information Society (METIS) project proposed three major service types to improve 5G capabilities: eMBB, mMTC, and uRLLC [19]. The use of eMBB for 5G upgrade’s the 4G LTE mobile broadband service, with faster connections, higher throughput, and more capacity. Consequently, cities, stadiums, and other high-traffic areas will benefit. While the uRLLC, on the other hand, is using the network for mission critical applications, that is, services that require continuous and robust data transfer. In this case, short packet data transmission is used to meet the requirements of reliable and low-latency wireless communication networks. Furthermore, mMTC is expected to be used for connecting many devices, particularly the numerous connected IoT devices.

-

These concepts were also approved by the International Telecommunication Union Radio-communication Sector (ITU-R) Working Party 5D (WP5D), as such eMBB was the first standard to be implemented in 2020 for 5G. The ITU-R recently issued a new recommendation titled "International Mobile Telecommunications (IMT) Vision Framework and Overall Objectives of Future Development for 2020 and Beyond" [20], [21]. Following that, standardisation organisations like 3rd Generation Partnership Project (3GPP), the Institute of Electrical and Electronics Engineers (IEEE), and others transformed the ITU's goals to 5G standards. Furthermore, other Standards Developing Organizations (SDOs) and industrial forums, such as the Broad-band Forum (BBF), Open Radio Access Network (O-RAN), Small Cell Forum, and others, continue to develop competing or complementary technologies to the 3GPP standards [22]–[25].

2.3 5G Radio Access Network Architecture

The most important part of a mobile network is the radio access network (RAN). This becomes clear during the deployment stage, which is then followed by the operational complexity and cost. The RAN, which uses radio access technologies (RATs) to offer coverage in a particular area, is made up of a group of interlinked base stations that are connected to the core network (CN). The conventional mobile networks rely on a dispersed RAN architecture where a single RRH is connected to a single, baseband unit (BBU) through a common public radio interface (CPRI) or coax cable link. For instance, the RRH positioned on the tower's pole is connected to the BBU situated at the base of the tower. It is not economical nor energy-efficient to use this one-to-one RRH-BBU architecture. Given the rapid evolution in the number of connected devices and the rising data speeds for the 5G standards, it is necessary to adapt classical RANs to meet a variety of service requirements. Despite advancements made to the traditional distributed RANs, the design was unable to meet the demands of 5G; as a result, the novel idea of a cloud RAN (CRAN) has been suggested in several studies [26]–[28].

2.3.1 Cloud radio access network (CRAN)

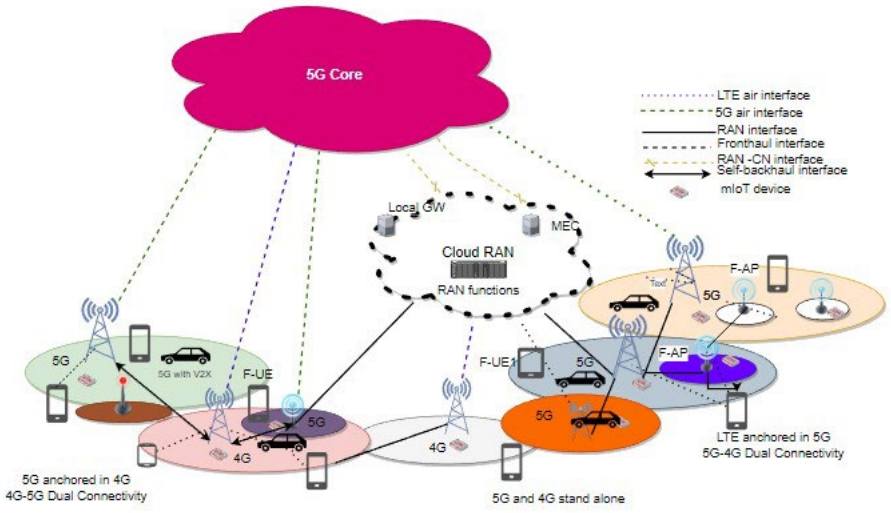

Figure 2.1: 5G Cloud Radio Access Network Architecture

The 5G mobile design called CRAN was developed because of current traffic characteristics and technical advancements. The foundation of CRAN is centralised computation, shared radio, and real-time cloud architecture, which entails dividing the base station (BS) RRHs and BBU pool to establish a centralised pool as depicted in Figure 2.1. By virtualization and cloud computing technologies, the centralised BBU has improved processing. Besides reducing the number of BS, the CRAN design intends to save the network energy. To further enable dynamic resource allocation, improved spectrum efficiency, high bandwidth utilisation, flexibility in design, and operational efficiency, the CRAN makes use of collaboration and virtualization technology.

The fundamental idea behind CRAN's division of the BS into BBU and RRH. The RRHs oversee light computational functions including signal modulation and amplification, whereas the BBUs oversee intensive baseband signal processing activities [29]. Moreover, this breaks the direct connection between BBU and RRH, resulting in a dynamic provisioning of each RRH to the BBU pool. In [30], the real-time virtual allocation BBU pool is used to increase the processing capacity of the virtual BS, which is used by RRH to send and receive signals.

2.3.1.1 Benefits of CRAN

The following are some significant advantages of CRAN over traditional RAN.

-

2.3.1.1.1 Enhanced Energy and Spectral Efficiency

CRAN, in comparison with conventional RAN, employs fewer BBUs, leading to a substantial decrease in energy usage [31], [32]. Furthermore, because RRHs can be normally cooled by hanging on towers or building walls, radio modules do not require a lot of sites supporting equipment such as air conditioning. Furthermore, CRAN enables the offloading of energy-intensive data computations from UEs and BSs to a nearby cloud, saving energy. Furthermore, the introduction of collaboration radio technology aids in the reduction of distance between RRHs and UE, reducing interference among RRHs while reducing energy usage in the 5G network. The benefits and drawbacks of employing licenced and unlicensed spectrum of various technologies for the fronthaul of the 5G-CRAN were examined by [33] as part of the related studies on improving spectrum efficiency by installing the CRAN. Furthermore, enhanced inter-cell interference coordinated (eICIC) and coordinated multiple point transmission (CoMP) over the RRHs connected to the same cloud were examined in [34], [35] as a cooperative and coordinated transmission strategy for improved spectral efficiency (SE).

2.3.1.1.2 Reduced CAPEX/OPEX

By virtualizing baseband processes in the cloud infrastructure, mobile network operators (MNOs) can decrease CAPEX and OPEX of 5G-CRAN. Low-cost RRH spatial-time deployment, installation, and operation are required for the CRAN architecture. Thus, in [36] investigated the cost of deploying BSs, transport networks, and data centres based on several processes in the spatial domain. Based on the costs of BSs and the combination of backhaul methods for connecting BSs with data centres, the analysis revealed that CRANs can reduce CAPEX by up to 15%. Similarly, in CRAN, a two-phase framework for determining the best BS clustering scheme was proposed in [37]. The cost of deployment was reduced by 12.88% thanks to the framework. In [38], the cost for three BBU hotelling strategies were modelled and compared: BBU stacking, BBU pooling, and CRAN BBU. A network planning and provisioning framework was proposed in [39] to optimise the cost of deployment in CRAN. The framework made use of a mixed integer quadratic constrained programming (MIQCP) method to reduce the cost of deploying a virtualized 5G service chain.

2.3.1.1.3 Enhanced Mobility Management

The capacity to automatically allocate BBU resources to RRHs for congestion control mobile traffic between the BBU pools is a significant advantage of BBU pooling. CRAN is suitable for dynamic distributed traffic because to the congestion control functionality in the distributed BBU pool [40]. The serving BBU stays in the same pool even when the serving RRH changes continuously due to UE movement because its coverage is wider than that of a conventional

BS. As a result, non-uniformly distributed traffic generated by UEs can be routed through a virtual BS inside the same BBU pool, helping the MNOs manage mobility. Location-based algorithms for mast segmentation and BBU cluster categorization using forecasts of movement and traffic trends in the CRAN were reported in [41]. The CRAN provides the 5G mobile communication system with several additional advantages, which are covered in depth in [42].

2.3.1.2 Challenges of CRAN

In addition to the advantages CRAN brings to 5G systems, there are still certain restrictions, which include:

2.3.1.2.1 Security

Because security breaches seem to be on the increase, the CRAN potential threat has prompted concern. In besides conventional wireless network safety threats such as primary user emulation attack (PUEA), CRAN faces extra safety threats such as eavesdropping and jamming, MAC spoofing, IP spoofing and hijacking, transmission control protocol (TCP) flooding attacks, and file transfer protocol (FTP) attacks [43]. Studies like (Wu et al., 2015) have searched in to the CRAN cyber-attacks and mitigations as [10] [44]–[46].

2.3.1.2.2 Fronthaul Link Capacity

The network may become functionally useless if the cloud fails because the BBUs of several BSs have been deployed there. Backhaul links enable a connection between BBUs and the core mobile network, whilst fronthaul links allow a seamless connection between RRHs and the BBU pool (centralised pool). Notwithstanding CRAN's advantages, performance is still constrained by fronthaul and backhaul link capacity constraints. In order to address fronthaul concerns with capacity for downlink and uplink CRAN, a number of studies on fronthaul compression have been conducted [47]–[61]. The methods strive for efficient fronthaul link transmission. Data sharing and compression techniques are two main methods of transmission. These methods vary depending on whether the information intended for mobile users is jointly encoded and pre-coded at specific BSs (data-sharing approach) or at the centralised BBU (compression technique). Before being transferred to the RRH through fronthaul links, generated signals from the latter are quantized and compressed [48]. For downlink CRAN, a generalised compression method was presented in [47]. For integrating the BSs to a centralised processor, a fronthaul with a limited capacity was developed.

This technique enables the centralised processing to collaboratively centrally encode signals that will be broadcast by the BSs using the fronthaul links. With the introduction of a combined data-sharing and compression solution for downlink CRAN in [48], fronthaul/backhaul capacity

use can now be effectively managed. In addition, the problem of combining the optimisation of digital baseband beamforming and fronthaul compression methods at the BBU with RF beamforming at the RRHs was investigated in [55]. Its front hauling and hybrid beamforming joint design maximises the weighted downlink sum-rate and network energy efficiency of the CRAN system. In [50] evaluated a normal point-to-point fronthaul compression technique, while [51], [52]. To accommodate multi-antenna users, a joint fronthaul compression and transmit beamforming architecture was taken into consideration for uplink CRAN [56] and [57]. Meanwhile, multi-antenna BSs were used by the users to communicate with the centralised processor. The digital fronthaul cables used to connect the BSs to the central processor had a small capacity. Thus, single-user compression (also known as point-to-point compression) and Wyner-Ziv coding for CRAN were two alternative fronthaul compression strategies that the authors of [56] investigated. They also suggested an approach to noise covariance matrix quantization optimisation. That each BS uses single-user vector quantization to compress the received signals from various BSs. With the Wyner-Ziv coding, on the other hand, makes full use of the correlation between the signals that were received to increase compression efficiency and hence improve performance.

In [60][58][53], researchers examined the joint design of CRAN with wireless fronthaul connection and OFDMA access link. Furthermore, an intelligent reflecting surface was used to improve the wireless fronthaul connectivity between the multi-antenna BBU pool and the multi-antenna RRHs in [62]. Meanwhile, the Wyner-Ziv coding was used to compress each incoming signal at a specific RRH. The fronthaul transmission rate must be kept to a minimum to maximise compression efficiency, according to a suggested adaptive compression strategy in [58]. In addition, fronthaul compression was carried out at the RRH to send baseband signal through the fronthaul connection to the centralised unit in [59]. Consequently, the algorithm optimised the power spectral density of the quantization noise at the RRHs using the Charnes-Cooper transformation and difference of convex technique.

2.3.2 Heterogeneous Cloud Radio Access Network (HCRAN)

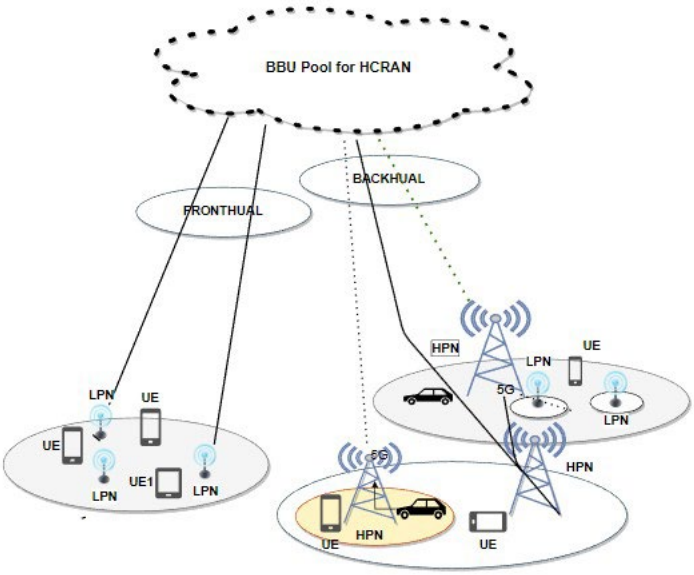

Figure 2.2: 5G-HCRAN Architecture

Heterogeneous Cloud RAN(HCRAN) is a concept developed in response to the demand for dense 5G RANs. The HCRAN is a low-cost solution that makes use of cloud computing and heterogeneous networks (HetNets), as seen in Figure 2.2. The HCRAN design described in [14] and [63] included a variety of cells, including macro-BSs, or HPNs, micro BSs, and RRHs. The macro-BSs offer system performance improvement, network control, and mobility management. In contrast, micro-BSs and RRHs increase system capacity while reducing transmission power. While the BBU pool comprises additional baseband physical processing functions connected to the higher layers, the RRHs enable radio frequency and signal processing. The HPNs also provide access to all physical and network layer functionalities. The UEs receive system broadcasting data and control signalling from the HPNs, which improves the fronthaul link capacity and decreases HCRAN time latency. Below are several advantages of HCRAN in 5G systems that are discussed.

2.3.2.1 Benefits of HCRAN

Numerous research has been done on HCRAN from various angles, including energy efficiency, spectrum efficiency, resource allocation, cost and load balancing, mobility management, performance enhancement, and so forth. This is so that HCRAN may properly exploit the 5G system's HetNets and cloud computing capabilities.

-

2.3.2.1.1 Enhanced Energy and Spectral Efficiency

The deployment of novel techniques for effective energy and spectrum utilisation is made possible by the combination of HetNets with CRANs. An illustration of coordinated transmission and reception is the cloud computing-based coordinated multipoint (CC CoMP) system discussed in [64]. The femto, pico, and macro cells of this CC-CoMP enabled HCRAN are effectively RRHs connected to a centralised baseband processing centre where signals are merged, comparable to a big, dispersed MIMO system. By reducing the strict backhaul and synchronisation requirements among distant cells, the collaborative processing in HCRANs becomes feasible and commercially sustainable. To improve spectral efficiency for cell edge users in NOMA-based networks, [65] used joint transmission coordinated multipoint (CoMP) transmission to mitigate inter-cell interference. Furthermore, [14] carefully designed the HCRAN coverage area to improve spectral efficiency by significantly shortening the communicating distance between serving RRH and desired mobile terminals.

2.3.2.1.2 Resource Sharing and Allocation

Spectrum sharing [66], [67], infrastructure sharing [68], [69], and network sharing [70], [71] are the three categories under which resource sharing for HCRANs has been categorised. It is possible to further segment the spectrum and infrastructure resources into sharing entities, network slices, and logical connections [72]. A base station, a network of connected base stations, or a base station component are examples of sharing entities that represent a collection of available resources. Network slices are used to divide up the available resources across two or more sharing organisations. The spectrum and infrastructure of the HCRAN are divided into network slices at the network level according to factors like throughput and processing. A dynamic pool of spectrum resources, improved infrastructure coverage, and virtual networks tailored to a particular service class are just a few of the advantages of resource sharing that are made possible by HCRAN's implementation of these processes.

2.3.2.1.3 Enhanced Mobility Management

The independent plane design of the HCRAN in 5G systems provides better movement. The result was the presentation of a cell outage management (COM) solution for HetNets with separate management and data BS layers [73]. The data BSs were used to manage user data, while the management BSs were used for control information transmission and subscriber movement. Moreover, in conventional HetNet, a cooperative cell outage detection (COD) method for pico cells was proposed in [74]. Moreover, [75] utilised the COD and cell outage compensation (COC) algorithms for HetNet's identification and mitigation of microcell and macro cell failures, respectively. In their paper [76], the authors of the very dense HCRAN

provided a method for user equipment to directly choose the cell with the highest priority in terms of cell features, mobile equipment, mobility features, and application features.

2.3.2.1.4 Enhanced Performance

The core cloud computing management mechanism, where different node entities interface, has improved system performance in HCRAN. To improve CRAN's usefulness and efficiency, the control and user planes were separated in the HCRAN architecture in [14] and [63]. As a result, the HCRAN design may greatly enhance the spectrum and energy efficiency of CRAN. Two distinct HCRAN designs with various cell sizes were presented in [64]. Wireless or optical backhaul lines were employed for a single cloud to link it to BSs from various tiers. On the other side, a collection of BSs were connected to several clouds through hybrid backhauling. In addition, a design that divides the HCRAN function into three distinct layers the infrastructure, control, and application layers was presented in [77]. To achieve broadband transmission, the design uses massive MIMO (mMIMO). Because RRHs are reduced and sophisticated computations may be carried out and managed by the BBU pool created by mMIMO, this strategy increases the system's flexibility and scalability.

2.3.2.2 Challenges of HCRAN

Heterogeneous cloud networks (HCRAN) in 5G systems have the benefits, but HCRAN has several drawbacks that are listed below.

2.3.2.2.1 Energy Consumption

To increase capacity despite increasing system energy consumption, RRHs and micro-BSs had been placed extremely densely in HCRAN. Also, a non-orthogonal multiple access (NOMA)-based radio resource management strategy was presented to maximise the energy efficiency of HCRAN in [78]. The network energy efficiency optimization problem was created by the authors as a mixed integer non-linear programming problem (MLNLP). The strategy reduces the overall energy consumption of the system by allowing users to access network services utilising non-orthogonal resources. The HCRAN network's energy usage was also improved through the introduction of a pre-coding technique [79]. The proposed architecture investigated a macro/femto HetNet in which a macro base station (MBS) was represented by a high remote radio head (HRRH), and a femto BS by a low remote radio head (LRRH). The method showed a higher throughput while using less energy as compared to the traditional HetNet without CRAN.

2.3.2.2.2 Backhaul Load Balancing

Ever rapidly expanding volume of information flow on the backhaul links between the BBU pool and RRHs is a major hindrance for the adoption of HCRAN's centralization as cellular communication networks transition to dense and dynamic HetNets [80]. Because to insufficient backhaul capacity, the RRHs cannot maximise the HCRAN system's current radio resources. Thus, it is necessary to effectively balance and reduce the transmission load on the backhaul links. To address the HCRAN transmission backhaul challenges, information compression techniques in the time and frequency domains have been extensively researched. These techniques boost the effectiveness of quantization but also make the system more complex. Backhaul workload balancing was presented as a method in [81], to reduce the need for backhaul data transfer by HCRANs. This method calls for compiling the total number of operational RRHs in each area and planning the service area for each RRH as an optimisation problem. Allocating data transmission among several users to either a macro-BS or an RRH, however, continues to be an issue. In the meantime, it has been investigated whether HCRAN integration with coordinated multipoint (CoMP) transmission in ultra-dense heterogeneous cellular networks is a workable solution for issues like insufficient fronthaul capacity and excessive noise. Be aware that some writing used lossless, infinitely capacity fronthaul link assumptions, which are unrealistic for fronthaul links with real-world capacity constraints. To facilitate transmission across capacity-constrained fronthaul networks, distributed clustering algorithms such distributed Wyner-Ziv compression [82] with joint decompression and decoding [83] are typically used.

2.3.2.2.3 Latency

Providing uRLLC services is one of the main goals of the 5G system, as was already mentioned. Other difficulties, such as establishing total self-organization for ultra-low-latency networks, still exist given the development of HCRANs to improve network management, resource management, and better throughput. In [84], resource scheduling techniques were used to improve end-to-end latency performance. By using this method, data transmission delays at the system's air interface are reduced. Also, a new access control strategy that decreases latency in wired and wireless network backhaul enhances the performance of the HCRAN system [85]. Nonetheless, there is still significant data transmission delay due to signalling overheads in the air interface. As a result, a delay-aware centric technique that reduces signalling overhead while improving latency performance was proposed in [86]. The method employs a pre-emptive network that connects intelligent mobile machines (IMMs) that use CoMP. Because the HCRAN design includes numerous access points (APs) for coverage and high-power nodes (HPNs) for broader coverage, the approach was successful. Additionally, [87] discussed a design that reduces three types of latency: air interface latency,

radio resource optimisation latency, and routing/paging procedures latency. To reduce air interface latency, the authors used a systematic design with novel open-loop radio access.

Similarly, an information-based resource optimisation strategy was used to reduce radio resource optimisation latency, while a social data cache-based routing/paging strategy was used to eliminate the latter type of latency. However, this design is not applicable for heterogeneous carrier communications over the HCRAN and remains an open issue because unreliable communication can negatively impact latency performance. Furthermore, [88] developed a dynamic stochastic resource optimisation model to achieve a balance between energy efficiency and queue latency limitation. The method employed a straightforward Lyapunov optimisation strategy based on local information at the transmitter. The method employed a straightforward Lyapunov optimisation technique based on local information at the transmitter. This method used a weighted minimal mean square error (WMMSE) technique to address the issues of energy efficiency optimisation, RRH power consumption, and interference limitations to achieve near-optimal system stability. Consequently, a balance between times averaged energy efficiency and queue bottlenecks is achieved.

2.3.2.2.4 Interference Mitigation

As CRAN operates centrally, it enables collaborative signal processing throughout networks and promotes coordination between BSs, making interference control in HCRANs more efficient than in a normal HetNet. Furthermore, due to the breadth of the network, fronthaul/backhaul restrictions, and the significant variability of BSs, mitigating interference in HCRANs may be difficult. Several studies have been conducted as a result to address the HCRAN interference issue. According to [89], a cloud computing unit was employed to manage the operation state of the BSs as part of an interference-conscious user association approach based on cell sleeping. To improve the aggregated user utility, a suboptimal user association problem was proposed, and a distributed heuristic algorithm was offered as a solution. Also, a shared transmission-based approach for collaborating that lessens interference from users near the cell's edge The CoMP clustering method was first introduced in [90]. Two clusters of RRHs—measurement RRHs and coordinated RRHs—were obtained by the authors using a pseudo-dynamic clustering technique. In contrast to the latter, which gathers and pre-processes user input, the former saves measured RRH data such the received signal strength indicator (RSSI) and channel status information (CSI). Comparable to this, [91] developed a system for cooperatively mitigating interference between RRHs and MBSs that was based on a user-weighted probability technique for dividing the spectrum into dedicated and shared portions. The approach assists in maximising overall throughput for users at every tier as well as allocating each BS into the proper spectrum parts based on QoS requirements.

-

In addition, a framework for coordinating interference that is based on contracts and that lessens interference between tiers was provided in [92]. The method divided the downlink transmission period into three stages: RRH with all UEs, RRH with individual UEs, and RRH-MBS with separate UEs. The BBU pool of all RRHs submits a contract request to the MBS, which serves as the decision agent. The MBS chooses whether to accept or reject the contract based on a logical constraint. The implementation of an ideal contract design for rate-based utility maximisation was made possible by the assumption that both the principal and the agent have perfect CSI. Also, the study looked at contract optimisation for partial CSI. For suppressing HCRAN inter-tier interference, the collaborative processing and cooperative radio resource allocation (CRRA) technique was introduced in [93]. The technique uses beamforming (BF) and interference collaboration (IC) techniques to reduce inter-tier interference. To increase the sum rates of users accessing RRHs depending on power allocation, CRRA optimisation models were further developed for the IC and BF schemes.

2.3.3 Fog radio access network (FRAN)

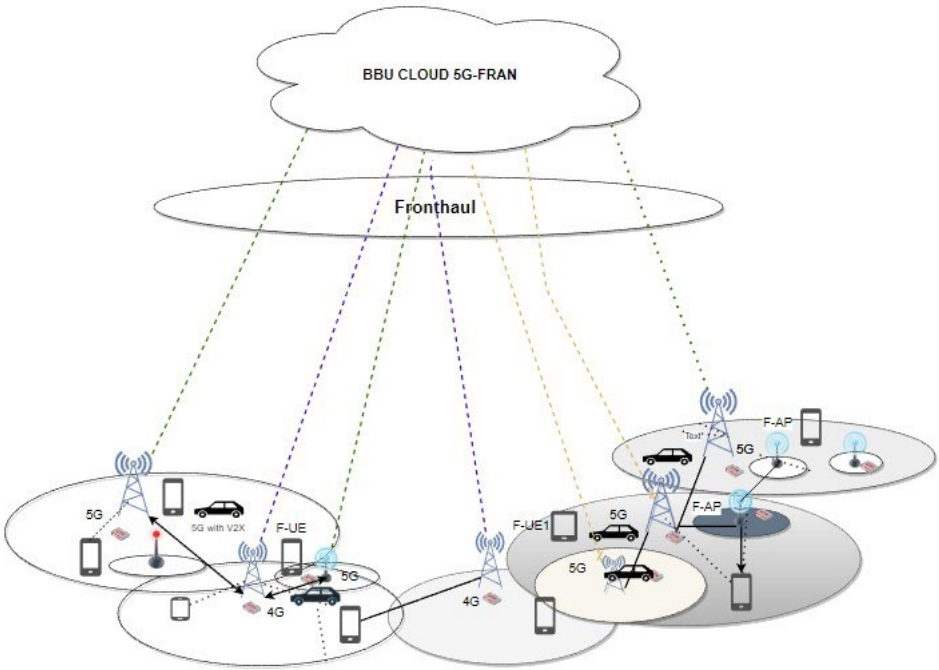

Figure 2.3: 5G FRAN Architecture

To address the HCRAN challenges and provide improved QoS to users, a new RAN design based on fog computing was lately examined in the scientific literature. Based on the advantages of CRAN and fog computing, fog computing-based RAN (FRAN) design aims to tackle the problem of rising traffic demands while also improving QoS for end-users. Given the

current world market for IoT end-user solutions, cloud computing is one of the most appropriate solutions, in which data generated by IoT devices is collected, processed, managed, and evaluated by remote servers hosted on the internet. It enables organisations to focus on their core competencies instead of planning, deploying, and retaining network infrastructure [94]. The cloud storage approach, on the other hand, is plagued by challenges such as long end-to-end latency, traffic congestion, massive data processing, and connectivity costs. As a result, [95] suggested a fog computing method to replace the remote servers' functionalities. This method is used to bring cloud computing to the edge nodes, allowing for the allocation of a huge number of processing, storage, connectivity, control, configuration, monitoring, and management functions at the mobile network's edge. Apart from CRAN, which uses a centralised content storage structure, [96] categorised FRAN into distributed and centralised networks. Some of the BBU's tasks, such as computing, caching, and resource management, are assigned to the RRHs and UEs in the distributed FRAN. The centralised FRAN, on the other hand, uses software defined networking (SDN) and network virtualization to support logical centralised control planes, improve resource management, and simplify resource sharing. Numerous studies [97]–[99] have examined the FRAN architecture and the effects of major enabling techniques on the network.

The standard FRAN system architecture is depicted in Figure 2.3. The FRAN architecture in [97] consists of three layers: terminal, access, and cloud computing layers. In particular, the fog computing layer includes fog user equipment (F-UEs) and fog access points (F-APs) from the terminal layer and network layer, respectively. Using the D2D mode or mobile relay mode, nearby F-UEs can communicate with one another at the terminal layer. In addition, F-UEs can connect to high power nodes (HPNs) to receive system signal information. In addition, HPNs and F-APs exist in the network access layer. F-APs process data received from F-UEs and transmit it to the cloud computing layer via fronthaul connections. Conversely, the HPNs connect to the BBU pool via the backhaul connections. The BBU pool that is compatible with the HCRAN BBU pool and centralised caching is in the cloud computing layer. Notate that designating a substantial amount of collaboration radio signal processing (CRSP) and cooperative radio resource management (CRRM) functions to the F-APs and F-UEs reduces the fronthaul and BBU pool load constraints.

2.3.3.1 Benefits of FRAN

The research explored the FRAN technology for 5G mobile communication systems from various perspectives, including system level efficiency, energy consumption, radio resource allocation, caching, and service admission control, among others. The following is the latest overview of the FRAN's methods in different scenarios detailing the benefits.

2.3.3.1.1 System Level Efficiency

On the usefulness of the FRAN system, numerous investigations have been conducted. Considering FRAN system with ultra-low applications, such as IoT application, a trade-off between system performance, processing, and communication costs was discussed in the FRAN system [98]. The research confirms how crucial it is to carefully select these parameters to deliver ultra-low latency services. In [100], a hierarchical content caching method for FRANs was developed to alleviate constraints on fronthaul capacity and transmit latency. To leverage the use of the central and edge caches in the strategy, RRH clusters were created at the BBU pool using a centralised cloud, while local content was delivered by F-APs. Using stochastic geometry, the system's ergodic rate was derived. In addition, delay time and transmission latency were modelled with queueing theory. Using the embedded game paradigm, a system control method for FRAN was developed [101]. To maximise system efficiency, a collaborative design of spectrum allocation, cache placement, and service admittance algorithms was considered. In addition, [102] proposed a hybrid cloud and edge processing method to reduce FRAN downlink latency, while channel encoding and fronthaul compression methods were implemented to reduce fronthaul content delivery delay from the BBU to the requesting UEs. Similarly, in [103], a superposition coding approach that utilises fronthaul connections in both hard and soft transfer modes were proposed for the delivery phase design of any prefetching scheme. Using this design, caches of an enhanced RRH (eRRH) were generated in memory [103]. In [104], the Markov chain was used to model and determine the effect of mobile social networks on the effectiveness of FRAN edge caching. A combined distributed computing and content sharing strategy was proposed in [105] to reduce fronthaul connectivity issues and achieve ultra-low latency. In addition, [106] proposed a dynamic resource balancing strategy to improve system reliability in the FRAN.

2.3.3.1.2 Spectrum and Energy Efficiency

Various studies [107] have proposed the FRAN as a solution to the CRAN issues and to enhance the spectral and energy efficiency of 5G networks. Allowing a single user equipment (UE) to access a specific group of adjacent RRHs increases the spectral efficiency of CRANs in general. This group of surrounding RRHs is frequently established using a disjunct clustering method or a user-centric clustering approach, where the cluster generation is dependent on the reference signal's received power threshold. Although a disjoint clustering strategy can reduce inter-cell interference, periphery users may experience substantial interference from neighbouring clusters. Due to the absence of peripheral users, the user-centric approach reduces interference between clusters. To increase spectrum efficiency, CRAN performs power allocation and dynamic scheduling in addition to the massive CRSP. Users can readily access F-APs with content caching capabilities, unlike CRAN. In the FRAN system, the

distributing cluster is typically determined by the desired cached content of the user and the received signal strength of the F-AP at the UE. The spectral efficacy is subsequently enhanced based on user scheduling and power allocation.

Recent research in FRAN has sought to find a compromise between fronthaul capacity and transmit power. This is because increasing spectral efficacy could place a significant strain on the fronthaul, which has a limited capacity for data traffic. Considering this, [96] investigated a coordinated resource allocation and outsourcing scheme in a centralised FRAN. Under consideration of each UE's delay tolerance, fronthaul and backhaul capacity, and available resources, the objective was to offload all computational activities using the least amount of energy feasible. Accordingly, [108] proposed an energy-saving method that considered fronthaul and backhaul capacity limits in addition to the utmost delay tolerance. The method resulted in enhanced allocation of computing resources and transmission capacity for several UEs. The design of the multicast FRAN downlink was evaluated in [109]. There was discussion regarding information compression-based (soft) and information sharing-based (hard) front haulage. Each RRH was designated a storage module with its own set of capabilities for limited signal processing. Consequently, storing in-demand data during off-peak hours reduced network energy consumption while improving latency. In this study, the challenges and efficacy of mmWave-based FRAN access and fronthaul links were investigated [110]. The study investigated energy-efficient power distribution in an orthogonal frequency-division multiple access network to manage interference. To enhance the energy efficiency of the system, the optimisation issue was presented as a nonlinear programming problem. A gradient-based iteration technique was proposed to discover the most energy-efficient power allocation solution.

2.3.3.1.3 Resources Allocation

The following studies have examined resource allocation for FRAN in a 5G system: [101], [111]–[113]. A shared resource allocation and structured computation dispatching scheme was proposed in [101] to reduce energy costs and improve the allocation of computational resources for a variety of user equipment (UE). Moreover, the strategy intended to minimise the transmission power and splitting factor delivered to each UE. Additionally, off-loading options and allocation of computational resources were combined to enhance the likelihood of UE relocation in [111]. Using a mixed integer nonlinear programming (MINLP) strategy, the problem was solved, and a Gini coefficient based FCNs selection algorithm (GCFSA) was devised to provide a suboptimal off-loading strategy. Furthermore, the authors used a distributed resource optimisation approach based on a genetic algorithm (ROAGA) to address the computing resource allocation issue. Using an embedded game model, the authors of [112] designed a system control strategy and dynamic mode selection for the FRAN system. Co-

designed algorithms for cache placement, spectrum allocation, and service admittance improved the system's performance. In a similar vein, [113] presented a joint mode selection and resource allocation strategy for device-to-device (D2D) communication. The method increased energy efficiency while taking time and resource reuse constraints into account.

2.3.3.2 Challenges of FRAN

As with other 5G RAN designs, the FRAN continues to be plagued by significant research obstacles. Edge caching improves FRAN performance by reducing traffic congestion at the cloud server and enabling F-UEs to acquire content more quickly. Contrary to centralised caching schemes, the caching capacity of each F-AP and F-UE is limited in FRAN. Edge caching strategies, such as determining which content to cache and when to deliver caches in various edge devices, must be optimised to improve overall caching performance in FRAN. In addition, conventional caching principles, such as least recently and least frequently used and first-in-first-out, should be enhanced to increase the FRAN cache hit ratio. The network functions virtualization (NFV) facilitates the virtualization of the SDN controller to host a cloud server in FRAN. Consequently, the server can be relocated based on the network's requirements. However, the virtualization of the SDN controller in the 5G- FRAN system remains unsure. Due to the distributed nature of peripheral devices, this is the case. Moreover, security and privacy, computational complexity, scalability, fundamental operational and maintenance compatibility with existing RAN designs, among other research issues, remain unresolved.

2.4 The impact of machine learning 5G architecture

The development of a new radio access network (RAN) instead of evolving the existing RAN is due to the service scenarios of eMBB, uRLLC, and mMTC. The two new service scenarios of uRLLC and mMTC cater for the massive increase in IoT devices, which require data access that is delay sensitive. This new RAN will be a precursor for 5G new radio (NR). In [10], Cloud RAN (CRAN) architecture was proposed to provides a modification to centralized processing, collaborative radio, real-time cloud computing, which further translates to power-efficient design architecture. Also, the architecture combines all base station computational resources into a central base station (BBU) pool.

In the central pool or BBU pool, radio-frequency signals from geographically distributed antennas are collected by remote radio heads (RRHs) and transmitted to the cloud platform through the common protocol radio interface (CPRI). The CPRI interface, known as the fronthaul, is a design limitation for the CRAN architecture due to its limited capacity [114]. Moreover, a joint resource management technique that improves the system capacity with

reduced network design cost was proposed in [10]. Also, different wizhaul methods: Wizhaul-backtracking branch-and-bound (BBB), Wizhaul-greedy algorithm greedy routing (GA-GR), and Wizhaul-greedy algorithm randomized rounding routing (GA-RR), which contribute to maintaining the centralization of CRAN were considered in [114]. However, these algorithms may not be suitable solutions for the internet of thing (IoT) devices since they are delay sensitive.

Besides, the Fog radio access network (FRAN) architecture extends on the CRAN architecture by adding Fog access points (F-APs) and smart fog user equipment (F-UE) with local caches for storing frequently requested contents and signal processing capabilities. FRAN combines the advantages of fog computing and CRAN, which extends cloud computing to the network edge [115]. The cooperative processing reduces the load on the constrained fronthaul links while reducing queuing and transmission delay. However, the management of computational and communication resources (cost) remains a challenge in the FRAN architecture.

A joint resource allocation and structured computation offloading scheme for the FRAN was suggested in [96] to reduce the cost of energy and optimize computational resource allocation for several user equipment's (UE). The technique also aimed to reduce the amount of transmission power allotted to each UE as well as the splitting factor. In [101] a FRAN system control scheme based on the embedded game model was developed for maximizing system efficiency for spectrum allocation, cache placement, and service admission algorithms. The embedded game methodology further captures the dynamics of FRAN and effectively negotiations the centralization, while optimally using distributed intelligence for faster and less complex decision-making process. Therefore, a dynamic mode selection for FRANs, which compete with groups of potential users' space was formulated as a dynamic evolutionary game, and the game is solved by an evolutionary equilibrium in [112].. Moreover, the basis for RAN slicing was developed to mutually deal with content caching and mode selection within the dynamic channel state and cache status [116]. The cost-efficient method of network slicing enables mobile network operators (MNO) to innovatively propose their networks to clients as-a-service [117], [118]; thus, enhancing network flexibility and scalability.

In the FRAN, a major concern is to lower end-to-end latency. Hence, a joint cloud and edge processing architecture was investigated in [103], [105] to reduce FRAN downlink latency. In [103], the authors considered improving the delivery phase such that the least delivery rate of a requested file is optimized while satisfying the fronthaul capacity and power constraints of each enhanced remote radio head (RRH) in the system. Additionally, [98] examined the trade-off between efficiency, communication cost, and cost of computing in the FRAN architecture. The authors utilized mobile augmented reality (AR) services to demonstrate a participating FN collection and task assignment technique. According to the article, if the trade-off between

these variables is properly managed, FRAN may achieve ultra-low-latency services. FRAN with hierarchical content caching was proposed in [100], with each F-AP having its cache to enable a portion of demands to be handled locally. The average ergodic rate for transmitting information is calculated using probabilistic geometry-based theory. The system's delay and latency ratio were also calculated using queuing theory.

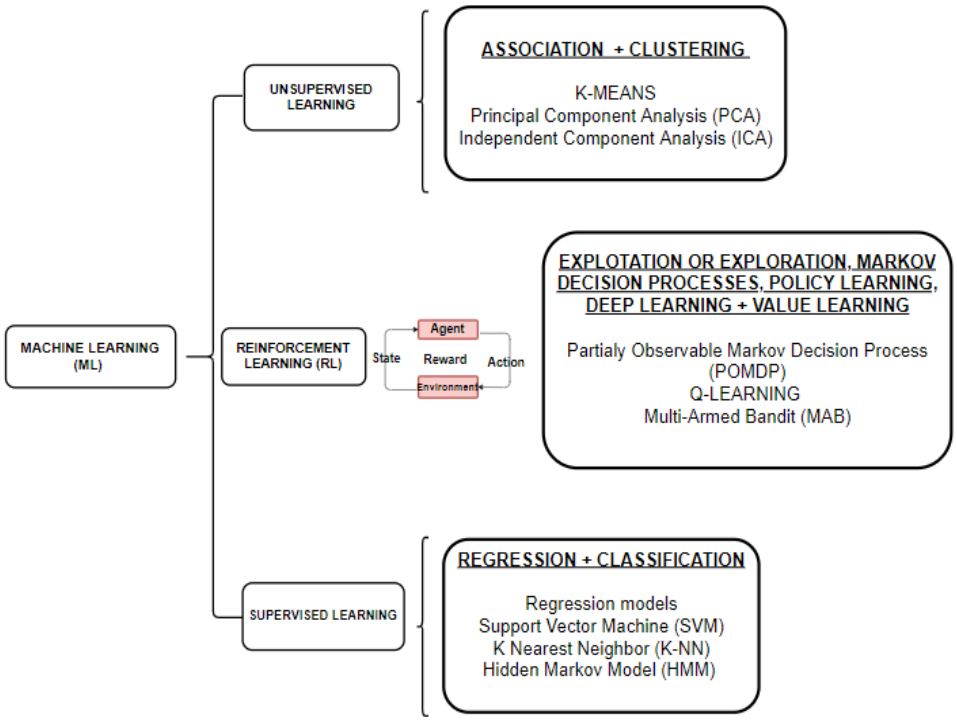

Figure 2:4 Machine learning techniques: supervised, unsupervised and reinforcement learning (*adapted diagram*). [139]

Moreover, there are still challenges of reducing computational complexity while dynamically allocating and optimizing the limited resources in the 5G-FRAN system. Therefore, machine learning (ML) techniques, which dynamically optimizes resource management in 5G-FRAN are investigated in this research. Figure 2.4 shows the three well-known techniques: unsupervised, supervised, and reinforcement learning.

2.4.1 Unsupervised (UL)

Unsupervised learning utilizes unstructured input data to create a relationship between the data set to perform a specific task [119]. If no consistent theoretical foundation exists for unsupervised learning, the algorithm attempts to learn through observation rather than explicit feedback [120]. Some examples of the well-known unsupervised learning algorithms are K-means clustering, principal component analysis (PCA), independent component analysis

(ICA), etc. In [121], the K-means clustering for 5G RAN within Heterogeneous networks (HetNets) was utilized for diverse cell sizes and Device to device communication (D2D). Additionally, a complex ICA with the PCA technique was implemented for smart meter data recovery in [122]. These techniques are promising for resource allocation in 5G network design and resource allocation management [119], [123]. Given the high demand for limited spectral resources, further research into new optimization techniques in this area would be extremely profitable.

2.4.2 Supervised Learning

Supervised learning is a type of machine learning technique that aims to learn a mapping function between a known input and output data set. The generalization capability of the supervised learning technique is heavily reliant on a training phase. The training phase uses known inputs and develops a structured algorithm to map the data to the desired output scenario. After that, the ML technique is tested on a new data set with the same distribution as the original training data set. This analysis produces a training error, which is used to fine-tune and train the supervised learning algorithm [119].

Supervised learning algorithms are anticipated to be utilized as classification methods for structured input data groups to output data groups for QoE optimization, traffic steering, QoE-based traffic steering, and Vehicle-to-Everything (V2X) handover management. In [124], the K-nearest neighbours (KNN) was proposed as a supervised learning method to jointly optimize gateway and virtual-channel placement, reduce the overall wireless transmission and encourage utilization of optical infrastructure in the 5G networks. A hierarchical support vector machines (H-SVM) scheme was introduced in [125], which distribute resources to classes in a multidimensional hyperplane based on the network information such as the hop counts, without requiring particular ranging hardware. Additionally, the HMM was proposed as a robust supervised learning algorithm for resource scheduling [126], [127], device cell association in 5G networks [128], and traffic prediction in 5G networks [129], [130]. [131] employed the HMM to minimize handover delays in Cognitive 5G Cellular Networks. Handover management is dependent on several conditions that determine when a handover should occur. Enhancing the handover process is critical to meeting 5G's 1ms latency requirement. Therefore, the HMM is proposed as an ideal for learning, implementing, modifying, and ultimately optimizing the resource management issues in the 5G-RAN network.

In this research, the HMM will be implemented to effectively model and optimize resource allocation, ultra-low latency, spectrum, and energy efficiency issues in the 5G FRAN networks. To the best of the author's knowledge, the HMM has not been implemented for this purpose. More so, the combination of HMM with unsupervised ML (hybrid) will be investigated to

enhance the performance of the proposed algorithm while considering the computational cost. The results of the proposed HMM technique for resource management will be compared with existing machine learning algorithms in the literature.

2.4.3 Reinforcement Learning (RL)

Reinforcement learning utilizes a known input training data set and creates mapping rules for a known output data set with the aim of optimizing long-term conditions [132], [133]. To achieve this, the algorithm interacts with environmental data, which is theoretically like unsupervised learning. Through this environmental interaction, feedback loops are used to periodically update based on changes to the wireless network environment [134]. Allowing for data set changes over time bestows the opportunity to modify and optimize operation [119].

The class of reinforcement learning techniques such as Partially Observable Markov Decision Process (POMDP), Q-Learning, and Multi-Armed Bandit (MAB) provides much promise for 5G network implementation by modifying network settings to achieve the best outcomes [120]. In [135], the POMDP was used to solve the transmission power control problems for energy harvesting sensor nodes (EHS). Also, [136] designed a cell outage management (COM) framework for HetNets with a split in the control and data planes of the 5G base stations. A MAB-based network-assisted distributed channel selection method was proposed in [137]. This technique allows device-to-device (D2D) users to utilize vacant cellular channels [137]. The scenario was modelled as a multi-player multi-armed bandit game with side information, for which a distributed algorithmic solution was implemented.

Furthermore, the deep reinforcement learning (DRL) algorithms for FRAN resource managements in 5G networks were examined in [116], [138]–[147]. These techniques have a significant advantage over the straightforward approach of deciding based on a fixed threshold of utility since they quickly learn the optimal decision thresholds from the IoT environment. Thus, the DRL method achieves an improved performance regardless of the environment.

Despite the performance enhancement that the RL and DRL methods provide in the dynamic systems IoT environment, there is still a trade-off between the resource management performance and computational efficiency of these machine learning algorithms. Hence, the need to implement other classes of machine learning techniques, such as supervised and unsupervised machine learning to obtain an enhanced (in terms of performance and computational complexity) resource management technique for Fog RANs in 5G (5G-FRANs).

2.5 Fog Computing Applications

Fog computing allows for the close accessibility of cloud computing services to the client, hence enabling the provision of assistance for applications that are sensitive to delays in data transmission [148][149]. Examples of such applications include those used in self-driving cars, computerized systems, and digital banking. Certain processing deadlines govern the above applications, and the accuracy and reliability of their results hinge on their availability. If a real-time application fails to meet its specified deadline, the consequences can vary greatly, including outcomes that range from rendering the application's results useless to causing catastrophic events [150]. Furthermore, fog computing offers assistance for applications that necessitate mobility, geographic dispersion, positional comprehension, and real-time analytics [151] (Wang et al., 2018). [95] assert that fog nodes (FNe) or the cloud can process non-real-time applications. In this context, FNe with limited resources can integrate data from the Internet of Things (IoT) [152]. On the other hand, applications used in medical facilities and high-definition (HD) video broadcasting need a lot of storage space and computing power since they must handle large amounts of data [153][154].

The Internet of Things (IoT) technology has become increasingly important in numerous facets of everyday lifestyle. Thus, IoT and its related technologies have expanded Internet connectivity to include a wide range of items such as sensors, machinery, cars, and wearable devices [155]. These devices can perform a range of activities, such as health monitoring, intelligent traffic control, smart retail, logistics, and vehicle networking. To get the information required for IoT applications, the data generated by these devices must be evaluated. However, IoT devices' processing and storage capacities are insufficient to handle this volume of data [156]. Cloud computing, on the other hand, is a distributed computing solution that provides a pool of resources on demand over the Internet [157]. The transfer of resource-intensive tasks to more capable cloud nodes can overcome the inherent limitations of IoT devices of short battery life, low computational power, and limited storage [158].

Based on the findings in [159], it is projected that the global usage of connected devices would exceed 50 billion by the year 2030. This will lead to the establishment of an extensive network including various interconnected devices, encompassing automobiles, cellphones, household appliances, and other similar entities. Numerous designs and technological advancements have been proposed in the academic literature, specifically focusing on the Radio Access Network (RAN) for 5G networks [160]. The primary objective of these designs is to meet the growing demands for capacity while simultaneously reducing both Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) in network infrastructure. Additionally, these designs aim to provide ultra-low latency and enhanced bandwidth to many end-users, while also incorporating and integrating existing technologies.

-

Thus, the integration of Fog and Cloud will create a distributed computational environment. This environment will consist of nodes with varying processing and storage capacities. These nodes can be linked through numerous switches and links at different layers [161][162]. Furthermore, fogs are made up of nodes that are densely distributed. These nodes can include proxies, mini-clouds, smart edge devices, routers, cellular base stations, and access points [163].

The technology company Cisco originally suggested fog computing to alleviate the burden on cloud data centres by extending cloud computing to the network's periphery [164]. Fog computing encompasses several products, such as controllers, telecom base stations, embedded servers, access points, gateways, switches, routers, and surveillance cameras [165]. It offers numerous advantages, such as conserving network bandwidth, decreasing energy consumption, supporting the mobility and geographical distribution of IoT devices, and achieving location awareness and low latency for delay sensitive IoT applications [164], [166]. However, the computing and storage capacities of FNe are insufficient for significant Internet of Things applications, such as big data analytics. Consequently, cooperation between fog and cloud locations is required. This integration introduced a new computing paradigm known as cloud–fog computing [167].

2.5.1 Classification of Service Types

The classification of service (CoS) is a technique for classifying and ranking network traffic or services according to their unique specifications and features. It enables the treatment of various traffic classes differently, guaranteeing that each class obtains the proper degree of service and resource allocation. Figure 2.5 illustrate the architecture of network design and QoS implementation, the idea of CoS is frequently utilized. It entails allocating distinct types of network traffic or services varied levels of priority, treatment, and resource allocation according on their demands or requirements. Network administrators may efficiently manage network resources, guarantee optimum performance, and satisfy the quality requirements of various applications or services by using CoS as list in Table 2.1.

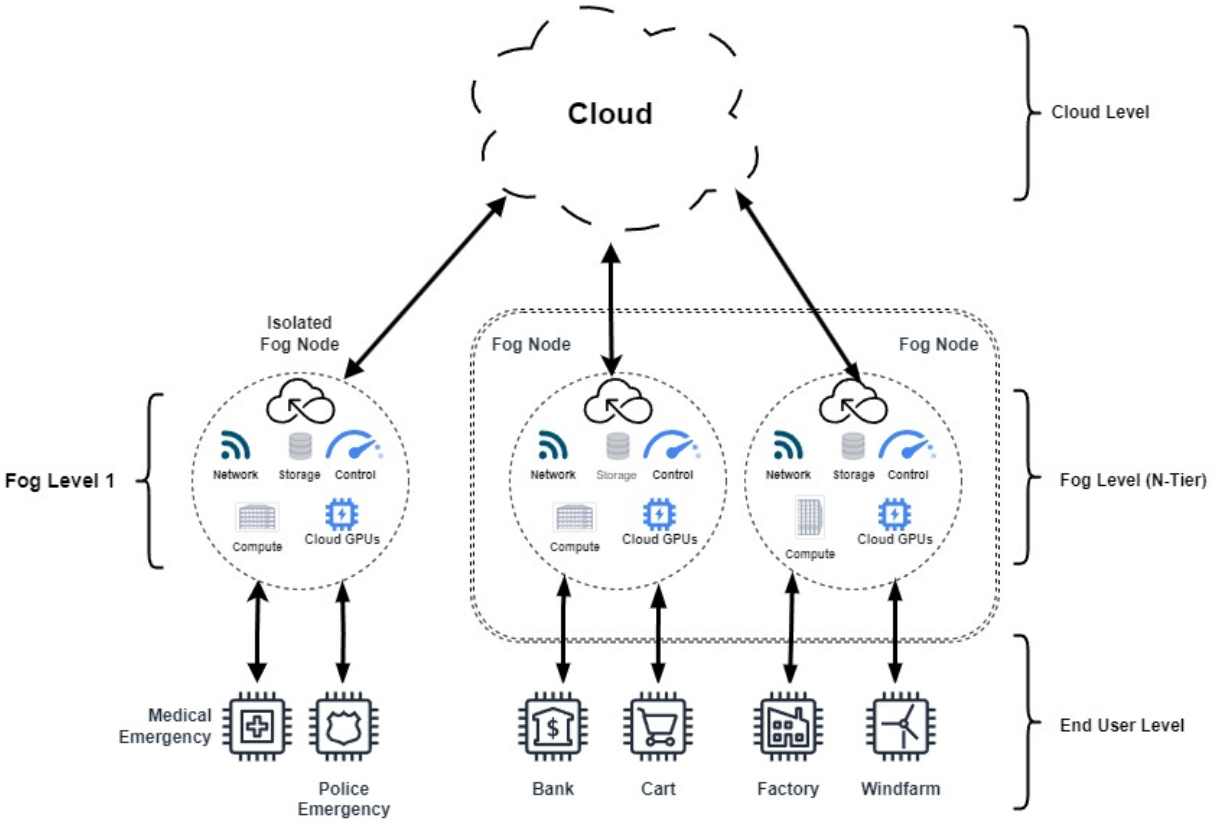

Figure 2.5: Fog Computing Network diagram with Isolated FNes

Table 2.1: Class of Service (CoS) properties descriptions

CoS	Description of Class Service Class
1. Mission Critical	This category encompasses applications characterized by a short time frame between events and corresponding actions, adherence to regulatory requirements, implementation of security measures comparable to those employed by the military, and prioritization of privacy. Additionally, it includes applications where the failure of a single component results in a substantial escalation of safety risks for individuals and/or the surrounding environment. The examples provided encompass many sources of traffic, such as military forces combat systems, drone activities, certain healthcare systems, healthcare facilities, and ATM banking systems [94]. If the network possesses the capability to accommodate priority services, it is advisable to allocate a high level of priority to this kind of service. Additionally, it is important to consider the inclusion of applications pertaining to the analysis of data streams to identify items or hazardous events under this category.
2. Realtime	Real-time applications are subject to limits related to delays. Several examples of applications include industrial control systems, Internet of Things (IoT) and smart cities applications, online gaming, as well as virtual and augmented reality. Within the realm of interactive applications, the user can assume the role of either an end device or an individual. Some examples of these services are interactive television, object hyperlinking technologies like RFID, NFC, and QR-Code, as well as data transaction services like e-commerce. Conversational applications encompass the facilitation of real-time discourse among individuals or collectives. Conversational apps exhibit a sensitivity to delays while also demonstrating a tolerance for data loss. The quality of service (QoS) needs is determined by the human perspective. Two examples of the final category include Voice over Internet Protocol (VoIP) and videoconferencing.
3. Interactive	In this scenario, the importance of responsiveness cannot be overstated, since it is imperative that the elapsed time between the user's request and the resulting actions executed on the client side is limited to a few seconds or less. Furthermore, the consumers of interactive apps might encompass both end devices and persons. This class encompasses various applications, including interactive television, web surfing, database retrieval, server access, automatic database inquiries by tele-machines, pooling for measurement collecting, and certain Internet of Things (IoT) deployments.
4. Conversational	These applications encompass several forms of video and Voice-over-IP (VoIP) technologies. These entities possess the attribute of being sensitive to delays while also being tolerant of losses. Humans can see delays shorter than 150 milliseconds, while delays ranging from 150 to 400 milliseconds may be deemed acceptable. However, delays over 400 milliseconds render speech communications entirely unintelligible. Conversely, it should be noted that conversational multimedia systems exhibit a certain degree of tolerance towards loss, wherein sporadic losses may result in occasional disruptions in the playback of audio or video. However, it is worth mentioning that such losses may often be mitigated to some extent or completely masked [168].
5. Streaming	The course encompasses the study of applications involving extended file transfers, including the streaming of archived content and the streaming of real-time content. In the context of applications pertaining to the transmission of archived information, the content is preexisting and subsequently kept on servers. Users express their desire to obtain

-

	accessibility to this content whenever they require it and engage in interactions with the content. YouTube, Netflix, and Hulu are prominent entities in the streaming video industry. Applications that use the streaming of live material enable users to engage in both the transmission and reception of real-time multimedia events through the Internet.
6. CPU-Bound	The class is utilized by applications that include intricate processing models, particularly in decision-making scenarios, which may need extensive periods of time ranging from hours to days or even months. Some instances of CPU-bound applications include face recognition, animation illustration, speech analysis, and dispersed camera networks.
7. Best Effort	In the context of Best-effort applications, extended periods of latency may be regarded as inconvenient but not inherently detrimental. Conversely, the preservation of the entirety and accuracy of the sent data holds utmost significance. Several instances of the Best-Effort class include the retrieval of electronic mail, real-time messaging, the transmission of short message service (SMS), the transfer of files via the File Transfer Protocol (FTP), peer-to-peer file sharing, and machine-to-machine (M2M) communication.

2.5.2 QoS Requirement Parameters

Table 2.2: Quality of service requirements for Fog computing applications

QoS Requirements	Nominal Categories	Class of Service						
		MC	**RT**	**IN**	**CO**	**ST**	**CB**	**BE**
Bandwidth (Mbps)	Low Medium High	$0 < x \leq 1$	$5 < x \leq 1000$	$0 < x \leq 1$	$0 < x \leq 1$	$1 < x \leq 5$	$1 < x \leq 5$	$0 < x \leq 1$
Reliability	Low Important Critical	$x = 3$	$x = 2$	$x = 1$	$x = 2$	$x = 2$	$x = 2$	$x = 1$
Security	Low Medium High	$x = 3$	$x = 2$	$x = 3$	$x = 2$	$x = 2$	$x = 3$	$x = 1$
Data storage (h)	Transient Short duration Long duration	$0 < x \leq 1$	$0 < x \leq 1$	$0 < x \leq 1$	$0 < x \leq 1$	$730 < x \leq 8760$	$1 < x \leq 730$ $730 < x \leq 8760$	$730 < x \leq 8760$
Data Location (ms)	Local Vicinity Remote	$0 < x \leq 10$	$0 < x \leq 10$	$10 < x \leq 20$	$10 < x \leq 20$	$20 < x \leq 100$	$20 < x \leq 100$	$20 < x \leq 100$
Mobility (Km/h)	Low Medium High	$0 < x \leq 5$ $5 < x \leq 25$ $25 < x \leq 100$	$0 < x \leq 5$	$0 < x \leq 5$	$25 < x \leq 100$	$5 < x \leq 25$ $25 < x \leq 100$	$0 < x \leq 5$	$0 < x \leq 5$
Scalability (No. of IoT user/end users)	Low Medium High	$120 < x \leq 200$	$0 < x \leq 60$	$120 < x \leq 200$	$120 < x \leq 200$	$60 < x \leq 120$	$60 < x \leq 120$	$120 < x \leq 200$
Delay sensitivity (Interaction latency in ms)	Low Moderate High	$0 < x \leq 10$	$0 < x \leq 10$	$10 < x \leq 10^3$	$10 < x \leq 10^3$	$10^3 < x \leq 10^4$	$10^3 < x \leq 10^4$	$10^3 < x \leq 10^4$
Loss sensitivity (PERL)	Low Moderate High	$0 < \mathrm{x} \leq 10^{-6}$	$10^{-6} < \mathrm{x} \leq 10^{-3}$	$0 < x \leq 10^{-6}$	$10^{-3} < x \leq 10^{-2}$	$10^{-6} < x \leq 10^{-3}$	$0 < x \leq 10^{-6}$	$0 < x \leq 10^{-6}$

2.6 Fog Computing Applications Service Level Agreement

Several studies [169]–[180] have been conducted to determine the application requirements necessary to construct service architectures for both Cloud and Fog Computing. Research has been done on Service Level Agreements (SLA) [169], [170], [173], [174] and Quality of Experience (QoE) management [175] for cloud computing. In the meantime, Fog Computing processing and analysis applications [176], assignment of applications to resources [177], resource estimation [178] and allocation [179], [180] for application processing, and service placement [171], [172].

The SLA is a document that contains a description of the agreed service, service level parameters, guarantees, actions and remedies in the event of service level violations [174]. As a contract between the consumer and the provider, the SLA is extremely essential. The primary purpose of SLAs is to provide a precise definition of formal agreements concerning service terms such as performance, availability, and invoicing. Thus authors [169] in focused on the non-functional service requirements such as availability, scalability, and response time.

In accordance with the type of service, authors in [181] develop a resource management framework for fog computing. In this framework the Cloud service provider (CSP) was able to link virtual resource values, and the physical resource pool based on the type of service through mapping between the two. Meanwhile Video on Demand (VoD) and health monitoring were examples of services used for demonstration. Furthermore, in the Fog-Cloud context, [182] a suggested integer linear programming (ILP) model. That used the objective of planning scenarios based on the capacity (number of positions) required by each service to be allocated is to optimize latency. To accomplish this, services are divided into two groups: mice, representing many services requiring few slots, and elephants, representing a small number of services requiring a greater number of slots. In [183] a model for resource prediction in Fog, customer type-based resource estimation and reservation, advance reservation, and pricing for new and existing IoT customers, based on their characteristics and the type of customer-defined by the number of times a specific service is requested by the same client.

2.7 The Classification of Service For Fog Applications

In the context of telecommunications systems, the collection of network flows of resources and traffic profiles can be categorized into two distinct classes. These groups are often recognized as integrated service (Intserv) and differentiated service (Diffserv), collectively referred to as classification of service (CoS). The CoS is employed to establish the parameters governing the operation of traffic control systems, including schedulers and queue managers. While the rules

for Quality of Service (QoS) (see Table 2.2.) provisioning encompass the definition of CoS and traffic control mechanisms to cater to the diverse QoS requirements of network flows. In a separate study [184], a significant contribution is presented to the establishment of the Classes of Service for Fog computing, considering the QoS demands.

It is thus crucial for the efficient provisioning of services that the demands of applications arriving at the edge of the network be well understood and classified so that resources can be assigned for their processing. The mapping of applications to CoS should facilitate the matching between task requirements and resources, since labelling these tasks removes the burden of the analysis of the application requirements by the scheduler. Without a precise classification, the scheduling of application tasks and the allocation of resources can be less than optimal due to the complexity in dealing with the diversity of QoS and resource requirements. The practice of assigning applications to specific CoS is commonly observed in communication network technologies that facilitate QoS such as LTE, 5G,[185] and ATM [186] networks. This approach is crucial for network operators as it enables them to deliver various levels of service quality.

Furthermore, it is imperative to employ effective classification techniques that can effectively handle the unique attributes of the cloud-to-things (C2T) environment. The classification of network traffic has been the subject of substantial research over the past few decades to facilitate the provision of secure network services [187][188]. Nevertheless, there has been a lack of emphasis on the categorization of the demands and prerequisites of applications and services on the Internet [189]. Moreover, device mobility and the IoT have introduced new applications to the Internet, and these applications have not been considered in any of the classification schemes. This research paper presents a novel approach that utilizes the Hidden Markov model to define a comprehensive set of CoS for Fog computing. The proposed framework considers the QoS requirements of various Fog applications, enabling the distinction of demands from a wide range of services.

Table 2.3: Quality of Service (QoS) properties descriptions

QoS	Description of features
1. Latency	The concept of latency. Certain applications require the maintenance of a set threshold for latency, wherein latency must be guaranteed, particularly for applications that operate in real-time. It is imperative to establish delay limitations for real-time applications, such as facial recognition in crowded environments.
2. Scalability	Scalability is related to the capability of an application to work efficiently, even in the presence of an increasing number of requests from end users. The population of users within a Fog computing environment is subject to variability because of user mobility, as well as the activation of apps or sensors. The processing of data streams in big data applications may necessitate adherence to predetermined time constraints. The demand placed on FNes may experience fluctuations, necessitating the provision of resource flexibility to effectively accommodate these varying demands.
3. Bandwidth	The capacity or range of frequencies that can be sent or processed inside a communication system. Certain applications require a minimum assured data transfer rate, commonly referred to as a Guaranteed Bit Rate (GBR). Multimedia applications exhibit sensitivity to bandwidth, with certain applications employing adaptive coding techniques to encode digitized speech or video content at a rate that aligns with the presently accessible bandwidth.
4. Data Location	The concept of data location pertains to the designated storage place for application data. Data can be kept in several locations, including the local storage of the end device, near the device at a FNe, or in a remote repository located in the Cloud. The data placement requirements for an application are contingent upon various factors, including reaction time limitations, the computing capabilities of each Fog layer, and the available capacity on network lines.
5. Data Storage	The concept of accessibility in the context of data storage refers to the frequency at which the resources of the Fog computing environment may be accessed by end-users. Applications and services that require continuous operation, such as mission-critical applications, necessitate the implementation of high availability measures.
6. Security	Security encompasses the strategic development and practical execution of methods for verifying the identity and granting access privileges to safeguard sensitive and essential data produced by those utilizing a system.
7. Reliability	The concept of reliability pertains to the capability of Fog components to successfully execute the intended operation while encountering various forms of failure. Certain applications require the rapid reestablishment of failing Fog components to ensure that actions can be executed within specific latency constraints.
8. Packet loss	Loss sensitivity refers to the percentage of packets that fail to reach their intended destination. Lossless transfer services may be required for loss-sensitive applications, such as those involving financial data.
9. Mobility	The attribute of mobility is inherent to numerous edge devices. It is imperative to guarantee the continuity of the services provided, particularly for end-users who are very mobile. Uninterrupted connectivity is necessary for the requisite processing.

-

2.7.1 Fog Computing Application Classification Based on Machine Learning Methods

This section provides a detailed explanation of state-of-the-art techniques for classifying Fog applications using the CoS. *Additionally, the proposed Machine Learning models (ML) classification method is discussed in detail. The ML categorises Fog computing applications based on their QoS requirements*. ML techniques are categorised into supervised, unsupervised and reinforcement learning based on data input for to analyzed.

2.7.2 Unsupervised Machine learning

Unsupervised learning is a machine learning methodology that uses unstructured input data to develop a relationship within a certain dataset, hence enabling the system to perform a specified task [119]. The algorithm resorts to observational learning instead of explicit feedback due to the lack of a comprehensive theoretical underpinning for unsupervised learning [120]. There are several unsupervised learning methods that have gained widespread recognition, namely K-means clustering, principal component analysis (PCA), and independent component analysis (ICA).

2.7.3 Artificial neural network (ANN)

Artificial Neural Networks (ANNs) are a computational framework characterized by nonlinearity, which enables them to perform machine learning tasks such as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. These capabilities make ANNs applicable in diverse wireless networking scenarios [184][190][191][192].

2.7.4 Supervised Machine Learning

Supervised learning is a machine learning approach that attempts to acquire a mapping function from a predetermined input and output dataset. The efficacy of the method of supervision in making assumptions is highly dependent on the quality and comprehensiveness of the training phase. During the training phase, the algorithm is constructed by utilizing known inputs to establish a systematic mapping of the data to the intended output scenario. Subsequently, the machine learning technique is evaluated on a novel dataset that possesses an identical distribution to the initial training dataset. The present study yields a training error, which is afterward utilized for the purpose of refining and training the supervised learning algorithm [119].

2.7.4.1 K-Nearest Neighbors (K-NN)

The class of estimators known as nearest neighbor estimators adjusts the level of smoothing based on the density of data in the nearby vicinity. The level of smoothing is determined by the

-

parameter k, representing the number of neighboring data points considered. It is important to note that k is significantly smaller than N, the total sample size. In this context, we shall establish a metric to quantify the distance between two entities, denoted as a and b. For instance, we can utilize the absolute difference, $|a - b||$. Additionally, for each given value x, we can construct a series of distances in ascending order, representing the distances from x to the various points inside the sample. Specifically, $d_1(x)$ denotes the distance from x to the nearest point in the sample, $d_2(x)$ represents the distance to the subsequent nearest point, and so forth. Let us consider a set of data points, denoted as x_t. We can define a function $d_1(x)$ as the minimum absolute difference between x and any xt. Furthermore, we can determine the index i of the closest sample by finding the argument that minimizes the absolute difference between x and x_t. Using this information, we can define another function $d_2(x)$ as the minimum absolute difference between x and any other sample x_j, where j is equal to i. This process can be repeated iteratively to define subsequent functions, such as $d_3(x)$, and so on.

2.7.4.2 Decision Tree (DT)

A decision tree is a hierarchical model used in supervised learning, where the local region is determined through a series of iterative breaks, resulting in a more efficient process with fewer steps [193]. A decision tree consists of internal decision nodes and terminal leaves. Every decision node, denoted as m, incorporates a test function, $f_m(x)$ which produces discrete outcomes that are used to designate the branches. When provided with an input, a test is executed at each node, resulting in the selection of one branch based on the test outcome. The method commences from the root and is iteratively executed until a leaf node is encountered, at which juncture the value inscribed within the leaf serves as the resultant output. The decision tree can be classified as a nonparametric model due to its lack of assumptions regarding the parametric form of class densities. Additionally, the structure of the tree is not established in advance but rather evolves during the learning process by adding branches and leaves based on the inherent complexity of the data.

2.7.4.3 Support Vector Machine (SVM)

Support vector machines (SVMs) are widely utilized in several domains for addressing classification, regression, and novelty detection tasks. One notable characteristic of support vector machines is that the estimation of the model parameters aligns with a convex optimization problem, so ensuring that any local solution is also a global optimum. The support vector machine (SVM) algorithm has been widely studied and discussed in the academic literature by prominent researchers such as [194], [195], [196], [197], [198], and [199]. The Support Vector Machine (SVM) is a classification algorithm that operates as a decision machine, meaning it assigns data points to certain classes without providing posterior probabilities.

-

2.7.5 Reinforcement Learning

Reinforcement learning is a computational approach that uses a given set of input training data to establish matching patterns for a known output data set, with the ultimate objective of optimizing long-term circumstances [132], [133]. To accomplish this objective, the algorithm interfaces with ambient input, presenting characteristics like unsupervised learning in a theoretical sense. According to [134], feedback loops are employed in this environmental interaction to regularly update in response to alterations in the wireless network environment. The inclusion of dynamic data sets enables the possibility to adapt and enhance operations [119].

Other studies [120] [135] [137] on reinforcement learning approaches, including Partially Observable Markov Decision Process (POMDP), Q-Learning, and Multi-Armed Bandit (MAB), hold significant potential for the development of 5G networks. These techniques offer the ability to optimize network parameters to attain optimal outcomes according to [120]. In [135] the authors employed the Partially Observable Markov Decision Process (POMDP) framework to address the challenges associated with transmission power control in energy harvesting sensor nodes (EHS). Meanwhile, in [136] a cell outage management (COM) paradigm for heterogeneous networks (HetNets) is suggested, wherein the control and data planes of the 5G base stations are separated.

Whereas for research related to reinforcement and Fog radio access networks we have the following [116], [138]–[147] which dealt with resource management algorithms for 5G. These strategies possess a notable benefit compared to the conventional method of determining decisions based on a predetermined utility threshold, since they efficiently acquire the optimal decision thresholds from the Internet of Things (IoT) environment. Therefore, the Deep Reinforcement Learning (DRL) approach demonstrates enhanced performance irrespective of the specific environmental conditions.

2.8 Performance Evaluation

The measures employed to assess the efficiency of each approach created and evaluated differ. Various techniques have been established for the identification and categorisation of 5G FRAN applications. The outputs of detectors and classifiers are defined by certain parameters that influence their performance level. Nonetheless, the reporting methodology for performance metrics differs across scholarly articles. This variety might be ascribed to the utilisation of diverse approaches for evaluating network applications.

Similar to all scientific research projects, the findings in this field are communicated using quantifiable measurements. These metrics demonstrate the accuracy attained by the designed

-

detector and classifier. Although several standard reporting metrics are available, certain writers opt to employ their own reporting styles for findings. No technique is flawless; any technique can deliver false negatives or false positives, depending on the applications assessed, the FE methodology utilised, or the detectors and classifiers implemented.

The common metrics used in reporting the outcomes of detection and classification processes include accuracy (ACC), error rate (ERR), false positive rate (FPR), true positive rate (TPR), and precision (PREC). These metrics are dependent upon the four result variables in a binary classification algorithm. The four result criteria are specified as follows:

- True positives (TP): This refers to the count of occasions where a detector's output aligns with the class of interest as determined by an experienced professional.
- True negatives (TN): This refers to the accurate identification of the absence of the relevant class by the detector.
- False positives (FP): This occurs when a detector inaccurately identifies an application as one of interest, specifically referring to the quantity of classes misidentified and categorised as relevant classes.
- False negatives (FN): This transpires when the relevant applications are overlooked. False negatives are often referred to as missed calls.

The performance measuring metrics are defined as follows:

1. **False positive rate (FPR):** This is the proportion of applications of interest that are wrongly detected by the detector; it can be mathematically expressed as:

$$FPR = \frac{FP}{TP+FP} \tag{2.1}$$

The best FPR is 0.0 while the worst is 1.0.

2. **True positive rate (TPR):** This is the proportion of classes of interest that are correctly detected by the detector. The TPR is also known as sensitivity or recall. It can be mathematically expressed as:

$$TPR = \frac{TP}{TP+FN} \tag{2.2}$$

The best TPR is 1.0 while the worst is 0.0.

3. **Error rate (ERR):** This is computed as the total number of all wrong predictions (FP and FN) divided by the total number of prediction outcomes (FP, FN, TP, TN) of the dataset; it can be mathematically expressed as:

$$ERR = \frac{FP+FN}{Total\ number\ of\ predictions} \tag{2.3}$$

The best ERR is 0.0 while the worst is 1.0.

4. **Accuracy (ACC):** This is computed as the total number of correct predictions (TP and TN) divided by the total number of prediction outcomes (FP, FN, TP, TN) of the dataset; it can be mathematically expressed as:

$$ACC = \frac{TP+TN}{Total\ number\ of\ predictions} \tag{2.4}$$

The best ACC is 1.0 while the worst is 0.0.

5. **Precision (PREC):** This is computed as the total number of correctly detected class of interest divided by the total number of detection (both TP and FP) in the dataset; it can be mathematically expressed as:

$$PREC = \frac{TP}{TP+FP} \tag{2.5}$$

The lower the number of FP predictions, the better the PREC value. Therefore, the best PREC value is 1.0, while the worst is 0.0. Other metrics are used to further evaluate the performance of detection and classification models. These metrics rely on combinations of the above metrics with the goal of gaining more insights and drawing conclusions about the performance of models. They include:

6. **Receiver-Operating-Characteristics curve (ROC):** This is a plot of the FPR on the x-axis against the TPR on the y-axis at various probability thresholds. It indicates the sensitivity and specificity of a model at different thresholds, thereby enabling the selection of a point that establishes the right trade-off.

7. **Area Under the Curve (AUC):** This is also a graphical metric that is used to assess the overall performance of models, often associated with the ROC curve. The AUC represents the area under the ROC curve, ranging from (0, 0) to (1, 1). The combination of the ROC curve and AUC reveals how well a model can effectively differentiate between positive and negative predictions.

-

8. **Precision-Recall (PR) Curve:** This is another graphical metric that involves the plotting of the recall (TPR) on the x-axis against the precision (PREC) on the y-axis. It is particularly useful when working on imbalanced datasets, where one class is dominant over the other.

9. $\boldsymbol{F_1}$ **Score**: This is defined as the harmonic mean of precision (PREC) and recall (TPR) and can be mathematically represented as:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.6}$$

The best F_1 score is 1.0, while the worst is 0.0. The F_1 score integrates PREC and TPR into a single metric to provide more insight about the model's performance.

Each designed detector or classifier has a unique set of thresholds or sensitivity, subject to what it is intended to be achieved. For instance, in a survey of a relatively divers application or use cases such as the class of service for fog applications, the detector may be configured in such a way that there are as few missed calls (FN) as possible due to the availability of few datasets, but such a detector may have many FP. On the other hand, in a survey of a common species with abundant datasets available, where an accurate index of detection or classification is important, the detector can be configured in such a way that the sensitivity level is low to reduce the number of FP detections and achieve high TPR [200].

Generally, it is common practice to adopt more than one metric to evaluate the performance of a model. This is because of the limitations (or biases) that are associated with each of the metrics and the characteristics of the dataset involved. For instance, the ACC performs poorly in evaluating the performance of an imbalanced dataset, while a metric like the F1 score combines PREC and TPR into one metric to provide more insight about the model's performance on an imbalanced dataset. Moreover, the PR curve has been suggested to be more advantageous than the ROC curve because of its ability to work on a skewed dataset, owing to its independence on TN predictions [44]. In summary, the choice of performance metrics to be adopted to evaluate the performance of a model will depend on the specifics of the problems and the goal of the model.

2.9 Performance Metrics

CLASSIFICATION
PERFORMANCE EVALUATION

Accuracy

$$\mathrm{A} = \frac{\mathrm{TruePositive(TP)} + \mathrm{TrueNegative(TN)}}{\mathrm{TotalSample}} \tag{2.7}$$

-

The F1-score is a statistic that incorporates precision and recall, so providing a more comprehensive evaluation. The utilisation of this technique proves to be particularly advantageous when dealing with skewed datasets. The F1-score is a metric used in evaluation that calculates the harmonic mean of accuracy and recall. accuracy is determined by dividing the number of genuine positive results by the total number of positive results, while recall is determined by dividing the number of true positive results by the total number of positive results that should have been returned.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}} \tag{2.8}$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \tag{2.9}$$

$$\mathbf{F1 - Score} = \mathbf{2} \times \frac{\mathbf{Precision \times Recall}}{\mathbf{Precision + Recall}} \tag{2.10}$$

2.10 Performance Metric

2.10.1 DT, SVM, K-NN, RF and ANN performance for difference τ.

The machine learning algorithms tested were Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbours, and Random Forest (RF) performance in terms of application detection. The finding and result analysis process is presented in for different dimension, τ with regards to accuracy, α, sensitivity, δ, false discovery rate (FDR), and area under curve (AUC) as defined in [201]. Accuracy, α measures the proportion of all instances positive and negative that correctly classified, defined by (2.11).

$$\boldsymbol{\alpha} = \frac{\mathbf{TP+TN}}{\mathbf{TP+FP+TN+FN}} \tag{2.11}$$

Meanwhile, true positive (TP) refers to the number of accurate positive forecasts, while false positive (FP) represents the number of inaccurate positive predictions. True negative (TN) indicates the number of accurate negative predictions, while false negative (FN) represents the number of inaccurate negative predictions.

The sensitivity, also known as recall, δ, is calculated as the number of true positives (TP) divided by the sum of true positives (TP) and false negatives (FN) in the real positive examples. The sensitivity, as described in references [202], refers to the accuracy of the detector in detecting signals or classifying categories. A higher sensitivity number indicates a higher level of detection correctness. This may be expressed using the following formula: 2.12.

$$\boldsymbol{\delta} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \tag{2.12}$$

-

Conversely, a low value of the FDR, ϑ, is preferred because it signifies that the output of the detector is reliable. As indicated in (2.13).

$$\mathfrak{\vartheta} = \frac{\mathbf{FP}}{\mathbf{FP+TP}} \tag{2.13}$$

2.10.2 Decision Trees for Fog Computing application CoS

Within the realm of fog computing, a decision tree can function as a robust hierarchical framework for effectively organising and processing data across remote FNes [203]. Every FNe has the capability to function as a decision node, where it carries out a particular test function $f_m(x)$ to evaluate incoming data and choose the appropriate branch or node in the network for further data processing [193]. In this procedure, like a conventional decision tree, the starting point is the root node. This node can either represent the central cloud or a high-level FNe. The process then proceeds recursively down the hierarchy until it reaches a terminal leaf node. This leaf node serves as the ultimate processing point or decision-making entity inside the fog network, producing the intended outcome or executing the necessary action. This approach enables a flexible and data-driven model in fog computing, where the tree structure adjusts to the intricacy of the processed data, optimizing the allocation of tasks throughout the network. The tree expands by incorporating branches and leaves according to the data's computing requirements and attributes, making it very compatible with the varied and dispersed nature of fog situations.

In the context of classification tasks in fog computing, the decision tree utilises an impurity measure to assess the efficiency of splits at each node. A pure split occurs when all data objects reaching a node belong to the same class. This means that all data at a node corresponds to a specific processing task or priority level in the context of fog computing. The probability estimate for class E_t at node d, which is determined by the number of data instances M_d and the number of instances belonging to class $E_t \setminus M_d^t$, plays a crucial role in guiding the decision-making process and optimising data routing within the fog network. The decision tree structure enables efficient data processing among scattered FNes as well as the derivation of clear and understandable rules that guide the system's operation. Alternatively, one can acquire these principles by direct learning, which offers a versatile method for handling the intricate and ever-changing settings that are typical of fog computing.

$$\hat{P}(E_t|x,d) \equiv p_d^t = \frac{M_d^t}{M_d} \tag{2.14}$$

$$\mathcal{I}_d' = -\sum_{j=1}^{n} \frac{N_{dj}}{M_d} \sum_{i=1}^{K} p_{dj}^i \log_2 p_{dj}^i \tag{2.15}$$

-

2.10.3 Random Forest (RT)

In the context of fog computing applications, a random forest can function as an effective classifier for the management of complicated datasets that contain several types of services [184]. More specifically, consider a dataset that has nine attributes and seven different categories of service. In the context of this scenario, a random forest is comprised of a collection of tree-structured classifiers denoted as $\{g(t,p), p = 1, \dots\}$, where p denotes independent random vectors that are identically distributed. Every tree in the forest performs its own independent analysis of the input t and then casts a vote to determine which type of service is the most suitable.

Decision trees are particularly useful in fog computing due to their capacity to adapt to the unique characteristics of the data processed at various nodes dispersed across the network. However, decision trees trained on different subsets of data may yield drastically different outcomes, and the technique itself can incur significant computational costs. Moreover, once a decision tree splits, it loses the ability to revisit previous decisions, potentially leading to poor outcomes.

By training numerous decision trees simultaneously and pooling their results of those trees, the random forest technique can successfully address these limitations. "Bagging" (bootstrap aggregating) is a beneficial technique that stabilizes the model's predictions and reduces variance. It is possible that this strategy may prove to be especially helpful in a fog computing environment. It will ensure more reliable categorization throughout the distributed network by leveraging the collective wisdom of numerous decision trees. By utilising this strategy, the fog computing system can handle a wider variety of workloads that are dynamic in nature. The dataset includes seven different classes of service associated with these workloads.

2.10.4 K-Nearest Neighbours (KNN)

The K-Nearest Neighbours (KNN) algorithm is a popular method in supervised learning that classifies items based on their resemblance to other objects in pre-defined categories. The degree of resemblance is commonly measured by computing the distances between data points, with the Euclidean distance being the most used approach. The algorithm categorises an item by considering the categories of its t closest neighbours, where t is a quantity set by the user. The Euclidean distance is a crucial factor in evaluating proximity. It is calculated by detecting the differences in attributes of neighbouring entities and combining them. Equation 2 mathematically represents this strategy, defining the Euclidean distance as the square root of the total of the squared differences between comparable attributes of the entities [204][205].

-

When considering Fog Computing, the utilisation of KNN can be extremely efficient when applied to datasets that require the classification of services based on different attributes. Let's take the example of a Fog Computing application that has 9 properties (features) and produces 7 distinct types of services. The KNN algorithm can be utilised to categorised incoming data items, such as requests for various services, by identifying the category that most closely aligns with the attributes of the request. The program will compute the Euclidean distance between the new data point and all the existing data points in the dataset. It will then assign the service class based on the majority class among the t nearest neighbours. The use of KNN in Fog Computing situations is highly beneficial because to its simplicity and efficacy in enabling quick classification without requiring significant computational resources (Zhang, 2012; He et al., 2018).

In addition, the K-nearest neighbours algorithm's capacity to operate efficiently with little datasets, without requiring a complicated training phase, makes it very suitable for decentralised Fog Computing infrastructures. Within systems that often disseminate data across multiple edge devices, the local implementation of KNN can be utilised to classify services promptly and accurately. This improves the responsiveness and efficiency of the fog layer [206](Deng et al., 2019). The ability to select the distance metric gives KNN the flexibility to adjust to the unique attributes of the dataset, hence enhancing its performance in Fog Computing applications [207] (Duda, Hart, & Stork, 2001; Altman, 1992).

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \tag{2.16}$$

2.10.5 Support Vector Machines (SVM)

The Support Vector Machine (SVM) technique, which was first introduced by Vapnik and Chervonenkis in 1963, originated from the discipline of statistical learning[208]. The method aims to optimise hyperplanes to efficiently divide data points into discrete clusters. Meanwhile Boser, Guyon, and Vapnik in [209] introduced nonlinear classifiers about three decades later by applying the kernel approach to maximum-margin hyperplanes. This breakthrough greatly increased the range of applications for SVM by allowing it to handle more complex data that is not linearly separable. Cortes and Vapnik introduced the soft margin SVM, the current standard form of SVM, in the mid-1990s. This version enables better generalization by allowing a certain number of misclassifications [70, 138].

SVMs gained popularity due to their robustness and ability to effectively solve issues related to classification, regression, and novelty detection. An important advantage of SVMs is that the process of finding the model parameters requires solving a convex optimization problem.

This guarantees that any solution found locally is also the global optimum. This characteristic makes SVMs highly reliable for intricate datasets. A comprehensive understanding of SVMs frequently requires a thorough comprehension of Lagrange multipliers. Therefore, it may be advisable to revisit and familiarise oneself with these fundamental principles. To obtain more information, individuals can consult seminal publications by [210][211][212][213].

Within the realm of fog computing, SVMs have significant utility in the processing and categorisation of data at the network's periphery. An SVM, in a fog computing setting with nine characteristics and seven categories, can categorised incoming data streams in real-time, enabling effective decision-making close to the data origin. Utilizing Support SVMs in fog computing guarantees quick response times and the ability to manage the varied and dispersed characteristics of fog environments, thereby improving the overall effectiveness of the system.

$$y(x) = \max_{k} y_k(X) \tag{2.17}$$

2.10.6 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a machine learning model that emulates the learning mechanism of the human brain[214]. An input layer gathers the data for processing, multiple layers analyse the data, and an output layer generates the result. In ANNs, the hidden layers take intermediate inputs, assign random weights and biases to each input, and calculate multiple weighted sums. Layers with weights and sums transfer these sums until they reach the final layer. The final layer employs an activation function to calculate the output. We send the inaccurate outputs back to the preceding layers (by backpropagation) based on a cost function to adjust the weights until we obtain accurate responses.

The Fog computing applications structure of a multi-layer neural network. In classification tasks, the objective of the neural network is to establish a mapping between a given collection of input data and their corresponding classifications. Our primary objective is to train the neural network to precisely associate each data point x_j with its corresponding label y_j. The input space is characterised by the dimensionality of the raw data, denoted as $y_j \in \mathbb{R}^n$. The output layer has the dimensionality of the intended classification space. The construction of the output layer will be further explained in the following section. Upon initial observation, it becomes evident that there are numerous design enquiries pertaining to neural networks. What is the optimal number of layers to use? What should be the size or magnitude of the layers? What is the optimal design for the output layer? Which type of connections should be used between layers: all-to-all or sparsified? What is the preferred method for mapping between layers: a linear mapping or a nonlinear mapping? Like the tuning choices available for support vector

machines (SVM) and classification trees, neural networks offer a substantial range of design variables that can be adjusted to enhance performance.

ANNs are composed of layers of interconnected nodes (neurons) that process input data to learn complex patterns.

$$a_j = \sigma\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{2.18}$$

-

2.11 Summary

5G necessitates a complete rethinking of mobile network design, particularly the RAN architecture, because to the rapidly growing subscriber population, huge connectivity, massive data, and new expectations for extremely low latency and higher data rates. An evaluation of the current RAN designs for 5G mobile networks, especially CRAN, HCRAN, and FRAN, was undertaken as a response. The goal was to explore the benefits of CRAN and HCRAN while not disregarding the challenges that each network design encounters. Consequently, current CRAN and HCRAN solutions were investigated. For energy efficiency, fronthaul capacity, latency, and other factors, detailed summaries were supplied in tabular style. In the case of FRAN, a different method was followed, with the focus being on resource categories within the design layers and then an analysis of resource management system.

CHAPTER:3 ADAPTIVE RESOURCE CONTROL IN 5G-FRANs USING THE HSMM-BASED MOBILITY AWARE CELL ASSOCIATION TECHNIQUE

3.1 Introduction

The unprecedented rise in internet usage is putting a strain on 5G networks, causing management challenges and resource limitations. In this chapter bandwidth optimisation is discussed by implement an algorithm that predict best condition to perform cell handover. This also help with system level optimisation to keep up with high traffic demand, cellular networks are changing from long established radio access networks (RANs) to next generation networks with the addition of leading-edge technologies like cloud computing, fog computing, the Internet of things (IoT), device-to-device (D2D), and vehicular communications [159]. The administration of cell resources, specifically the optimal utilisation of bandwidth, poses a significant challenge when utilising this extensive and diverse architecture. The data from subscriber behaviour, such as the subscriber movement status, downlink rate demand, and reports of cell load, can be used to inform cell association and bandwidth allocation procedures, thereby optimising the system's capacity. Cell association is a crucial aspect of resource management in 5G networks. As a result, [215]–[219] presents a few user cell association systems. However, the incorporation of subscriber movement data in the cell association approach has not received much attention. Subscriber movement modelling can be used to anticipate the cells that provide the greatest rate by identifying the next desired cells based on subscriber network activities.

The mobility prediction algorithm's ability to effectively forecast the results determines the capabilities of mobility-based network bandwidth management schemes. The literature study reveals various mobility prediction schemes utilised to ascertain the most probable upcoming position the device will be in wireless systems. To increase forecast accuracy, mobility prediction systems based on machine learning (ML) are provided in [220]–[226]. To decrease the small cells transition message duration, a discrete time Markov chain (DTMC)-based transfer prediction approach was created [227]. The subscriber movement history is used by the predictive technique to identify potential transition events for small cells. Also shown was a homogenous discrete Markov chain for controlling 5G subscriber movement patterns. Moreover, the system used an *n-step* scheme to anticipate traffic overload and choose the most direct routes between the active base stations. In contrast, stay time distributed geometrically in DTMCs, where subscriber movement is presumptively memoryless. As a result, the DTMC method just predicts the following cells without considering the time of arrival and dwell duration. A continuous time Markov chain, on the other hand, presupposes that dwell

time has a finite exponential distribution and no memory for forecasting subscriber movement. The subscriber movement behaviours in cellular networks display memory features, which can be evaluated than memoryless exponential distributions, according to the authors of [228]. As a result, a semi-Markov model was provided in [222] and [229] for forecasting the next cell transition as well as the holding time. The models do not, however, consider the impact of ping-pong, incomplete data, and limited context challenges that are typically connected with real edge customer mobility data for network capacity maximization. The fact that data generation and usage are widely dispersed and situated at the network's edges is a significant feature of the evolving RAN design. As a result, there is less latency and a higher quality of user experience because information is closer to the end user devices (QoE). According to published research, cloud radio access networks (CRANs) use the idea of a centralised data centre to handle a wide range of applications and a large amount of data. Large bandwidth and extremely low latency should be provided through the fronthaul links for the centralised baseband unit (BBU) pool. Since edge network connections are not designed to handle the huge data problem, the transmission capacity is typically limited and time-delayed in real-time applications.

Recent research has investigated how edge computing capabilities could be incorporated to modern distributed computing architectures like mobile clouds, D2D, and fog computing to address the CRAN challenges. To solve the challenge of rising traffic needs while improving end-user QoE, the fog radio access network (FRAN) makes use of the CRAN and fog computing. On base stations (BSs) and user devices, the FRAN architecture provides local content handling and cooperative and decentralised radio resource management (UEs). Additionally, because the fog architecture distributes basic network demands by placing processing activities close to network edges, it offers better spectral and energy efficiency as well as low latency. Therefore, by anticipating future cell mobility of users at the FNe, cell association policies for dynamic bandwidth management in 5G-FRANs can be improved.

This work offers a cell association approach that takes subscriber movement in account to dynamically manage bandwidth in 5G-FRANs. The suggested method combines distributions of duration period with information about cell identifications (cell-IDs) to create a hidden semi-Markov model (HsMM) that predicts the next cell that will best meet downlink rate requirements. To comprehend subscriber movement patterns, the proposed HsMM considers ping-pong effects, missing values, and constrained context from wireless networks. The remaining sections of this chapter are structured as follows. Section 3.2 develops the subscriber movement prediction model. Section 3.3 discusses the steady state distribution of users in cells. The process for tracking HsMM mobile users is highlighted in Section 3.4. Section 3.5 introduced the suggested dynamic bandwidth control plan. Section 3.6 presents the simulation findings. Section 3.7 brings the chapter to a close.

-

3.2 User Mobility Prediction Model

A mobile user has both spatial and temporal attributes in a network. The spatial attribute includes cell location of the user while temporal attribute indicates the user's dwell time in a designated cell area. The user's mobility is modelled as a semi-Markov process as opposed to a continuous-time Markov process since the dwell time that occurs when a user is associated to a cell is heavy-tailed distribution as described in [230] rather than being an exponential distribution. The dwell times can be distributed freely in the semi-Markov process, which can be thought of as an embedded Markov chain such that the moments that a user associates with a new cell are the embedded points. Let T_q be the time instant of the qth transition and X_q be the state at the qth transition. The user movement is represented by a Markov renewal process $\{(X_q, T_q)\}$ having discrete state space $D = \{1,2,\ldots,k\}$, where k is the total number of cells and $\mathcal{D} \in \mathcal{D}$. A switch from a cell i to another j signifies a state transition. Suppose that the semi-Markov renewal process is homogeneous in time and the user μ has stayed in the ith cell for a period t. The time-homogeneous semi-Markov kernel, corresponding to the transition probability of μ from ith cell to jth cell is given as [222].

$$\mathcal{A}_{i,j}(t) = \mathcal{P}_r\big(\mathcal{X}_{q+1} = j, \mathcal{T}_{q+1} - \mathcal{T}_q \leq t \big| \mathcal{X}_q = i\big), q \geq 0$$

$$= \rho_{i,j} \Upsilon_{i,j}(t), \tag{3.1}$$

Where $\rho_{i,j}$ indicates the probability that the user μ would switch from ith cell to jth cell, obtained as:

$$\rho_{i,j} = \lim_{t\to\infty} \mathcal{A}_{i,j}(t)$$

$$= \mathcal{P}r\,(\mathcal{X}_{q+1} = j | \mathcal{X}_q = i). \tag{3.2}$$

Thus, the embedded Markov chain has a $k \times k$ transition probability matrix:

$$A = \begin{bmatrix} \rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,k} \\ \rho_{2,1} & \rho_{2,2} & \ldots & \rho_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{k,1} & \rho_{k,2} & \cdots & \rho_{k,k} \end{bmatrix}. \tag{3.3}$$

Note that matrix A is a hollow matrix, that is, $\rho_{\mathrm{i,i}} = 0$ since a user cannot switch from a cell to itself.

Also, $\Upsilon_{i,j}(t)$ denotes the user's dwell time distribution in ith cell when j is the future cell, given by:

$$\Upsilon_{i,j}(t) = P_r(\Gamma_{q+1} - \Gamma_q \leq t | \mathrm{X}_{q+1} = j, \mathrm{X}_q = i). \tag{3.4}$$

Moreover, the dwell time probability distribution of μ in ith cell irrespective of the subsequent cell is given by:

$$\vartheta_i(t) \triangleq P_r(\Gamma_{q+1} - \Gamma_q \leq t | \mathrm{X}_q = i). \tag{3.5}$$

Hence, $\vartheta_i(t)$ for which μ is associated with the ith cell before moving to the next cell is obtained as:

$$\vartheta_i(t) = \sum_{j=1}^{k} \mathcal{A}_{i,j}(t). \tag{3.6}$$

Suppose the time-homogeneous semi-Markov process is $\mathcal{H} = (\mathcal{H}_t, t \in \mathbb{R}_0^+)$, having state transient distribution as [222]:

$$\Psi_{i,j}(t) \triangleq P_r(\mathcal{H}_t = j | \mathcal{H}_0 = i)$$

$$= \big(1 - \vartheta_i(t)\big)\delta_{i,j} + \sum_{r=1}^{k} \int_0^t \Psi_{r,j}(t-\tau)\, d\mathrm{A}_{i,r}(\tau)$$

$$= (1 - \vartheta_i(t))\delta_{i,j} + \textstyle\sum_{r=1}^{k} \int_0^t \frac{d\mathrm{A}_{i,r}(\tau)}{d\tau} \Psi_{r,j}(t-\tau) d\tau, \tag{3.7}$$

Where $\delta_{i,j}$ represent the Kronecker delta function such that $\delta_{i,j}$ equals 1 if I and j are equal, and 0 is otherwise. Given that the process is a discrete-time homogeneous semi-Markov process, (7) can expressed as:

$$\Psi_{i,j}(\hat{t}) = \eta_{i,j}(\hat{t}) + \textstyle\sum_{r=1}^{k} \sum_{\tau-1}^{\hat{t}} \varsigma_{i,r}(\tau) \Psi_{r,j}\,(\hat{t} - \tau), \tag{3.8}$$

Where $\eta_{i,j}(\hat{t})$ and $\varsigma_{i,j}(\hat{t})$ represent $(1 - \vartheta_i(t))\delta_{i,j}$ and $\frac{d\mathrm{A}_{i,r}(\tau)}{d\tau}$ respectively. Since the derivate of $\mathrm{A}_{i,r}(t)$ is derived as histogram distribution from trace data, $\varsigma_{i,r}(\hat{t})$ can be further approximated as:

$$\varsigma_{i,r}(\hat{t}) = \begin{cases} \mathcal{A}_{i,r}(1), & \hat{t} = 1, \\ \mathcal{A}_{i,r}(\hat{t}) - \mathcal{A}_{i,r}(\hat{t} - 1), & \hat{t} > 1. \end{cases} \tag{3.9}$$

Note that $\Psi_{i,j}(\hat{t})$ is real square matrix, having each row add up to 1.

An estimation of the parameters of the semi-Markov model based on user states observation is necessary to keep record of the user's movement. Suppose that 0_t is the observed user-ID in a specific location at time t, the $k\ x\ c$ observation probability matrix is given as:

$$\mathfrak{P} = \begin{bmatrix} \mathfrak{b}_{1,1} & \mathfrak{b}_{1,2} & \cdots & \mathfrak{b}_{1,c} \\ \mathfrak{b}_{2,1} & \mathfrak{b}_{2,2} & \cdots & \mathfrak{b}_{2,c} \\ \vdots & \vdots & \vdots & \vdots \\ \mathfrak{b}_{k,1} & \mathfrak{b}_{k,2} & \cdots & \mathfrak{b}_{k,c} \end{bmatrix}, k \in \mathcal{D} \tag{3.10}$$

Where $\mathfrak{b}_{k,c}$ is the conditional probability that the observed cell-ID at time t is $0_t = c$, given that the state of the user is $\mathcal{X}_t = k$.

3.3 Steady-State Distribution of Users In Cells

The long-term average distribution of users across cells can be determined by computing the steady-state distribution $\xi = [\xi_i], 1 \leq i \leq k,$ which represents the probability distribution of users in cells at a given instance in time.

$$\xi_i = \frac{\hat{\vartheta}_i \varpi_i}{\sum_{j=1}^{k} \hat{\vartheta}_j \varpi_j}, \tag{3.11}$$

where $\hat{\vartheta}_i$ and ϖ_i are the mean dwell time and the steady-state transition probability of a user in cell i respectively. Note that $\varpi = [\varpi_1, \varpi_2, \ldots, \varpi_k]$ is obtained by solving.

$$\varpi \mathrm{A}, \ \sum_{i=1}^{k} \varpi_i = 1. \tag{3.12}$$

The Markov state process is characterized by the initial state probability vector c, state transition probability matrix $\mathcal{A}$, observation probability matrix $\mathcal{B}$, and state duration probability matrix $\Theta = [\vartheta_k(t) \ni k \in \mathcal{D}]$. This implies that the HsMM mobility prediction model can be represented by $\lambda = (\pi, \mathrm{A}, \widehat{\mathcal{B}}, \widehat{\Theta})$. The HsMM becomes an HMM when $\vartheta_k(t)$ is given as a geometric distribution.

3.4 HsMM Mobile User Tracking Process

The forward-backward and re-estimation schemes [231] are adopted for tracking the user's state. The major steps of the HsMM training process are summarized as follows.

1. Obtain initial HsMM parameters $\hat{\lambda}$. Initial estimates $\hat{\lambda} = (\hat{\pi}, \widehat{\mathrm{A}}, \widehat{\mathrm{B}}, \widehat{\Theta})$ of the HsMM parameters are obtained from the observations 0_t based on HsMM re-estimation algorithm.
2. Obtain current user state $\hat{\mathrm{X}}_t$ at time tt. From the measured user-ID observation sequence O_I^I, obtain the current state of the user at time t using the forward-backward algorithm.
3. Obtain refined estimates $\tilde{\lambda}$. The initial HsMM parameter estimates $\tilde{\lambda}$ are refined using the Baum-Welch based HsMM re-estimation algorithm on O_t^I to obtain $\tilde{\lambda} = (\tilde{\pi}, \widetilde{\mathrm{A}}, \widetilde{\mathrm{B}}, \widetilde{\Theta})$.

3.5 Proposed Dynamic Bandwidth Management

The HsMM-based prediction scheme is used by the fog server to create a mobility pattern by tracking the user's movement across cells. The ability to forecast future mobility enables the fog server to dynamically manage the bandwidths of accessible cells so that it may predict the next user cell association to a specific cell at a particular time instance for the subsequent time interval. Given a next cell probability vector and received signal strength (RSS) indicator, the fog server estimates optimal cell of a user according to the downlink rate demands. Like [232], the load of user μ on a cell $\kappa \ \varepsilon \ \mathcal{D}$ is obtained as

$$\mathcal{L}_{\kappa u} = \frac{R_U}{R_{KU}} \tag{3.13}$$

where R_U and R_{KU} are the realistic and maximum achievable rates respectively. Also, the cell load is obtained as

$$\mathcal{L}_K(t) = \sum_{u \in U} \mathcal{T}_{KU}\, \mathcal{L}_{KU}, \tag{3.14}$$

where $\mathcal{T}_{KU}$ is the indicator of the user-cell connectivity and U is the entire users in the network. This implies that if u and $\mathcal{K}$ are connected, $\mathcal{T}_{KU} = 1$ and $\mathcal{T}_{KU} = 0$ if they are not. The cell

load at a given time instance t signifies the number of active users at each cell. Hence, the load of the system is given as

$$\mathcal{L}_s(t) = [\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k]. \tag{3.15}$$

Generally, users evenly share the downlink rate from a specific cell among each other. Thus, for each $\mathcal{L}_K(t) \in \mathcal{L}_S(t)$, the effective downlink rate $\mathcal{E}_{KU}$ for each user is given as

$$\mathcal{E}_{KU}(t) = \frac{R_{KU}}{\mathcal{L}_K(t)}. \tag{3.16}$$

However, the process of comparing cell association condition and user's requirement may entail examining every probable cell in the system, which may not be realistic for a large $\mathcal{D}$. Consequently, it is important to derive an optimal solution that determines the maximum downlink rate offering next cell for users' association according to the rate requirements. The downlink rate and next cell mobility-based optimal next cell prediction is characterized as a convex problem, which is solved using the Lagrangian dual decomposition approach to obtain a $k \times \cup$ user-cell connectivity matrix at time $(t + \acute{t})$.

$$\wedge\,(t + \acute{t}) = \begin{bmatrix} \mathcal{T}_{1,1} & \mathcal{T}_{1,2} & \cdots & \mathcal{T}_{1,U} \\ \mathcal{T}_{2,1} & \mathcal{T}_{2,2} & \cdots & \mathcal{T}_{2,U} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{T}_{k,1} & \mathcal{T}_{k,2} & \cdots & \mathcal{T}_{k,U} \end{bmatrix}, \in \{0,1\}. \tag{3.17}$$

The procedures of the HsMM-based mobility aware cell association method is outlined in Algorithm 1.

Algorithm 1: HsMM-based mobility aware cell associated procedure

Procedure CELLASSOCIATION(A, N, k, rq, R_{lm}**)**

Input: Dataset A, number of users N, total number of cells *k*, rate requirement r_q, available resources R_{lm}.

Output: Optimal cell k_{max}.

Data Pre-processing:

Transform cell-ID to numeric integers: $\bar{A} = A$;

Represent $\bar{A}$ as sequential cell-IDs, $\acute{A} = \bar{A}$;

Eliminate ping-pong effects $\check{A} = \acute{A}$;

HsMM Training:

repeat

  perform HsMM training process as described in section IV;

until $u_i = end$;

The next cell probability matrix $\mathcal{M}$

Cell Association Matching

Compute optimal cell for each user u_i

repeat

  update Λ;

  repeat

    update predicted next cell from $\mathcal{M}$;

    set r_q

    obtain new cell k

    update $\mathcal{T}_{\mathcal{KU}}$

  until $\mathcal{R}_{lm} > r_q \; k \; \epsilon \; \mathcal{D}$;

until $u_i = \text{end}$;

Obtained optimal cell K_{max} from updated Λ

End procedure

3.6 Experimental and Simulation Results

The two main properties of the semi-Markov process; transition probability matrix A and dwell time distribution ϑ_i are used to represent user mobility in the 5G-FRAN system. Experimentally, A and ϑ_i have been determined using the user movement trace dataset from Crawdad Laboratory [233]. The recording was obtained from 10 participants at Rice Community, Houston, Texas, USA, over the period of 3 to 6 weeks. The datasets include a timestamp in hours, minutes, and seconds, the cell identification number (cell-ID), received signal strengths, and other information. The timestamp and related cell-IDs were chosen from the trace dataset. The data for each user is divided to 70% for training and 30% for testing to determine the performance of the user mobility model.

3.6.1 Data Preprocessing

The raw data is pre-processed to resolve missing and redundant values in the dataset before sending the data into the HsMM. Figure 3.1 shows the stepwise data pre-processing stages of the user mobility trace dataset. The cell-IDs are transformed to numeric integers in the first step, while the numeric cell-IDs are further represented as sequential cell-IDs in the sec step. This is to ensure that the datasets are appropriately labelled and suitable for next cell prediction using HsMM. Each sequential cell-ID in the table represents a class. Thereafter, the dwell time for a user at each cell is determined by the date and time, comprising all the user's mobility data for predicting the subsequent cell.

It was observed in the data that the user's equipment occasionally suffered ping pong effects at certain areas. This suggests that users frequently re-associate to a different cell in a brief period. The ping pong transitions are eliminated from the user's initial cell association se-

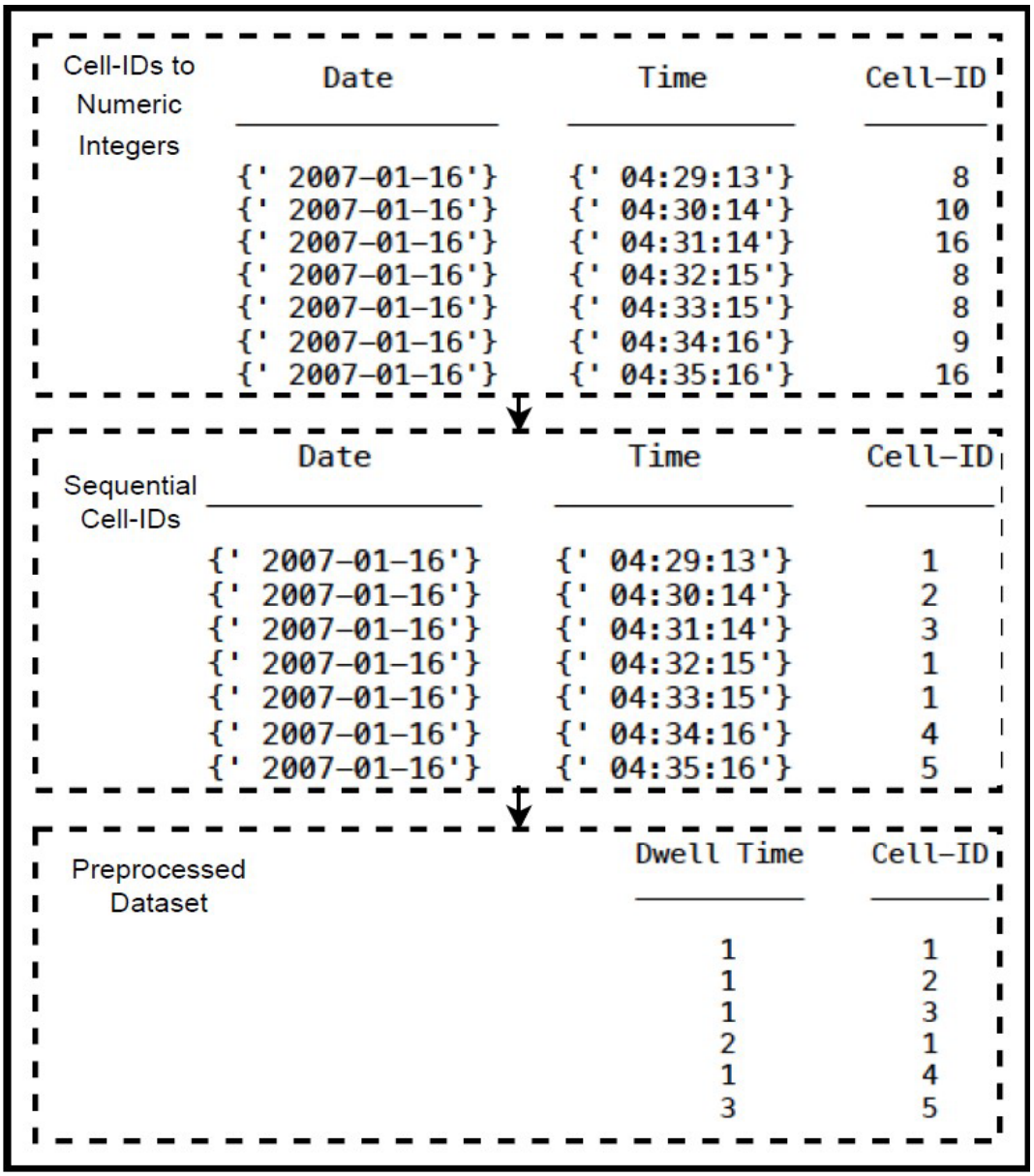

Figure 3.1: Feature associate estimate

quence by locating each transition and creating a cluster state that includes all the discovered ping-pong transitions. The dominating cell that the user has always connected among the cells in the cluster state then replaces each discovered ping-pong transition. Hence, only well re-fined datasets are used to develop the HsMM.

3.6.2 HsMM Prediction Performance

A prediction of the mobility pattern is performed using the HsMM up to the next 10. 800 sec period and compared with the actual mobility pattern to derive the number of correct α_c and

incorrect α_i next cell predictions. Hence, the HsMM prediction accuracy is obtained as a ratio of U2 and U8, given by

$$A_{cc} = \frac{\alpha_c}{\alpha_c + \alpha_i}. \tag{3.18}$$

Figure 3.2 shows the next cell prediction accuracies for 10 users. The observation time interval w is varied at every 900 sec in the range of $900 \leq w \leq 3600$ sec to determine the optimal observation sequence length. The result shows that the minimum and maximum prediction accuracy varies from A_{CC}= 55.7% to A_{CC} = 60.1% and A_{CC} = 99.5% to A_{CC} = 100% respectively. For example, user 3 yields A_{CC} = 100% at w = 1800 sec while the minimum accuracy of the same user is 98.1% at w = 900 sec. Moreover, considering the average accuracy of all users at each w, the result further indicates that utilising an observation time interval of w = 2700 sec exhibits the best accuracy of A_{CC} = 85.77% in the HsMM-based user mobility prediction model.

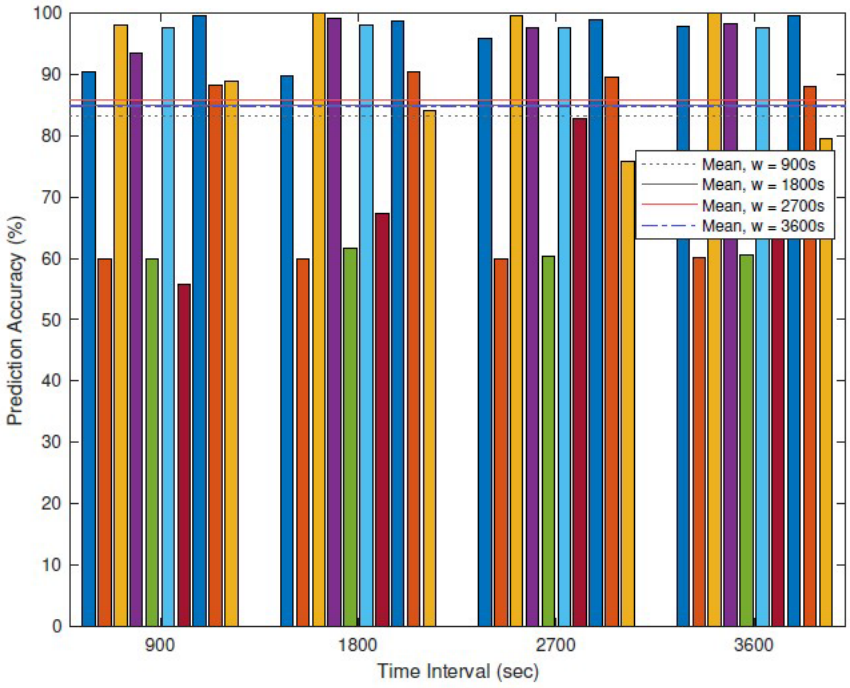

Figure 3.2: Next cell prediction accuracies for w =900s, 1800s, 2700s, 3600s

3.6.3 User satisfaction performance analysis

The proposed HsMM-based mobility aware cell association scheme enables the system to dynamically locate the best cell that satisfies downlink rate requirement of users by utilizing the next cell information with the intended rate requirements. Unlike assigning a user to a congested cell, the proposed scheme detects congestion proactively using the information on next cell mobility and rate requirements. This implies that at time t the next cell with the highest

offering rate for time $t + \acute{t}$ is predicted as described in Algorithm 1. The proportion of satisfied users in the network is analysed using a user satisfaction percentage, given by

$$F = \left(\sum_{u=1}^{U} \left(\frac{R_d}{R_a}\right) \div U\right) \times 100, \tag{3.19}$$

where R_d is the desirable downlink rate and R_a is the user's assigned rate.

Figure 3.3 indicates the percentage of user satisfaction using the proposed HsMM, reference signal received power (RSRP), and long short-term memory (LSTM) based cell association schemes. The RSRP approach does not require the mobility prediction information of users, while the LSTM functions like HsMM. The results are based on using an observation time

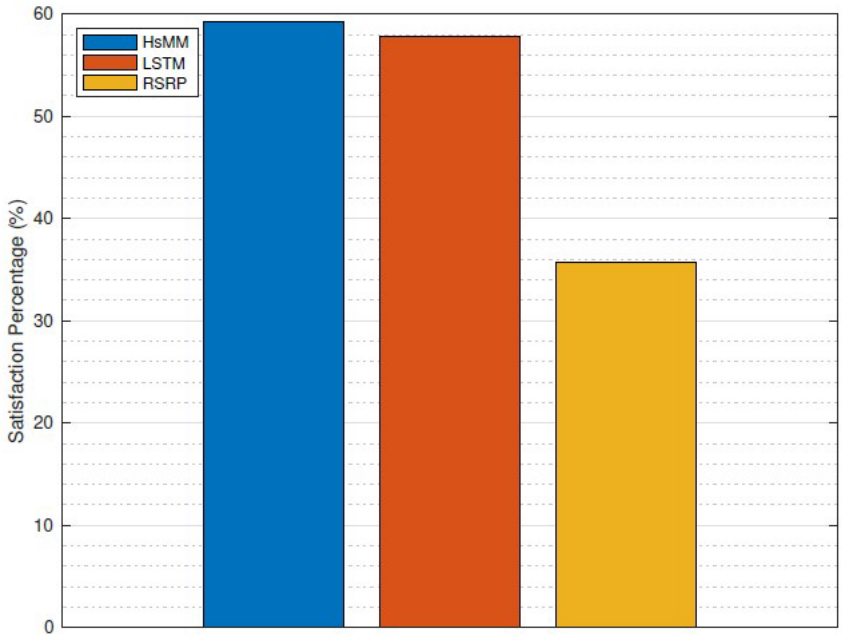

Figure 3.3: User satisfaction according to downlink rate allocation

interval w = 2700 secs since it produces the optimal next cell prediction accuracy. The HsMM-based mobility aware cell connectivity approach yields a user satisfaction of 59.2% with the offered rate at new cells compared to the LSTM scheme which exhibits 57.8% user satisfaction and RSRP producing 35.7% user satisfaction based on their request.

3.7 Summary

This chapter presents a cell association technique that focuses on subscriber movement to manage bandwidth in 5G fog radio access networks (5G-FRANs) with anticipation. The proposed approach integrates probability distributions of time intervals with cell identifications (cell IDs) to construct a hidden semi-Markov model (HsMM) that predicts the most suitable next cell for meeting downstream rate requirements. Furthermore, the optimal cell that achieves the highest rate is determined based on the chosen optimization strategy. The suggested Hidden semi-Markov Model (HsMM) takes into consideration the ping-pong effects,

absent values, and restrictive environment from wireless networks to understand user traffic situations. The results indicate that the HsMM-based method achieves a high level of accuracy with respect to the number of observation time intervals. Furthermore, in comparison to the existing system, the suggested approach leads to a higher level of user satisfaction.

CHAPTER:4 EIGEN DECOMPOSITION BASED FEATURE EXTRACTION METHODS FOR 5G FRAN

4.1 Introduction

Feature extraction (FE) [234] is crucial in fog computing applications (FCAs) within 5G FRAN because of the distinctive attributes of dispersed computing environments and the real-time requirements of 5G services. Various IoT devices and sensors produce substantial quantities of data at the network edge in 5G FRAN. Directly processing this raw data is resource-intensive and inefficient, especially in latency-sensitive applications such as autonomous driving, smart healthcare, or real-time video analytics. Hence, the use of FE mitigates these challenges by converting raw data into a collection of essential, representative characteristics, therefore substantially decreasing the data's dimensionality. This improves computing performance and reduces communication overhead between edge devices and the cloud or central servers. Furthermore, collecting relevant characteristics at the fog layer facilitates expedited decision-making near the data source, improving response times and reducing latency, which is essential for adhering to the rigorous service-level agreements (SLAs) in the 5G network. 5G FRAN incorporating FE may lessen the effects of noisy, redundant, or unnecessary data by focussing on the important features. This could lead to more accurate and reliable machine learning (ML) models used in the fog environment. FE enhances resource efficiency while guaranteeing real-time, scalable, and intelligent service provision in 5G FRAN.

As, this chapter focusses on implementing eigen decomposition-based FE (ED-FE) methods, which are suggested to enhance the effectiveness of ML algorithms for the detection and classification. From literature, the two ED based FE methods to be discussed in this section principal component analysis (PCA) and linear discriminant analysis (LDA). Both methods will modified to extract important information from the 5G FRAN dataset. The are five, ℓ dataset samples sizes, τ, and along with incremental addition of noise, φ. The dataset, A is a matrix, $m \times n$, which get to be amended with the addition of noise, φ, and the ℓ-dimension are changed by samples sizes, τ. The changes of the dataset , A, size move from testing the complete dataset to testing only two percent of data. This assist in testing the ML models robustness, generalization and overall evaluation. As a result, the study's primary contributions are the identification of applications and the classification of service categories in order of priority.

Dataset structure:

A dataset of m samples and n features.

The dataset is represented by a matrix A of size $m \times n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad (4.1)$$

Where each row $a_i = [a_{i1} \quad a_{i2}, \quad \cdots \quad , a_{in}]$ represents a single sample, and each column a_j represents a feature across all samples.

Feature Vector Notation:

Each sample a_i can be represented as an n-dimensional vector:

$$a_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix} \in \mathbb{R}^n \quad (4.2)$$

Target Vector:

Let y represent the vector of labels for all samples, where:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad (4.3)$$

Where y_i is the label corresponding to the sample x_i. For classification, y_i that is categorical CoS.

Data Size Adjustment

$$\tau = \in \mathbb{R}^{m \times n}, y' \in \mathbb{R}',$$

4.2 Eigen decomposition

Given a dataset matrix, A, ED is the process of decomposing A to eigenvectors B

$$A = \begin{bmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_m \\ | & | & \cdots & | \end{bmatrix},$$

to eigenvectors B,

$$B = \begin{bmatrix} | & | & \cdots & | \\ b_1 & b_2 & \cdots & b_m \\ | & | & \cdots & | \end{bmatrix}, \tag{4.4}$$

and corresponding eigenvalues, C,

$$C = \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & C_m \end{bmatrix}. \tag{4.5}$$

Each column in Equation (4.5) indicates an eigenvector in A and Eq. (4.5) is a diagonal matrix whose diagonal elements give the corresponding eigenvalues of the eigenvectors. The above equations can be expressed as:

$$AB = BC,$$

$$A = BCB^{-1}. \tag{4.6}$$

The decomposition of A in terms of B and its corresponding C provides useful information about the properties of the matrix. This simplifies the analysis of complex problems and thus, enhances the understanding of the behaviour of the system. ED is significant in ML because it is involved in optimization problems and has a wide range of applications such as spectral analysis, quantum mechanics, structural engineering, stability analysis, image compression, probability theory, and stochastic processes ([235].

4.3 Singular Value Decomposition

The application of singular value decomposition (SVD) is a key component in several ED schemes because of its effectiveness for matrix factorisation. SVD decomposes a matrix A of size $m \times n$ into three separate matrices: $A = U\mathrm{K}V^T$. In this context, U is an $m \times n$ orthogonal matrix with columns referred to as the left singular vectors, K is a $n \times n$ diagonal matrix that contains the singular values, and V is an $n \times n$ orthogonal matrix whose columns denote the right singular vectors [236]. The diagonal members of M stand for the singular values, which are non-negative and usually arranged in descending order. This shows how much variation there is in each dimension of the matrix. Hence, SVD is a good tool for reducing the number of dimensions since keeping only the largest singular values is usually enough to copy the original data with little information loss [237].

The significance of SVD transcends dimensionality reduction; it is also crucial for noise reduction and image compression. By eliminating minor singular values in K, SVD efficiently mitigates noise, thereby improving the quality of data representation [238]. In image processing, SVD enables compression by approximating image pixels with just the most

significant singular values and their related singular vectors, thereby minimising storage needs while maintaining visual integrity. Additionally, recommender engines employ low-rank matrix approximations derived from SVD to forecast user preferences by representing user item interactions with less complexity [239]. The varied uses highlight the adaptability and importance of SVD in computational methods and data analysis.

In defining SVD, $A \in \mathbb{R}^{mxn}$ is decomposed as [240]:

$$A = UKV^T, \tag{4.7}$$

U and V are unitary matrices with orthonormal columns in $\mathbb{R}^{mxn}$ and $\mathbb{R}^{nxn}$, respectively, where T denotes the transpose. A diagonal matrix with zeros off the diagonal and non-negative elements on the diagonal is denoted by the symbol $K \in \mathbb{R}^{nxn}$. The left singular vectors of A, which correspond to the columns of K, carry information about the column space of A. Because the vectors are arranged hierarchically, U_1 is prioritised over U_2, and so forth. The right singular vectors of A, which represent the information about $A's$ row space, are represented by the columns of V. $K \in \mathbb{C}^{mxn's}$ diagonal elements are singular values that represent the relative importance of U and V^T. These components are arranged in decreasing order, such $k_1 \geq k_2 \geq \cdots \geq k \geq 0$. Consequently, SVD in Equation (4.7) represent the factorisation of A into several intrinsic components, each of which has a distinct meaning on the application.

The number of non-zero items in K is limited to n when $m \geq n$. The additional rows of zeros in K are eliminated, leaving just the first m singular values. The term economic SVD [241] is used in the literature to describe this notion. The singular values and related singular vectors required to precisely represent A are the sole numbers that the economic SVD computes. The economy SVD of A is therefore described as follows:

$$A = \underbrace{\begin{bmatrix} | & | & \cdots & | \\ U_1 & U_2 & \cdots & U_m \\ | & | & \cdots & | \end{bmatrix}}_{U^{mxn}} \underbrace{\begin{bmatrix} K_1 & 0 & \ldots & 0 \\ 0 & K_2 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathrm{K_n} \end{bmatrix}}_{K^{nxn}} \underbrace{\begin{bmatrix} - & - & - & V_1^T & - & - & - \\ - & - & - & V_2^T & - & - & - \\ & \vdots & & \vdots & & \vdots & \\ - & - & - & V_n^T & - & - & - \end{bmatrix}}_{V^{nxn}}. \tag{4.8}$$

The fact that SVD offers a hierarchy of low-rank menu to useful for reducing dimensionality while maintaining the correctness of the decomposition, consequently, while enhancing execution time and storage needs.

4.4 Feature Extraction Methods

In the literature FE techniques such as PCA [242][243], LDA [244], and Independent Component Analysis (ICA) [245] have been considered to mitigate multicollinearity by creating new, uncorrelated features. The following section details adapting the PCA with ML for the classification of fog applications*.* The following subsections details the process of the PCA and LDA as possible FE methods to adopted with ML for 5G FRAN applications.

4.4.1 Principal Component Analysis

PCA continues to be an effective tool for identifying intrinsic features from datasets [226][184]. PCA, conceived by Karl Pearson in 1901 [246], is a longstanding and well-established approach, supported by several implementation theories. The PCA technique adheres to a statistical interpretation of the SVD. The method offers a hierarchical coordinate framework to depict the statistical variance within the dataset.

PCA reduces high-dimensional data, A, into a lower-dimensional space that encapsulates the most significant statistical variances of the original dataset. The dimensions are arranged hierarchically, from the most significant to the least significant statistical variance in the datasets. The PCA method executes an orthogonal transformation to turn a collection of correlated variables into a new set of linearly uncorrelated variables known as principal components (PCs). The scheme guarantees that the first PC possesses the highest variance in the dataset, the second PC has the subsequent highest variance, and so on. PCA often captures the heterogeneity in the dataset within the initial PCs.

Given a dataset, A, whose observations are arranged in column-vector format as:

$$A \in \mathbb{R}^{m\times n} = \begin{bmatrix} | & | & \cdots & | \\ A_1 & A_2 & \cdots & A_n \\ | & | & \ldots & | \end{bmatrix}, \tag{4.9}$$

Step 1: Data Normalisation. The normalization process ensures that each variable contributes equally to the computation and assists in the characterization of PCs along the axes of highest variation. By putting variables in the middle of the distribution, mean centering reduces bias and keep variables with large mean values from dominating the analysis, which leads to a fairer presentation of the data.

This phase is critical because of the PCA's sensitivity to the mean values of the initial variables. It is not unusual for certain variables to possess greater values than others, which may introduce bias into the study. We effectively reduce the impact of variables with elevated mean values by individually reducing the mean from each column. Therefore, we conduct data

normalisation to ensure that the calculated PCs accurately reflect the intrinsic variation in the dataset, not just the variables with large magnitudes.

Normalisation is achieved by first calculating the mean, $\bar{\mathrm{a}}$, of Equation (4.9) as specified by:

$$\bar{\mathrm{a}} = \frac{1}{n}\left(\sum_{i=1}^{n} a_{ij}\right), \tag{4.10}$$

where a_{ij} represent the elements at the i-th row and i-th column. Thereafter, a mean matrix, $\bar{A}$, is computed by multiplying an all-ones column vector of size n with $\bar{\mathrm{a}}$ as:

$$\overline{\mathrm{A}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathrm{a}}. \tag{4.11}$$

Subsequently, the mean-centred (normalization) data is obtained by subtracting $\overline{\mathrm{A}}$, from A as:

$$\hat{X} = \mathrm{A} - \overline{\mathrm{A}}, \tag{4.12}$$

Here all variables in $\hat{X}$ is given in the standardised uniform scale.

`**Step 2:** PCs computation using SVD helps to improve numerical stability and efficiency of an model. Essentially, SVD saves time by removing the need for additional sorting necessary to understand the hierarchical order of PCs. Consequently, the matrices of the PCs and their associated coefficients are derived by executing the SVD on $\hat{X}$ as:

$$\hat{X} \in \mathbb{R}^{m \times n} = \mathrm{UKV}^{\mathrm{T}}, \tag{4.13}$$

As shown in Equation (4.13), the columns of the right singular matrix, V, show the PCs, which are the directions of the dataset's axes with the biggest differences. The columns of the singular values, K, show the coefficients that correspond each PC and show how much each PC varies.

We arrange the coefficients in descending order. Equation (4.13) effectively sets the number of PCs and coefficients to equal the number of original variables in A. Consequently, for an n-dimensional dataset, there are n-eigenvectors and n-eigenvalues. Representing the PCs as:

$$\tilde{P}_c = \begin{bmatrix} \tilde{P}_{C1,1} & \tilde{P}_{c1,2} & \cdots & \tilde{P}_{c1,n} \\ \tilde{P}_{C2,1} & \tilde{P}_{C2,2} & \cdots & \tilde{P}_{C2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{P}_{cn,1} & \tilde{P}_{n,2} & \cdots & \tilde{P}_{n,n} \end{bmatrix}. \tag{4.14}$$

The calculated PCs using PCA facilitate the interpretation of datasets in a reduced-dimensional space while maintaining their informational integrity. Some writers contend that PCA is only a 'dimensionality reduction' instrument for 'big data' prior to analysis, yet, it is a versatile method for discerning structure within high-dimensional datasets [247]. The PCs derived through PCA do not depend on pre-established basis functions; instead, they are contingent upon the intrinsic nature, structure, and attributes of the dataset. This renders PCA an adaptable method for data analysis. Different literature have presented a variation and adaptations of PCA for analysing various data types, tailored to certain objectives [241][248]. The goal of this work is to use the PCs from PCA to build feature vectors for ML. Specifically, these vectors will be used to find and group service classes in fog or edge computing applications.

4.4.2 Proposed Principal Component Feature Vectors for Machine Learning

This work involves using the PCs derived from PCA in the construction of feature vectors for ML. The suggested feature vectors based on PCs are generated by projecting $\hat{X}$ with a chosen number of P_c. Consequently, a certain quantity of P_C, referred to as β, is chosen to map the initial data, $\hat{X}$, into a $n \times \beta$ low-dimensional space labelled as $\widetilde{D}$:

$$\widetilde{D} = \hat{X}\tilde{P}_C,$$

where $\hat{X} \in \mathbb{R}^{n \times m}$ and $\tilde{P}_c \in \mathbb{R}^{m \times \beta}$. As a result, $\widetilde{D} \in \mathbb{R}^{n \times \beta}$ represents low-dimensional data. As a result, the low-dimensional data, $\widetilde{D}$, is expressed as an $n \times \beta$ matrix:

$$\widetilde{D} = \begin{bmatrix} \widetilde{D}_{1,1} & \widetilde{D}_{1,2} & \cdots & \widetilde{D}_{1,\beta} \\ \widetilde{D}_{2,1} & \widetilde{D}_{2,2} & \cdots & \widetilde{D}_{2,\beta} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{D}_{n,1} & \widetilde{D}_{n,2} & \cdots & \widetilde{D}_{n,\beta} \end{bmatrix}. \tag{4.15}$$

Equation (4.15) denotes the original data $\mathrm{A} \in \mathbb{R}^{n \times m}$ transformed into $n \times \beta$ $\widetilde{D} \in \mathbb{R}^{n \times \beta}$. The matrix $\widetilde{D}$ will be utilised to build an appropriate matrix of feature vectors for application with the ML in the detection and classification of FCAs.

The P_c is calculated for each A_τ in Equation (4.4) as detailed from Step 1 to Step 2. A certain quantity of β is chosen to project the calculated $\tilde{P}_c$ as delineated in Equation (4.14). Subsequently, every single feature vector, H_y, is derived by calculating the column-wise average of Equation (4.15) for each X_τ as follows:

$$\sigma_{y,1} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{D}_{i1}$$

$$\sigma_{y,2} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{D}_{i2}$$

$$\vdots$$

$$\sigma_{y,p} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{D}_{i\beta}$$

$$H_\tau = \left[\sigma_{\tau,1}\ \sigma_{\tau,2}\ \ldots\ \sigma_{\tau,\beta}\right], \tag{4.16}$$

where $\tau = 1, 2, \ldots, h$. Consequently, given τ, the feature vectors generated for each H_τ are reorganised to create the feature matrix, H, for the machine learning model as specified by:

$$H = \begin{bmatrix} \sigma_{1,1} & \sigma_{1.2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_{2,2} & \ldots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{h,1} & \sigma_{h,2} & \ldots & \sigma_{h,p} \end{bmatrix}. \tag{4.17}$$

Each H denotes the feature vectors of a segment of the sampled datasets, τ, which may consist of. The number of PCs chosen for $\widetilde{D}$ directly influences the ML's computational burden and efficacy. Therefore, we will simulate multiple values of p for each φ in the experiment to determine the optimal p value. Algorithm 2 provides a summary of the PCA-FE procedure as outlined in this study.

Algorithm 2: Feature Extraction with Principal Component Analysis

---

1. Input: $[a, \tau, \ell]$
2. Output: H
3. Decide the portion of a to be taken from A by
4. Transform τ to h numbers of A with respect to φ and ℓ
5. Calculate the $\tilde{P}_c$ from Step 1 to Step 2
6. Calculate $\widetilde{D}$ from Equation (4.15)
7. Compute H_τ from $\widetilde{D}$ as described in Equation (4.16)
8. Do horizontal concatenation for each H_τ to obtain H for respective τ as described in Equation (4.17)

4.4.3 Numerical Example of PC Features Vectors

Given that a portion of the FCAs dataset is to be analysed, and its data matrix is represented as:

$$\mathrm{A}=\begin{bmatrix} -0.3505 & 0.2236 & -0.4082 & -0.6464 & -0.6007 & 2.7884 & 0.7437 & -0.6982 & 2.2917 \\ 0.0606 & 0.2236 & -0.4082 & -0.6463 & -0.9589 & -0.5467 & -1.2888 & -0.6985 & -0.4329 \\ 2.5302 & 0.2236 & -0.4082 & -0.6464 & -1.0418 & -0.5304 & -1.2527 & -0.6984 & -0.3620 \\ -0.3404 & 0.2236 & -0.4082 & 0.2050 & 2.2339 & -0.4346 & -0.5921 & 1.9506 & -0.0039 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -0.3375 & 0.2236 & 1.0206 & 2.3315 & 0.8320 & -0.6433 & -1.0075 & 1.8154 & -0.4350 \\ 2.5274 & 0.2236 & -0.4082 & -0.6462 & -0.9721 & -0.5763 & -1.5141 & -0.6986 & -0.1260 \\ -0.3486 & -1.3416 & -1.8371 & 0.9759 & 0.0017 & -0.61.57 & 0.1900 & -0.6523 & -0.4349 \end{bmatrix}. \quad (4.18)$$

The method for calculating the PCs is as follows. Equation (4.18) is normalised utilising Equations (4.10) - (4.12) to obtain the mean centred data $\hat{X}$ as:

$$\hat{X} = \begin{bmatrix} 0.0009 & 0.0003 & \dots & -0.0022 \\ 0.0023 & -0.0067 & \dots & -0.0028 \\ \vdots & \vdots & \ddots & \vdots \\ -0.0023 & -0.0000 & \dots & 0.0047 \end{bmatrix}. \quad (4.19)$$

As previously mentioned, SVD offers a more numerically stable and efficient method for calculating of the PCs. Consequently, the SVD procedure is executed on $\hat{X}$ to get the PCs and their associated coefficients utilising the economy SVD as:

$$\mathrm{A} = \mathrm{UKV^T},$$

where,

$$A \in \mathbb{R}^{127400\times 9} = \begin{bmatrix} 0.0009 & 0.0003 & -0.0036 & -0.0042 & -0.0011 & 0.0016 & -0.0018 & -0.0018 & -0.0022 \\ 0.0023 & -0.0067 & 0.0047 & -0.0003 & 0.0007 & -0.0003 & -0.0014 & 0.0000 & -0.0028 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.0022 & -0.0052 & 0.0038 & -0.0008 & 0.0006 & -0.0008 & -0.0002 & 0.0002 & 0.0017 \\ -0.0023 & -0.0000 & 0.0015 & -0.0045 & 0.0041 & 0.0025 & 0.0010 & 0.0006 & 0.0047 \end{bmatrix} \quad (4.20)$$

comprises left singular vectors,

$$K \in R^{9\times 9} = \begin{bmatrix} 601.996 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 494.683 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 419.546 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 384.558 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 251.743 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 211.996 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 204.404 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 192.835 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 167.525 \end{bmatrix} \quad (4.21)$$

is the singular values indicating the corresponding coefficients of the PC in descending order and,

$$\boldsymbol{V^T} \in \boldsymbol{R^{9\times 9}} = \begin{bmatrix} 0.1610 & 0.3121 & 0.2971 & -0.4865 & -0.4975 & 0.1682 & 0.0194 & -0.4804 & 0.2116 \\ -0.5253 & -0.0004 & -0.0835 & 0.0612 & 0.0748 & \boxed{\mathbf{0.4912}} & \boxed{\mathbf{0.5301}} & 0.0156 & 0.4304 \\ \boxed{\mathbf{0.4477}} & -0.0388 & -0.5941 & 0.0991 & -0.0220 & 0.3281 & -0.3370 & -0.0154 & 0.4619 \\ -0.06557 & \boxed{\mathbf{0.7022}} & 0.2843 & 0.1804 & 0.2127 & 0.3484 & -0.3659 & \boxed{\mathbf{0.2945}} & -0.0446 \\ 0.0807 & 0.3114 & -0.3438 & \boxed{\mathbf{0.4859}} & -0.3491 & 0.1220 & 0.3086 & -0.2726 & -0.4857 \\ -0.1165 & -0.2955 & \boxed{\mathbf{0.2849}} & 0.3727 & 0.2854 & 0.2088 & -0.3167 & -0.6734 & -0.0136 \\ -0.1153 & 0.3116 & 0.0272 & 0.4030 & -0.0790 & -0.6400 & 0.0120 & -0.1308 & \boxed{\mathbf{0.5416}} \\ 0.0900 & 0.3503 & -0.3033 & -0.3506 & \boxed{\mathbf{06717}} & -0.1577 & 0.2130 & -0.3543 & -0.0849 \\ -0.6723 & 0.0626 & -0.4229 & -0.2383 & -0.2009 & -0.0920 & -0.4798 & -0.1074 & -0.1339 \end{bmatrix} \quad (4.22)$$

is the right singular vectors representing matrix $\tilde{P}_c$ of Equation (4.14). In other words, the column of Equation (4.22) represents the PCs. Note that the PCs are hierarchically arranged according to the ordered structure of Equation (4.21), that is, the largest coefficient in Equation (4.21) (601.996) indicates the position of the first PC (column one in Equation (4.23)). Figure 4.1 provides insights into the amount of information captured by each PC and the cumulative information captured by a combination of PCs.

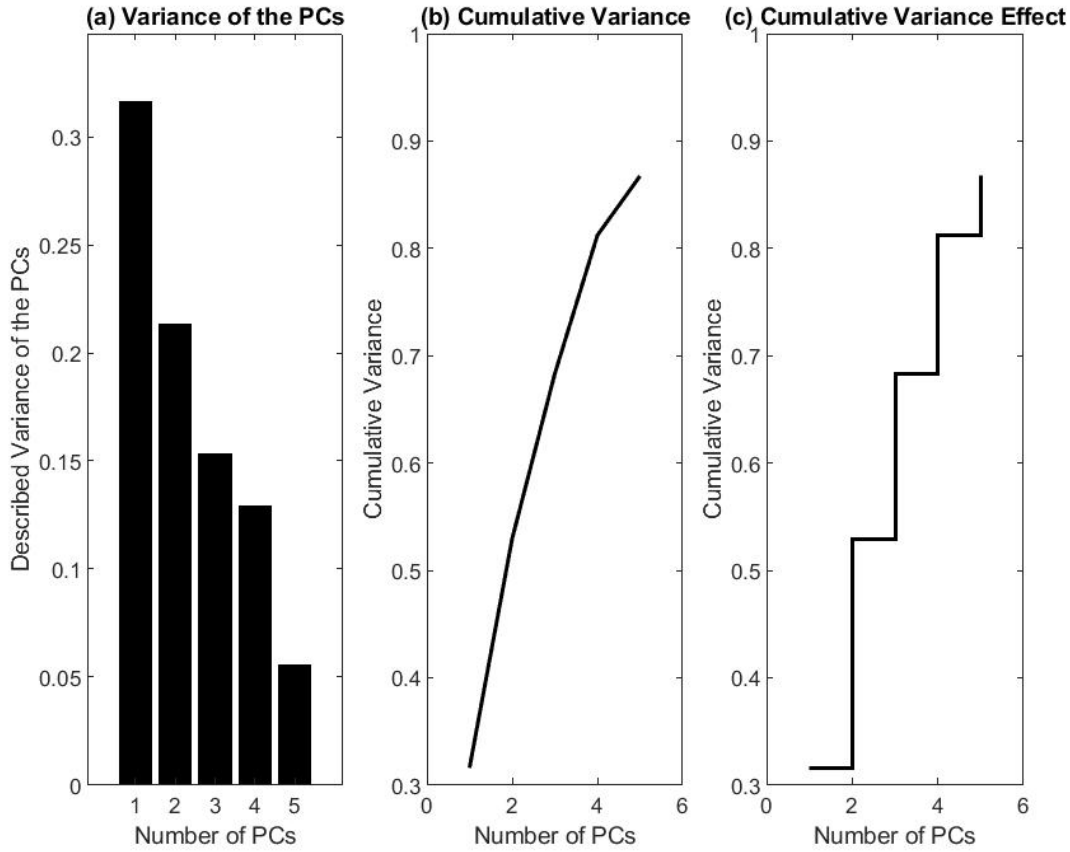

Figure 4.1: (a) Proportion of information captured by each PC, (b) cumulative variance described by a combination of PCs, and (c) incremental contribution of additional PC to the cumulative variance.

Figure 4.1 (a) shows the proportion of the described variance by each PC. The bar plot shows that the first PC accounts for 31.61% of the information in the original data, the second PC accounts for 21.34%, the third PC accounts for 15.35% and so on. It is evident that the first few PCs capture most of the variance, while the later PCs contribute less to the overall information. For example, while the first and second PCs account for 31.61% and 21.34% of the information, the fourth and fifth PCs account for only 12.92% and 5.52% respectively. Consequently, the dimension of the original data can be reduced while retaining the most important information since the top PCs account for most of the variance in the data. Hence, the reduction in dimension simplifies the data presentation, making subsequent analysis and

modelling more efficient. The bar plot expounds on the importance of each PC in capturing the variance in the data.

The cumulative variance plot in Figure 4.1(b) shows the accrued ratio of variance described by a combination of PCs. The cumulative sum represents the proportion of the total variance described by the PC. The percentage of the cumulatively described variance is computed by dividing the cumulative sum by the sum of all eigenvalues. This information is important as it gives guidelines on the optimal number of PCs to be retained for analysis.

Furthermore, the stair plot in Figure 4.1(c) gives more illustration on the cumulative variance. It explains how much of the total variance is described by each addition of a PC. Each step of the stair plot indicates a step-like progression showing the incremental contribution of an additional PC to the cumulative variance. This helps in assessing the trade-off between dimensionality reduction and information retention. For instance, a combination of the first four PCs accounts for 81% of the information in the original data.

Therefore, mean centred data can be projected to low-dimensional space by selecting β-numbers of PCs to obtain the transformed data in a new feature space. For instance, if $\beta = 5, \hat{X} \in \mathbb{R}^{m \times n}$ is projected by the first-five PCs to low-dimension data as defined by:

$$\hat{X} = \in \mathbb{R}^{n \times \beta} = \hat{X}\tilde{P}_{C,} = \begin{bmatrix} 1.8476 & 2.6497 & 2.0291 & -0.3513 & -1.3334 \\ -0.4752 & -0.6773 & -0.9915 & 0.8523 & -0.6048 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1.3360 & -2.5567 & 1.5964 & -0.2954 & 0.1533 \\ -1.3746 & -0.0022 & 0.6273 & -1.7219 & 1.0319 \end{bmatrix} \tag{4.23}$$

Linear Discriminant Analysis (LDA)

The LDA, often known as Fisher's discriminant analysis (FDA), is used in the fields of pattern classification and dimensionality reduction [249] [250]. This statistical technique is also utilised in various other disciplines like, biometrics [251][252], bioinformatics [253], and chemistry [254]. Its primary objective is to identify a linear combination of attributes that effectively describes or distinguishes multiple classes of objects or events. The LDA technique is used to reduce the dimensionality of a dataset by optimising the ratio of within-class variance to between-class variance, hence ensuring the highest possible level of class separability. In [249] the visualisation and analysis of lower-dimensional spaces or variables can be facilitated using of various ML tools. The LDA algorithm accomplishes this transition through three fundamental phases; Firstly, it calculates the separability, or between-class variance, among different classes. Secondly, it computes the distance between each sample within a class and the mean, also known as within-class variance. Lastly, it constructs a lower dimensional space that minimises within-class variance and maximises between-class variance [249].

Step 1: Between-class variance calculation

The between-class variance (G_S), also known as the between-class matrix, quantifies the dissimilarity between the mean of the τth class (μ_τ) and the overall mean (μ). The LDA technique aims to identify a lower dimensional space that maximises the variation between classes [249]. For instance, considering a matrix A, which is specified as:

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}, \tag{4.24}$$

where each sample, a_{mn} is indicative of a point in a space of dimension l. Thus, l features ($a_\tau \in \mathcal{R}^{ln}$) can represent each sample. In evaluating the efficacy of the LDA feature extraction method, the dimension n is a crucial criterion n, as will be elaborated upon subsequently, determines the dimension of the feature vector derived through the LDA method. When dividing A_{mn} into p classes according to the following definition:

$$A_{mn} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_p \end{bmatrix}, \tag{4.25}$$

where ψ_p is a matrix in $A_{m,n}$ ($\psi_p \in A_{m,n}$). Also, assume that each class has χ samples, where χ_τ denotes the number of samples of the τth class. Thus, the total number of samples (υ) can be represented mathematically as:

$$\upsilon = \sum_{\tau=1}^{\mathrm{p}} \chi_\tau. \tag{4.26}$$

Consequently, to calculate (G_S), the difference in distance $(d_\tau - d)$ between the various classes is initially ascertained according to the definition in [249] and Equation 4.27:

$$(d_\tau - d)^2 = (Z^T\mu_\tau - Z^T\mu)^2 = Z^T(\mu_\tau - \mu)(\mu_\tau - \mu)^T Z, \tag{4.27}$$

In the given context, d denotes an approximation of the overall mean across all classes $d = Z^T\mu$, whereas d_τ represents an approximation of the mean for the τth class $Z^T\mu_\tau$. The LDA's transformation matrix is represented by the symbol Z, where $\mu_\tau(1 \times l)$ represents the mean of the τth class as specified in Equation (4.28), and $\mu(1 \times l)$ represents the sum mean of all classes as specified in Equation (4.29) [249].

$$\mu_\varsigma = \frac{1}{\varepsilon_\varsigma}\sum_{a_\tau \epsilon \psi_\varsigma} a_\tau, \tag{4.28}$$

$$\mu = \frac{1}{\upsilon}\sum_{\tau=1}^{h} a_\tau = \sum_{\tau=1}^{\delta} \frac{\varepsilon_\tau}{\upsilon}\mu_\tau. \tag{4.29}$$

The formula $(\mu_\tau - \mu)(\mu_\tau - \mu)^{\ominus}$ represents the difference in distance between the mean of the τth class (μ_τ) and the total mean (μ) in Equation (4.27). This is also referred to as the between-class variance of the τth class G_{S_τ}. Consequently, Equation (4.27) can be expressed as [249]:

$$(d_\tau - d)^2 = Z^T G_{S_\tau} Z \tag{4.30}$$

Consequently, the calculation for the total between-class variance, which is the sum of all the τth between-class variances, is as follows [249]:

$$G_S = \sum_{\tau=1}^{P} \varepsilon_\tau G_{S_\tau}. \tag{4.31}$$

Step 2: Within-class variance calculation

The variation between samples and the class's mean is represented by the within-class variance (G_ς), which is also known as the within-class matrix. LDA, identifies a space with fewer dimensions to reduce the variance within classes [249]. In accordance with [249] the computation of the within-class variance for each class G_{ς_φ} is given by:

$$\sum_{a_\tau \epsilon \psi_\varsigma, \varsigma=1,\ldots,\mathrm{g}} (Z^T a_\tau - d)^2 = \sum_{a_p \epsilon \psi_\varsigma, \varsigma=1,\ldots,\mathrm{p}} Z^T G_{S_\varsigma} Z, \tag{4.33}$$

G_{ς_φ} is defined as [255]:

$$G_{\varsigma_\varphi} = \sum_{\tau=1}^{\varepsilon_\varsigma} (a_{\tau\varsigma} - \mu_\varsigma (a_{\tau\varsigma} - \mu_\varsigma)^{\ominus}, \tag{4.34}$$

The τth samples in the ςth class are denoted by $a_{\tau\varsigma}$. The total within-class variance, as defined in reference [249], is equivalent to the sum of all the τth within-class variances.

$$G_\varsigma = \sum_{\tau=1}^{\mathrm{P}} \mathrm{G}_{\varsigma_\varphi}. \tag{4.35}$$

Step 3: Lower dimensional space construction

Subsequently, the transformation matrix (Z) of the LDA technique is estimated utilising the between-class variance (G_S) and within-class variance (G_ψ), as [249]:

$$\arg\max_Z \frac{Z^T G_S Z}{Z^T G_\psi Z}, \tag{4.36}$$

where Equation (4.36) is the Fisher's criterion. The LDA method is therefore occasionally referred to as Fisher's discriminant analysis. The modified form of Equation (4.36) is Equation (4.37) [249]:

$$G_\psi Z = \pi G_S Z. \tag{4.37}$$

The eigenvalues of the transformation matrix Z are denoted by π in Equation (4.37). By calculating the eigenvalues $(\pi = \pi_1, \pi_2, \dots, \pi_D)$ and eigenvectors $(\Phi = \Phi_1, \Phi_2, \dots., \Phi_D)$, the solution to the $n \times n$ transformation matrix $(Z = G_{\varphi}^{-1} G_S)$ can be obtained. ϕ denotes non-zero vectors that represent the directions of the new space, while π are scalar values. This signifies that every Φ corresponds to an axis in the LDA space, and the π value associated with each Φ signifies the magnitude [249]. The Φ with the largest eigenvalue and, hence, the highest variance is the first LDA variable. This indicates that it conveys the most information achievable. As a result, the eigenvector is presented in decreasing order based on the message it contains. This method of arraying the LDA variables allows for dimensionality reduction with negligible or no information loss. With the dimension, n, indicating the number of LDA variables, these LDA variables can be created as a linear combination of the applications to be analysed.

The SVD factorizes the transformation matrix Z into three matrices:

$$\hat{X} = UKV^T \tag{4.38}$$

where U is the left singular vector, K is a diagonal matrix, and V is the right singular vector with a $n \times n$ dimension. The π are the entries of the diagonal of K arranged in descending order while the π are the Φ or the LDA variables.

4.4.4 Proposed Linear Discriminant Analysis Feature Vector for Machine Learning

In this section, we reconstruct the LDA variables obtained from Equation (4.38) to create a feature vector that can be utilised with the ML algorithms for the automated detection of service classes in FCAs. It is crucial to demonstrate the method by which the dimension, denoted as n, of the feature vector in LDA is computed. The dimension, n dictates the computational time complexity the LDA will impose on the ML training process. Therefore, to provide a reasonable computational time complexity, it is imperative to minimise the dimension, n . This implies that there exists a trade-off between the false discovery rate (FDR) and sensitivity performances, as well as the computational time complexity, when selecting a value for n. Hence, the selection of the dimension, n , in this chapter is determined through simulation runs, as elaborated.

$$Z = (z_1 \;\; z_2 \;\; \dots \;\; z_i) \;, \tag{4.39}$$

Essentially, the Z are represented by matrix $Z_{m,n}$

$$Z_{m,n} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m,n} & z_{m,n} & \cdots & z_{m,n} \end{bmatrix}. \tag{4.40}$$

Consequently, the computation of the between-class variance (G_S) and within-class variance (G_{φ}) is performed sequentially based on the Equations (4.26) and (4.29), respectively. Afterwards, the Fisher's criterion Equation (4.21) is utilised to calculate the transformation matrix (Z), which is then deconstructed to acquire:

$$V = \begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,m} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m,1} & V_{m,2} & \cdots & V_{m,m} \end{bmatrix}. \tag{4.41}$$

Thus, the feature vector is derived in this study from Equation (4.41), and it should be noted that the feature vector created from the LDA method has a dimension of m. Afterwards, if the Quality of Service (QoS) matrix under analysis, the feature vector can be computed as a matrix according to Equation (4.43).

$$Z_{LDA} = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,m} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{k,m} \end{bmatrix} \tag{4.43}$$

4.4.5 Numerical Example of Linear discriminant features vectors

Using A as your feature matrix, you have a dataset containing features and class labels, where each row symbolizes a sample, each column a feature, and y is the vector of class labels.

$$A_{mn} = \begin{bmatrix} 0.0003 & 0.0002 & 0.0002 & 0.5541 & 0.0052 & 0.0090 & 0.0100 & 1.3203 & 0.0000 \\ 0.0000 & 0.0001 & 0.0001 & 0.1172 & 0.0048 & 0.0001 & 0.0170 & 2.7394 & 0.0000 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.0586 & 0.0002 & 0.0002 & 0.0001 & 0.0003 & 0.0004 & 0.0042 & 0.0005 & 0.0000 \\ 0.0001 & 0.0001 & 0.0001 & 0.4258 & 0.0031 & 0.0002 & 0.0132 & 0.1432 & 0.0000 \end{bmatrix}$$

(4.44)

Therefore, the between-class variance (C_B) is computed as in Equation (4.8) to obtain:

$C_B =$

$$\begin{bmatrix} 0.0004 & 0.0000 & -0.0000 & -0.0015 & -0.0000 & -0.0000 & -0.0001 & -0.0191 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0001 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0001 & -0.0000 \\ -0.0015 & -0.0000 & -0.0000 & 0.0547 & 0.0006 & -0.0001 & -0.0001 & 0.6127 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 0.0006 & 0.0000 & -0.0000 & -0.0000 & 0.0077 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 & -0.0001 & -0.0000 & 0.0000 & 0.0000 & -0.0014 & 0.0000 \\ -0.0001 & -0.0000 & 0.0000 & -0.0001 & -0.0000 & 0.0000 & 0.0000 & -0.0022 & 0.0000 \\ -0.0191 & -0.0001 & -0.0001 & 0.6127 & 0.0077 & -0.0014 & -0.0022 & 7.7081 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 \end{bmatrix}$$

.

Similarly, the within-class variance (C_W) is compared as in Equation (4.21) to obtain:

C_W

$$= \begin{bmatrix} 0.0001 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0330 & -0.0000 & 0.0000 & 0.0000 & 0.0026 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0026 & 0.0000 & 0.0000 & -0.0000 & 4.4122 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

Next, we compute the transformation matrix, Z, using the equation (4.23):

$Z = C_W^{-1} C_B$

$$= \begin{bmatrix} 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0001 & -0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 0.0002 & 0.0000 & -0.0000 & -0.0000 & 0.0026 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0004 & 0.0000 \\ -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0004 & 0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -0.0031 & 0.0000 & -0.0000 & -0.1183 & -0.0012 & 0.0033 & 0.0021 & -1.5371 & 0.0000 \end{bmatrix}$$

The LDA variables can therefore be obtained by finding the SVD of LDA_matrix such that

$$\hat{X} = UKV^T,$$

Where,

$$U = \begin{bmatrix} 0.0001 & 0.0856 & 0.2144 & -0.1317 & -0.9626 & -0.0513 & -0.0030 & 0.0039 & -0.0041 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.8782 & 0.3951 & -0.2694 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0931 & -0.6937 & 0.7142 \\ -0.0000 & -0.0080 & 0.0116 & -0.0131 & -0.0500 & 0.9915 & 0.0555 & -0.0713 & 0.0765 \\ -0.0017 & -0.5432 & -0.6261 & -0.5479 & -0.1123 & -0.0098 & -0.0009 & 0.0011 & -0.0012 \\ 0.0003 & -0.3840 & 0.7431 & -0.5096 & 0.2016 & -0.0082 & -0.0005 & 0.0006 & -0.0006 \\ 0.0002 & -0.7417 & 0.0985 & 0.6501 & -0.1326 & -0.0052 & -0.0003 & 0.0004 & -0.0004 \\ -0.0000 & -0.0003 & -0.0004 & -0.0004 & -0.0003 & -0.1186 & 0.4658 & -0.5979 & 0.6415 \\ 1.0000 & -0.0007 & -0.0013 & -0.0010 & -0.0001 & 0.0000 & 0.0000 & -0.0000 & 0.0000 \end{bmatrix}$$

$$K = \begin{bmatrix} 1.5417 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.0020 & 0.6933 & 0.5966 & -0.2219 & -0.3366 & -0.0306 & 0.0066 & -0.0067 & 0.0000 \\ 0.0000 & 0.0002 & 0.0014 & -0.0048 & 0.0292 & -0.1636 & 0.8751 & 0.4545 & 0.0050 \\ -0.0000 & 0.0011 & 0.0056 & 0.0117 & 0.0342 & -0.0328 & 0.4546 & -0.8891 & -0.0191 \\ -0.0767 & -0.6806 & 0.7072 & -0.1719 & -0.0350 & 0.0053 & 0.0003 & -0.0000 & -0.0000 \\ -0.0008 & -0.0154 & -0.0464 & -0.0680 & -0.1548 & 0.9696 & 0.1647 & 0.0412 & 0.0023 \\ 0.0021 & -0.1133 & -0.3376 & -0.8984 & -0.2287 & -0.1154 & -0.0068 & -0.0221 & 0.0005 \\ 0.0014 & -0.2016 & -0.1570 & 0.3305 & -0.8985 & -0.1330 & 0.0166 & -0.0181 & -0.0000 \\ -0.9970 & 0.0505 & -0.0565 & 0.0122 & 0.0018 & -0.0015 & -0.0002 & -0.0001 & -0.0000 \\ 0.0000 & -0.0001 & -0.0004 & -0.0009 & -0.0010 & 0.0020 & -0.0039 & 0.0194 & -0.9998 \end{bmatrix}$$

The eigenvectors, , are represented as the entries in the columns of V, which serves as the decomposed LDA variables. Consequently, V is rebuilt using Equation (4.27) to create the LDA feature vector as specified by Equation (4.28):

$$Z = \begin{bmatrix} -0.0001 & -0.0001 & 0.0000 & 0.0001 & 0.0002 & -0.0003 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ -0.0002 & 0.0013 & 0.0003 & 0.0003 & -0.0006 & 0.0015 \\ 0.0004 & -0.0002 & 0.0004 & -0.0008 & 0.0011 & -0.0013 \\ 0.0010 & 0.0004 & -0.0003 & 0.0004 & 0.0006 & -0.0009 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 1.0000 & -1.0000 & 1.0000 & 1.0000 & -1.0000 & 1.0000 \end{bmatrix}$$

$$\phi = \begin{bmatrix} 7.9801 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.0553 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.3069 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4986 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3215 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0155 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0000 \end{bmatrix}$$

Ω

$$= \begin{bmatrix} -0.0001 & -0.0001 & 0.0000 & 0.0001 & 0.0002 & -0.0003 & -0.0002 & -0.0002 & -0.0002 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ -0.0002 & 0.0013 & 0.0003 & 0.0003 & -0.0006 & 0.0015 & -0.0022 & -0.0022 & -0.0022 \\ 0.0004 & -0.0002 & 0.0004 & -0.0008 & 0.0011 & -0.0013 & -0.0009 & -0.0009 & -0.0009 \\ 0.0010 & 0.0004 & -0.0003 & 0.0004 & 0.0006 & -0.0009 & -0.0004 & -0.0004 & -0.0004 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.0000 & -1.0000 & 1.0000 & 1.0000 & -1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$$

Algorithm 3: Feature Extraction with Linear discriminant analysis

Input: $[A_i, \tau, n]$
Output: Feature vector, Z_{LDA}
1. Find the dimension, n of the LDA components
2. Restructured A_i wrt τ as in Equation. (4.11)
3. Calculate C_B as in Equation (4.18)
4. Calculate C_w as in Equation (4.21)
5. Compute Z as in Equation (4.43)
6. Calculate the eigenvector, ϕ and eigenvalues Ω as in Equation (4.24)
7. Calculate the LDA components, as in Equation (4.28)

In addition, the computational complexity analysis of the PCA, and LDA algorithms is presented in Table 4.1 using the big O notation. As shown, LDA has more computational complexity as compared to the PCA. This is largely attributed to the calculation of the between-class and within-class covariance matrices, which scale with the number of class and sample size. Thus, aside from obtaining a better , the GPT also exhibits a reduced computational time complexity in comparison to the other considered. Lastly the most computational step for both is the eigen decomposition ($\mathcal{O}(d^3)$).

Table 4.1: Determining computation complexity for PCA and LDA algorithms

Steps	Algorithms	Description	Complexity	Type
Finding the dimension of components	PCA, LDA	Determining the number of principal or discriminant components	$\mathcal{O}(1)$	Time-efficient
Restructure data	PCA, LDA	Reshaping data with respect to input parame-ters	$\mathcal{O}(p.q)$	Computationally intensive
Calculate covar-iance matrix	PCA	Computing between-class or within-class covariance.	$\mathcal{O}(n.d^2)$	Computationally intensive
Compute eigenvalues and eigenvectors	PCA, LDA	Solving eigen decomposition for dimensionality reduction.	$\mathcal{O}(d^3)$	Computationally intensive
Project data onto reduced components	PCA, LDA	Determining the proportion of variance	$\mathcal{O}(N.d.n)$	Computationally intensive

		captured by components.		
Calculate variance explained	PCA	Determining the proportion of variance captured by components.	$\mathcal{O}(d)$	Time-efficient
Normalize feature vector	PCA, LDA	Scaling or standardizing the resulting feature vector.	$\mathcal{O}(N.d)$	Time-efficient

4.5 Summary

This chapter introduces PCA and LDA as FE algorithms. We specifically engineer the principal components from PCA and the discriminant components from LDA to extract attributes from datasets related to Fog Computing applications. We will feed the generated feature matrices into machine learning to detect and categorise service segments. PCA identifies a low-dimensional linear subspace that captures the majority of variance in a dataset, but it may not successfully construct a low-dimensional non-linear subspace if one exists inside the dataset. Additionally, when using LDA, non-linear attributes in the datasets may change how well the spatiotemporal trends in the basic system are shown. Nonetheless, the Fog Computing application dataset occasionally has nonlinear properties. Therefore, we propose advanced feature extraction strategies to enhance the use of PCA and LDA for feature extraction.

CHAPTER:5 DESCRIPTION OF THE DATASET AND EXPERIMENTAL SETUP

5.1 Introduction

In Chapter 4, ED-based FE techniques were developed, specifically utilising PCA and LDA. Thus, the emerging feature vectors developed from PCA and LDA will be separately fed into the ML models for the detection and classification of the FCAs. Firstly, a detailed description of the datasets used in this study is given. Secondly, a summary of the workings of ML models as used in this study is presented, and lastly, the implementation is described.

5.2 Data overview

This section describes the datasets used for training and testing the ML models in this study. Meanwhile, a major challenge in machine learning is the availability of data. As data from real-world operational systems may not be readily available by service providers for the purpose of research due to the sensitivity of such data. Thus, empirical datasets from [184] are utilized for experiment and analysis in this study. An independent random number generator was employed by the authors to generate synthetic data, ensuring that all features adhered to QoS requirements associated with specific classes of service. Each feature was sampled from a uniform probability distribution within intervals corresponding to the defined QoS constraints. This method allowed for the creation of a dataset that mirrors practical operational environments, facilitating the accurate evaluation of performance in FCAs while maintaining compliance with QoS benchmarks. The employed features were devoid of transitory data, eliminated by the Moving Average of Independent Replications method [256]. This guaranteed that the feature values comprised both safe and borderline instances. The safe instances were in highly uniform regions concerning the class label, whereas borderline examples existed around, class borders where distinct classes intersect. Furthermore, a third set of feature values was generated, referred to as noisy instances, to assess the robustness of the classifiers. In this study, the term noise sample denotes the samples generated to illustrate the degradation of their feature values.

The dataset is composed of 18,200 mutually exclusive applications at the acceptable QoS requirement interval values. Furthermore, when analysed it contains more than 127,400 samples with 9 features per individual applications. These features are bandwidth, delay sensitivity, reliability, security, data storage, data location, mobility, scalability, and loss sensitivity. As indicated in [184], the features are specifically mapped for the FCAs and 3can be classified in seven classes with priority levels of one to seven with one the highest and seven lowest. Table 2.1, located in chapter 2 summarises the description of the seven classes of service: (1) mission critical, (2) real-time, (3) interactive, (4) conversational, (5) streaming,

(6) CPU-bound, and (7) best effort for FCAs. These levels of service are ordered according to the QoS requirements and thus, enable the network traffic to be prioritised and differentiated based on its importance.

5.2.1 Data Preprocessing

Preprocessing is conducted to ensure that every sampling point within the dataset makes an equitable contribution to the analysis of features [226]. The process of normalising the variables in the dataset serves to reduce any potential bias that could result from a substantial difference between the large and small QoS values of the observation variables. The normalisation procedure encompasses the calculation of the mean and standard deviation of the quality of service (QoS) vector. Subsequently, the mean is subtracted from each sampling point, and the result is divided by the standard deviation. Ensuring the consistency and

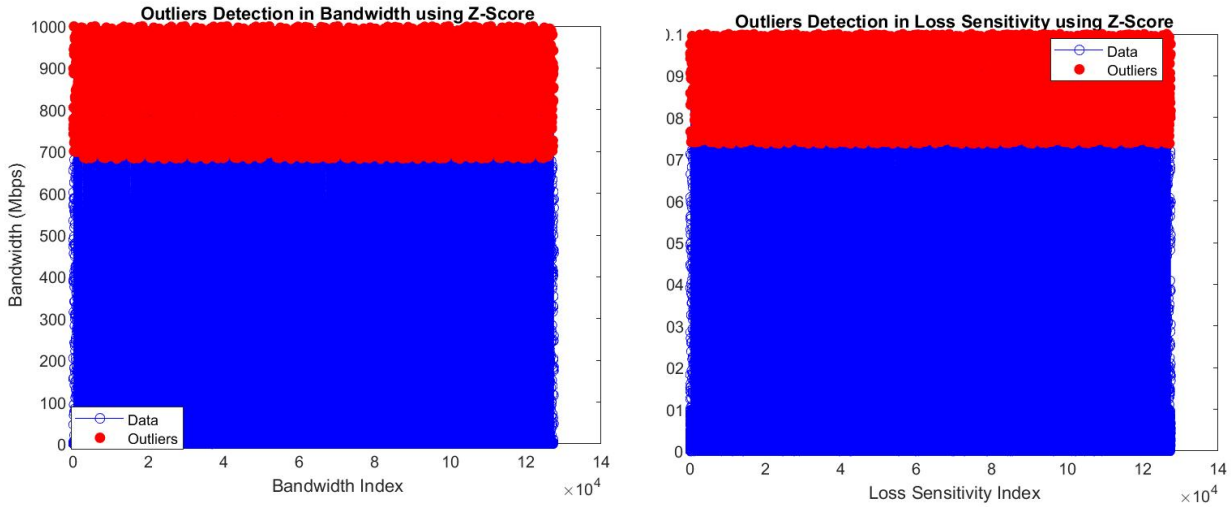

Figure 5.1: Data features with 3 or more outliers

impartiality of the data prior to conducting further analysis is a crucial step in the experimental setup.

The QoS observations may contain different features that may not contribute meaningful information for the classification of FCAs and thus, lead to poor generalization on new, unseen data. FE helps the model to capture the underlying patterns in the data more effectively and simplify the dataset, making it computationally more efficient and reducing the time required for training and testing the model [257]. There are no missing values, as complete analysed encompasses also checking for missing value and other factors that impact machine learning such as outliers and noise etc. Figure 5.1 indicates that there only two features with more than three outlier's bandwidth and loss sensitivity.

Meanwhile, another important relationship to analyse is covariance and correlation of the data using the covariance or correlation. The covariance matrix provides insights into the relationship between variables in the dataset. The variance along the diagonal shows how much each feature varies with itself, providing insight into the spread of each variable. Covariance in the off-diagonal cells indicates how two features vary together. If two features have a high covariance positive or negative, they are likely not independent and may contain redundant information. If covariance is positive, it implies that as one variable increases,+ the other tends to increase as well; conversely, a negative covariance indicates an inverse relationship, where one variable tends to decrease as the other increases. However, the magnitude of covariance is challenging to interpret because it depends on the units and scales

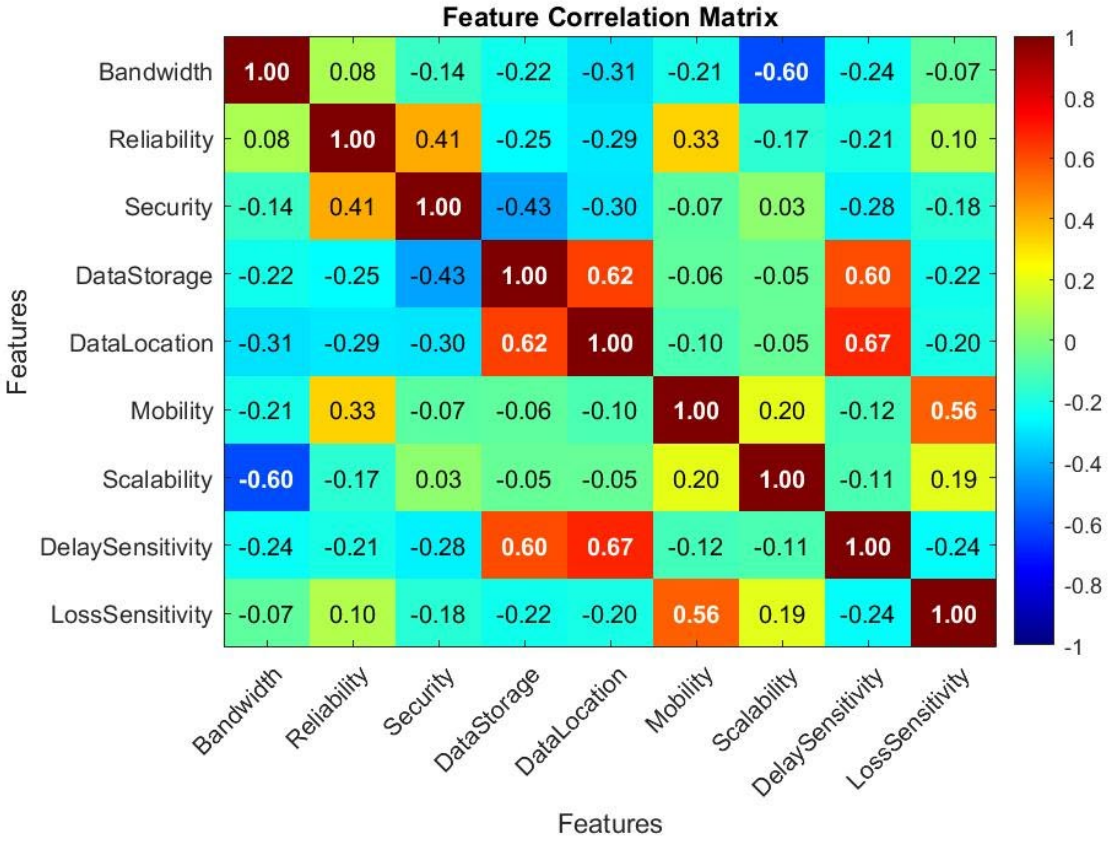

Figure 5.2: Feature association estimates through heatmap

of the variables, which limits direct comparisons across different datasets. Covariance is often used in multivariate analyses as part of the covariance matrix, which plays a central role in algorithms such as PCA by identifying relationships and variances within data dimensions, crucial for dimensionality reduction and data simplification.

Correlation, on the other hand, is a normalized version of covariance, yielding a dimensionless measure that ranges from -1 to 1. Correlation matrix Figure 5.2 helps with addressing multicollinearity, by detecting highly correlated variable. Hence, with our current data some of observations feature variables are highly correlated, which can lead to an instability in model

coefficients and make it challenging to interpret the importance of individual features. Figure 5.2 is a complete correlation analysis of all the different variables features. Indicated in bold font is the statistically significant relationship exceeding 50% between variables. Example of these variables' relationship are data storage and data location (0.62), data storage and latency (0.60), loss sensitivity and mobility (0.56), data location and latency (0.67). Meanwhile, the presence of a negative sign in (-0.60) the correlation coefficient variables of bandwidth and scalability suggests the existence of an inverse link between the two properties. Thus, the use of FE is a must on the current dataset.

Since we have correlation matrix A, which represent our dataset. The eigenvalues of A are found by solving the characteristic equation:

$$det(A - \lambda I) = 0 \tag{5 .1}$$

Where det denotes the determinant of a matrix, λ, is scalar the eigenvalue, and I is the identity matrix of the same size as A. Solving this equation give us the eigenvalues of A. These eigenvalues are real as illustrated by the symmetric matrix (covariance matrices). Once the eigenvalues are found, we can compute the corresponding eigenvectors. For each eigenvalue λ, solve the equation:

$$(X - \lambda I)v = 0 \tag{5.2}$$

where v is the eigenvector corresponding to the eigenvalue λ. This is a system of linear equations, which can be solved using SVD method.

The SVD method decomposes matrix A to yield unitary matrices, U and V, as well as a diagonal matrix K with the same dimension as matrix A and non-negative diagonal components in decreasing order, such that: Linear algebra methods are used to identify main components, which are linear combinations of the original qualities. The variables are mutually perpendicular and effectively represent the greatest extent of variability within the dataset.

$$A = UKV^T \tag{5.3}$$

Step 3: Formation of the Projection Matrix

Supposing that $n, n < m$ components have the largest variance, that is, the most information based on the PCA, the n −dimensional matrix is expressed as:

Each entry (j, k) in the covariance matrix Σ is the covariance between feature j and feature *k*:

$$\sum_{jk} = cov(j, k), \tag{5.5}$$

Table 5:1 $n \times n$ matrix for PCs dataset

Feature	**PC**	**PC**	**PC**	**PC**	**PC**	**PC**
	1	2	6	7	9	9
1-Bandwidth	31,61	31,61	31,61	31,61	31,61	31,61
2-Reliability	21,34	21,34	21,34	21,34	21,34	21,34
3-Security	15,35	15,35	15,35	15,35	15,35	15,35
4-Data Storage		12,92	12,92	12,92	12,92	12,92
5-Data Location			5,53	5,53	5,53	5,53
6-Mobility			3,92	3,92	3,92	3,92
7-Scalability				3,64	3,64	3,64
8-Delay Sensitivity					3,24	3,24
9- Loss Sensitivity						2,45
	68,30	81,22	90,67	94,31	97,55	100,00

Here, the importance level of each input feature on the model is evaluated based on the complementarity of a correlation analysis with PCA. This analysis helps to examine the degree of association between two QoS requirements.

The variables are mutually perpendicular and effectively represent the greatest extent of variability within the dataset. Figure 5.3 illustrates the scree plot, which displays the percentage of variability explained by each primary component. According to the data presented in Figure 5.3, it can be observed that the initial seven principal components account for a total variation of 94.31%. The initial component in isolation accounted for less than 35% of the variance, suggesting the potential requirement for further components. Additionally, it can be observed from Figure 5.3 that the initial three principal components account for approximately 66.67% of the overall variability in the standardized ratings.

Furthermore, alongside the percentage of variability elucidated by each principal component, a vector in the biplot reflected all nine qualities. The orientation and magnitude of the vector represent the influence of each characteristic on the two primary components depicted in the plot. As exemplified, Figure 5.4 displays the coefficients of each characteristic with respect to the first two main components. The remaining five primary components were likewise visualised as biplots. The contribution of the qualities to each primary component is presented in Table 5.4. The interpretation of the principal components relies on identifying the variables that exhibit the strongest correlation with each component. This entails determining which variables possess larger values, situated at the greatest distance from zero in either positive or negative direction. The determination of which values should be deemed as large is a subjective matter that is contingent upon one's understanding of the system being assessed.

The relevance of a correlation value in our study was established when it exceeded 0.48. This threshold was chosen as it represents the highest value within each primary component. Furthermore, most variables exhibiting this elevated value demonstrate a strong correlation,

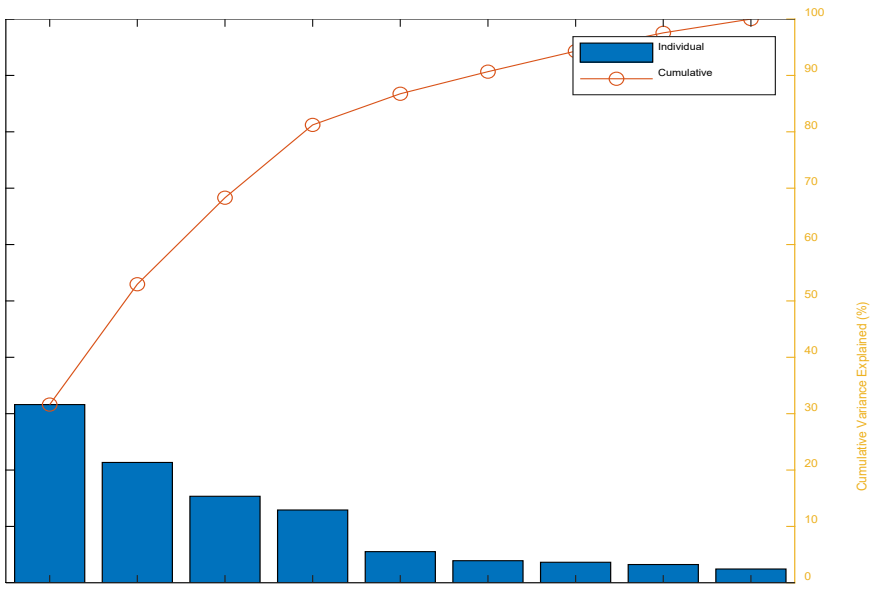

Figure 5.3: Scree plot illustrates the percentage of variance attributed to each PC

as evidenced by the components of the correlation matrix. The correlation values of significant magnitude are denoted in boldface within Figure 5.2.

The interpretation of the principal component results is contingent upon the determination of a significant threshold value. The initial main component exhibited a significant correlation with three of the original attributes. Hence, it can be shown that the initial principal component exhibits a positive correlation with the variables of Data storage, Data location, and Delay sensitivity, indicating a co-variation among these three properties. In contrast, the coefficients associated with the second major component indicate an inverse relationship between the feature bandwidth and the feature Mobility and Scalability. This observation indicates that increased mobility is correlated with alterations in the network topology, leading to heightened variability in communication links and diminished bandwidth availability. Furthermore, because of the limited nature of bandwidth, an increase in the number of users connected to the 5G FRAN results in a drop in the data transmission and reception rate for each individual user.

Additional characteristics that indicate a negative correlation are Data Storage and Loss sensitivity in the fifth principal component, as well as mobility and loss sensitivity in the seventh principal component. The third, fourth, and sixth principal components exhibit an increase in conjunction with a single variable that possesses a value of 0.48 or above. The variables under consideration are Security, Reliability, and Delay sensitivity, in that order. Consequently, the

third, fourth, and sixth major components can be construed as indicators of the extent to which the utilisation of was essential for the execution of the application, the promptness with which failed fog components should be reinstated, and the level of susceptibility of the FCA to latency. Based on the research and considering the nuance of the investigation, it has been determined that certain factors within the Fog computing ecosystem warrant greater attention. Consequently, redundant features such as Data location, Data storage, and Mobility have been eliminated. Therefore, a total of six attributes were chosen and retained for the classification stage, out of the initial nine attributes. The factors considered in this study were Delay sensitivity, Scalability, Loss sensitivity, Security, Reliability, and Bandwidth.

5.3 Summary of the ML Process for Detection and classification for Fog Computing applications

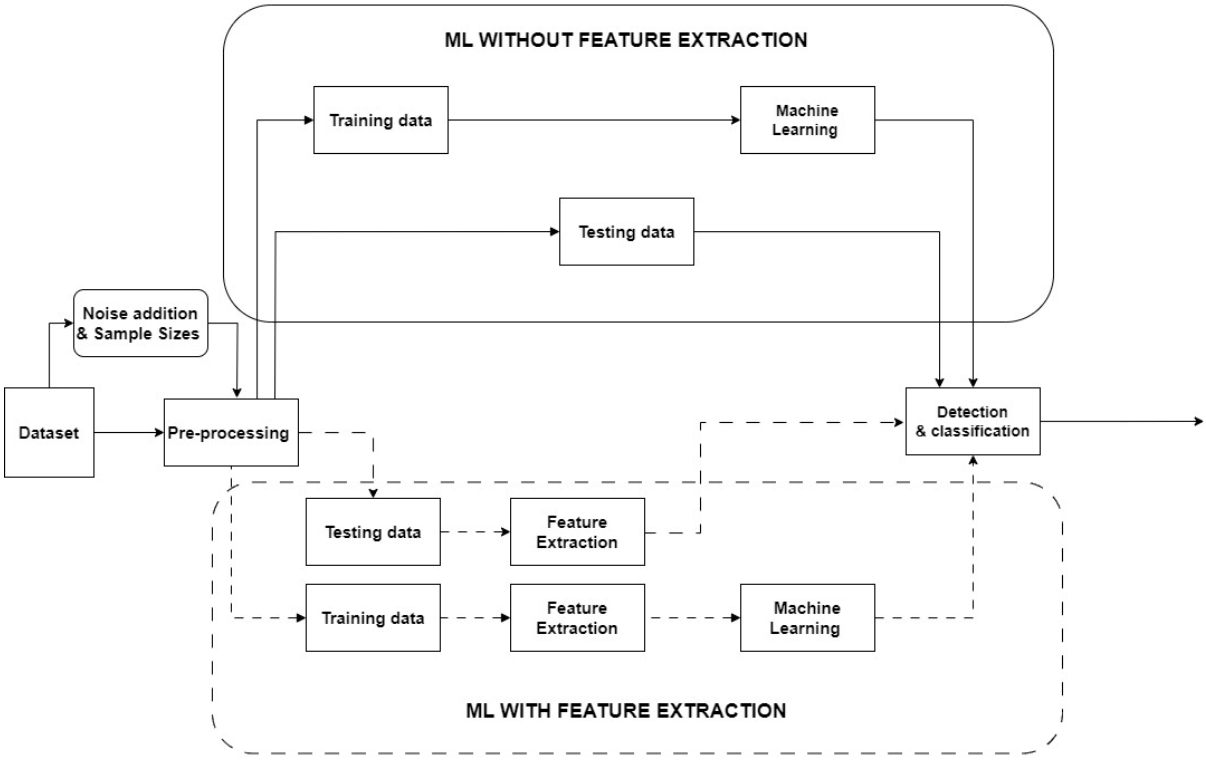

Figure 5.4: ML process with and without ED-FE for detection and classification of Fog Computing applications

Figure 5.4 presents the block diagrams that delineate the ML process for detecting and classifying FCAs. Prior to the division of the dataset into two uneven sets of training and testing data subsets, incremental noise (φ) at levels of 0, 20, 40, 60 decibels (dB)) for each samples size, τ, get to be added for complete model operation. The τ-demension of the dataset range from 100, 50, 20, 10, 2 percentage of complete data. These alteration to the dataset and test of the ML models contribute with improving the robustness and generalisation. These are key attribute for efficient ML model performance with new unseen data and contributing to reliable

trustworthy system. The data from the FCAs, as gathered from [184], undergoes preprocessing. The preprocessed dataset was partitioned into two unequal segments: 70% and 30%. With 70% of the dataset is allocated for the training phase of the ML process and was therefore designated as the “Training” data. The remaining 30% was allocated for the detection and classification phase of the machine learning process, commonly known as the “Testing” data.

As, for the preprocessing stage, the complete data was changed from table to array, which assist with computation efficiency. Thereafter the labels and feature were separated as this was a label dataset which make the choice of ML for supervised learning. Furthermore, depicted on the diagram were two independent ML sections one without ED-based FE and another with ED-based FE. Both systems were trained with 70 percentage of the t-dimensional data. Consequently, the remaining 30 percent testing data were also used for the testing phase for both systems. There were no dimensionality reduction methods such as PCA or LDA used on the training data in the first method. The Decision Tree, SVM, k-NN, Random Forest, and Neural Network models all took patterns straight from the raw features. To reduce forecast mistakes and avoid overfitting, the training method included tuning hyperparameters and validating the model on a regular basis. For the second method, eigen decomposition methods were used to change the data. PCA was used to narrow down the feature space by finding the parts that caught the most variation, and LDA was used to improve the ability to tell the difference between classes for classification tasks. This change made the set of features smaller and more important, which made computations easier and improved model performance.

Meanwhile, after the training stage the predict probability stage start where the test data is utilised with each of the recently trained models. The output which is the predicted labels is converted to categorical variables. Thereafter the predicted labels output from each model is compare the (original data) label test data, which is also a categorical variable. Consequently, the performance for each of the models is evaluated using the performance matric discussed in chapter 2. On the section, which include ED-FE techniques to extract features. The extracted features were then transformed into an appropriate feature matrix, for the ML models chosen. The ED-FE was used on the train and test data. Whereas the training and the predict function follow the same sequence for the each of the models.

The last 30% of the information, which had not been used for training, was used in the testing stage. In the first method, the models made predictions based on raw test data that they had not seen, and their success was measured by F1 score, ROC-AUC, accuracy, precision, and recall. Differences between what was expected and what happened were analysed for performance for solid and reliable each model was. In the second method, the test dataset

was changed using the same eigen decomposition methods that were used for training. The changed data was then used to test the model's performance, making sure that the way features were shown was consistent. By comparing the outcomes of both methods, we learnt more about the effects of reducing the number of dimensions. PCA and LDA not only got rid of noise and features that weren't important, but they also made computations faster, which was especially helpful for models that need to deal with high-dimensional data, like k-NN and Neural Networks. This organised two-stage test showed that feature extraction methods can improve the accuracy, reliability, and general performance of models.

5.4 Experiments

This section describes the experimental set-up for the implementation of the developed models.

5.4.1 Implementation

This study systematically examines the impact of PCA and LDA on classification accuracy, model complexity, and efficiency by implementing and comparing each model across various configurations, offering valuable insights into optimal preprocessing strategies for machine learning models in fog computing applications.

This study examines the implementation and assessment of five machine learning models—decision tree, support vector machine (SVM), k-nearest neighbors (k-NN), random forest, and artificial neural network (ANN) to classify fog computing applications utilizing eigen decomposition and feature extraction techniques (ED-FE). We evaluated the classification models using noise-injected datasets to replicate actual fog computing scenarios. We partitioned the datasets into training and testing segments, implementing progressively increasing noise levels to evaluate the robustness and efficacy of the models. We developed five unique machine learning models to manage different portions of the data, representing fog computing applications. We optimised each model with hyperparameters to identify fog computing applications while mitigating noise, enabling thorough assessment across diverse data circumstances.

Furthermore, we created and evaluated each machine learning model under three distinct conditions: first, without feature extraction; second, using principal component analysis (PCA) with singular value decomposition (SVD); and third, using linear discriminant analysis (LDA) with SVD. These configurations enable the study to evaluate the impact of dimensionality reduction approaches on classification accuracy, model efficiency, and resilience. The Decision Tree model used MATLAB's fitctree function, which was set up with the options "AlgorithmForCategorical" set to "exact" and "OptimizeHyperparameters" set to "auto." This made it easier to manage categorical data correctly and get the best hyperparameters. We

assessed this model without feature extraction on the original dataset (70% training and 30% testing), using PCA to capture primary variance components and LDA to enhance class separability, thereby elucidating the impact of dimensionality reduction on decision tree accuracy and efficiency ([258] [248] Fisher, 1936.

Meanwhile, we executed the SVM model using MATLAB's fitcecoc function with Error-Correcting Output Codes (ECOC) for multi-class classification. With model parameters like "AlgorithmForCategorical," "exact," and "OptimizeHyperparameters," "auto," this configuration breaks down the multi-class problem into several binary classifications. This makes the system run faster. We evaluated the SVM on the dataset without feature extraction, with PCA to preserve high variance features, and with LDA to enhance class separation, assessing the effect of each method on SVM performance [259][248].

We used the fitcknn function in MATLAB for k-Nearest Neighbours (k-NN), setting the parameters 'NumNeighbours' to 5, 'OptimizeHyperparameters' to 'auto', and 'Standardise' to 'true'. This configuration categorizes each observation based on the predominant class among its five nearest neighbors and implements standardization to normalize feature scales. We assessed the k-NN model under three conditions: without feature extraction, using PCA to convert features into uncorrelated principal components, and using LDA to improve class separation. We evaluated the impact of each dimensionality reduction technique on classification accuracy and efficiency [248].

We executed the Random Forest model using MATLAB's TreeBagger function, with 50 trees and the 'Method' set to 'classification.' This ensemble model consolidates predictions from many decision trees to enhance classification robustness. We evaluated the Random Forest model in three configurations: without feature extraction, with PCA to reduce dimensionality while maintaining informative variance, and with LDA to improve class discrimination. This methodology elucidates the impact of dimensionality reduction on the accuracy and computing efficiency of Random Forests [260][261]. MATLAB's patternnet function executed the Artificial Neural Network (ANN), incorporating 10 hidden neurones and a one-hot encoder for multi-class classification.

We assessed the ANN in three configurations: without feature extraction as a baseline; with PCA to streamline the model while preserving variance-rich components; and with LDA to enhance class separation, potentially enhancing the network's ability to distinguish intricate patterns. We can look at how PCA and LDA affect ANN performance in terms of model complexity and classification accuracy with these setups [262][248]. This study carefully looks at how PCA and LDA affect classification accuracy, model complexity, and efficiency by using

and comparing each model in several different setups. This gives us useful information about the best ways to prepare machine learning models for use in fog computing scenarios.

5.5 Conclusion

This chapter presents an in-depth discussion of the datasets used in this work and elaborates on the practical application of eigen decomposition and feature extraction (ED-FE) approaches, along with five machine learning (ML) models. We presented a block diagram that encapsulated the operational workflow of the ML models used in this investigation, followed by a comprehensive explanation of the experimental design. We used the deployed models, which were divided into baseline ML models (MLs), PCA-enhanced machine learning models (PCA-MLs), and LDA-enhanced machine learning models (LDA-MLs), in a planned way on fog computing application datasets that had different types of classes of activity. This chapter outlines the settings and modifications made to each model, examining the impact of various feature extraction methods on the models' performance in noise-injected and multi-class classification scenarios. Chapter 6 fully analyses and discusses the results of these experiments. The experiments investigate how reducing the number of dimensions affects accuracy, efficiency, and noise resilience in different machine learning settings.

CHAPTER:6 RESULTS AND DISCUSSION

6.1 Introduction

The developed ED-FE techniques in chapter four were fed into the proposed ML models of chapter 5 for the detection and classification of FCAs. In this chapter, we focus on the evaluation of these emerging models, namely: PCA-based Decision Tree (PCA-DT), PCA-based Support Vector Machine (PCA-SVM), PCA-based K-Nearest Neighbours (PCA-K-NN), PCA-based Random Forest (PCA-RT), PCA-based Artificial Neural Network (PCA-ANN), LDA-based Decision Tree (LDA-DT), LDA-based Support Vector Machine (LDA-SVM), LDA-based K-Nearest Neighbours (LDA-K-NN), LDA-based Random Forest (LDA-RF), and LDA-based Artificial Neural Network (LDA-ANN). Furthermore, these models were experimented on QoS datasets containing Fog computing applications information. The experiments were done using a Microsoft Windows 11 Pro computer with an AMD Ryzen 7 5700 CPU (@1.80 GHz) and 16 GB of RAM, running MATLAB 2020a. The results obtained from the various experiments carried out on each of the models, subject to various factors, are analysed and discussed in subsequent sections. The performances of the models are evaluated using the following metrics: Accuracy (α), Sensitivity (δ), False Discovery Rate (ϑ), AUC Mean and AUC Std. Lastly, also the comparisons are done of noise as related to accuracy, FDR, sensitivity, AUC Mean, AUC Std.

6.2 Performance Analysis of ML without ED-FE

In this section, we analyse and discuss the results obtained from the experiments conducted on DT, SVM, K-NN, RF, ANN, techniques. All these were simulated at, sample size, τ = 100, 50, 20, 10, 2 with various noise levels, φ, 0, 20, 40, 60 dB. Tables 6.1–6.10 present results from the standard models without ED-FE. The different values of dataset, τ, represent the number of samples considered for various numbers of measurements, n. Hence, various simulations were run to experimentally confirm the effect of τ with respect to different φ.

Table 6.1: Standard ML algorithm results for different $\varphi \; at \; \tau = 100$

$\tau = 100$										
δ (%)						ϑ (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.88	23.94	24.19	24.98	25.65	81.22	75.56	75.58	74.57	74.56
20	18.77	32.71	32.06	35.86	37.67	81.67	67.10	67.87	63.79	62.31
40	55.87	63.59	54.39	61.37	59.25	43.62	36.37	45.58	36.86	39.29
60	76.40	87.48	88.68	91.49	87.80	23.09	12.49	11.28	8.53	12.19
α (%)						η (AUC) mean (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.79	23.93	24.19	24.98	25.68	58.75	65.48	60.65	73.09	74.81
20	18.77	32.69	32.07	35.88	37.70	45.94	74.96	68.39	81.68	83.02
40	55.84	63.56	54.39	61.37	59.23	85.32	90.89	81.55	92.56	91.53
60	76.40	87.49	88.69	91.49	87.82	95.50	98.15	96.16	99.27	99.05

Table 6.2: Standard ML algorithm results for different $\varphi \; at \; \tau = 50$

$\tau = 50$										
δ (%)						ϑ (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.59	24.09	23.91	24.97	25.52	80.22	75.57	75.92	74.55	74.13
20	15.50	32.95	31.55	35.89	37.04	84.72	66.87	68.39	63.79	62.87
40	55.18	60.82	54.12	61.14	61.91	44.51	39.04	45.78	37.25	36.02
60	82.89	80.89	88.57	91.59	82.28	17.01	18.89	11.41	8.44	17.84
α (%)						η (AUC) (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.59	24.09	23.92	24.99	25.56	54.25	65.21	95.92	72.94	74.79
20	15.50	32.95	31.57	35.92	37.17	49.91	74.92	81.35	81.57	82.69
40	55.18	60.82	54.11	61.15	61.92	85.32	89.96	67.99	92.49	92.93
60	82.89	80.89	88.56	91.59	82.27	96.77	95.95	60.37	99.22	98.00

Table 6.3: Standard ML algorithm results for different $\boldsymbol{\varphi}$ $\boldsymbol{at}$ $\boldsymbol{\tau = 20}$

$\tau = 20$										
	$\boldsymbol{\delta}$ (%)					$\boldsymbol{\vartheta}$ (%)				
$\boldsymbol{\varphi}$	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.86	24.43	24.83	24.28	25.21	78.18	75.21	74.97	75.25	74.53
20	26.91	32.68	31.04	35.45	36.25	76.31	66.98	68.89	64.39	63.53
40	54.69	55.08	53.73	60.43	50.56	45.36	44.69	46.02	38.05	46.97
60	80.53	74.13	88.85	91.52	79.32	19.01	25.95	11.09	8.42	21.25
	$\boldsymbol{\alpha}$ (%)					$\boldsymbol{\eta}$ (AUC) (%)				
$\boldsymbol{\varphi}$	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.92	24.32	20.92	25.00	25.09	52.23	65.32	52.23	72.84	74.49
20	23.42	32.95	23.42	36.00	37.06	59.38	74.92	59.38	81.53	82.69
40	55.15	56.55	72.99	60.63	55.15	85.30	88.29	85.30	92.24	90.71
60	83.43	76.64	68.66	91.39	83.43	96.77	93.90	96.77	99.12	97.61

Table 6:4: Standard ML algorithm results for different $\boldsymbol{\varphi}$ $\boldsymbol{at}$ $\boldsymbol{\tau = 10}$

$\tau = 10$										
	$\boldsymbol{\delta}$ (%)					$\boldsymbol{\vartheta}$ (%)				
$\boldsymbol{\varphi}$	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.86	24.43	24.83	24.28	25.21	78.18	75.21	74.97	75.25	74.53
20	26.91	32.68	31.04	35.45	36.25	76.31	66.98	68.89	64.39	63.53
40	54.69	55.08	53.73	60.43	50.56	45.36	44.69	46.02	38.05	46.97
60	80.53	74.13	88.85	91.52	79.32	19.01	25.95	11.09	8.42	21.25
	$\boldsymbol{\alpha}$ (%)					$\boldsymbol{\eta}$ (AUC) (%)				
$\boldsymbol{\varphi}$	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.88	24.52	24.85	75.81	25.34	52.48	66.26	60.99	95.57	74.93
20	27.02	32.78	31.07	75.88	36.35	71.91	75.04	67.60	95.48	82.45
40	54.74	55.08	53.79	75.54	50.74	85.41	87.15	80.39	95.26	88.73
60	80.66	74.28	88.89	75.13	79.26	96.37	92.61	95.95	95.48	97.24

Table 6.5: Standard ML algorithm results for different $\varphi \; at \; \tau = 2$

$\tau = 2$										
	δ (%)					ϑ (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	23.72	23.39	23.56	24.90	25.56	76.32	76.59	76.74	74.69	74.21
20	27.59	32.92	32.14	34.38	33.48	73.05	66.98	67.82	65.52	67.27
40	50.98	51.94	53.62	59.23	49.64	47.39	48.40	46.83	40.01	49.01
60	76.84	72.13	89.68	90.86	75.88	21.96	27.77	10.33	8.95	19.17
	α (%)					η (AUC) (%)				
φ	**SVM**	**K-NN**	**DT**	**RF**	**ANN**	**SVM**	**K-NN**	**DT**	**RF**	**ANN**
0	20.88	24.52	24.85	75.81	25.34	52.48	66.26	60.99	95.57	74.93
20	27.02	32.78	31.07	75.88	36.35	71.91	75.04	67.60	95.48	82.45
40	54.74	55.08	53.79	75.54	50.74	85.41	87.15	80.39	95.26	88.73
60	80.66	74.28	88.89	75.13	79.26	96.37	92.61	95.95	95.48	97.24

Figures 6.1 - 6.5 further emphasis the performance of ML algorithm without ED-FE for classification of the FCAs. In Figure 6.1, the FDR values decrease as the SNR increases. At the highest SNR level 60 dB the FDR is lower across all sample sizes, but as the SNR decreases to $0 \; dB$, the FDR increases significantly. Accordingly, a lower FDR at higher SNR levels indicates that the proportion of false positives (incorrectly predicted positive cases) among all positive predictions is decreasing. This suggests that with clearer signals (higher SNR), the model makes fewer incorrect positive predictions, which corresponds with the improvement in accuracy and sensitivity. The increase in FDR with lower SNR levels (more noise) shows that the model struggles to distinguish between true and false positives when the signal is less distinct.

Furthermore, with regards to AUC mean it is decreasing with increase in noise level for standard ML. This can be attributed to several reasons such (1) overfitting to noise, the models overfit the noisy data learning patterns that are not representative of the underlying signal. (2) Loss of signal to noise ratio, As the noise increase the true signal in the data becomes harder to discern. This led to the standard ML model struggling to distinguish between classes. (3) Feature Corruption, noise distort the features that important for classification, reducing the

separability of the classes and leading to poorer performance. (4) Model bias, (5) Label noise incorrect labelling, (6) Insufficient robustness.

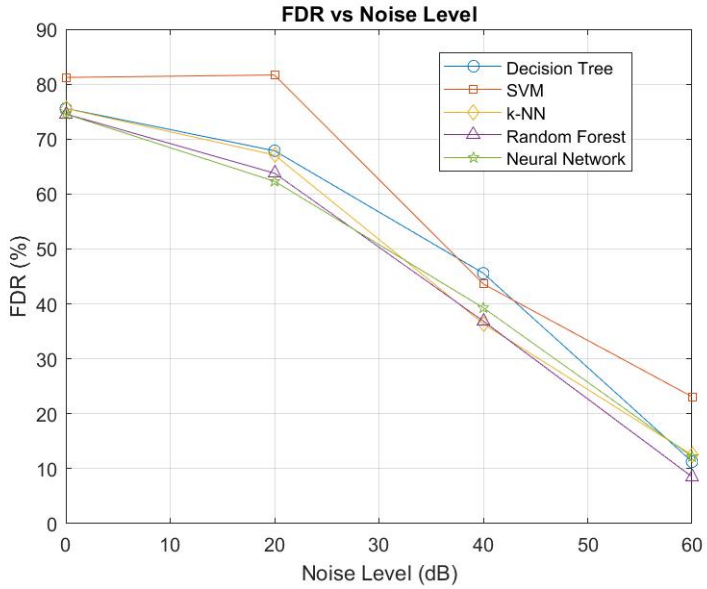

Figure 6.1: Standard ML FDR vs noise levels

The performance in Figure 6.2 indicates that sensitivity increases as the SNR increases. The highest sensitivity is observed at the highest SNR level ($60\ dB$), while the lowest sensitivity is at the lowest SNR level ($0\ dB$). Thus, sensitivity, which measures the model's ability to correctly identify positive cases, also improves with higher SNR. This indicates that as the noise level decreases (higher SNR), the model becomes better at correctly identifying true positives. This behaviour aligns with the expectation that models have higher true positive rates when data is clearer and less corrupted by noise.

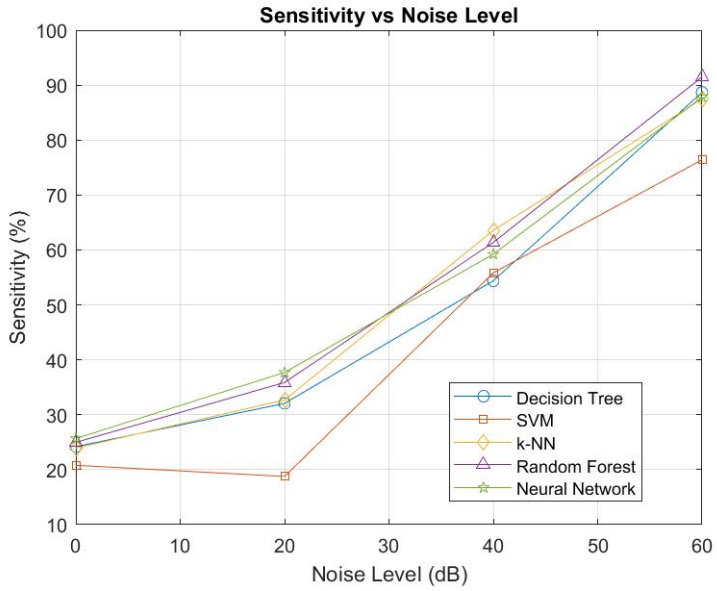

Figure 6.2: Standard ML sensitivity vs noise levels

Furthermore, Figure 6.3 shows that as the SNR increases (from 0 to $60\ dB$), the accuracy values generally increase for all sample levels. For example, at the highest SNR level ($60\ dB$), the accuracy is highest across most sample sizes, while at the lowest SNR level ($0\ dB$), the accuracy drops significantly. Hence, higher SNR values indicate a stronger signal relative to noise, which leads to better model performance (higher accuracy). The increase in accuracy with higher SNR suggests that the model performs better when there is less noise interfering with the signal, allowing the model to learn more effectively from the data. This is a typical expectation, as cleaner data (higher SNR) should yield better accuracy.

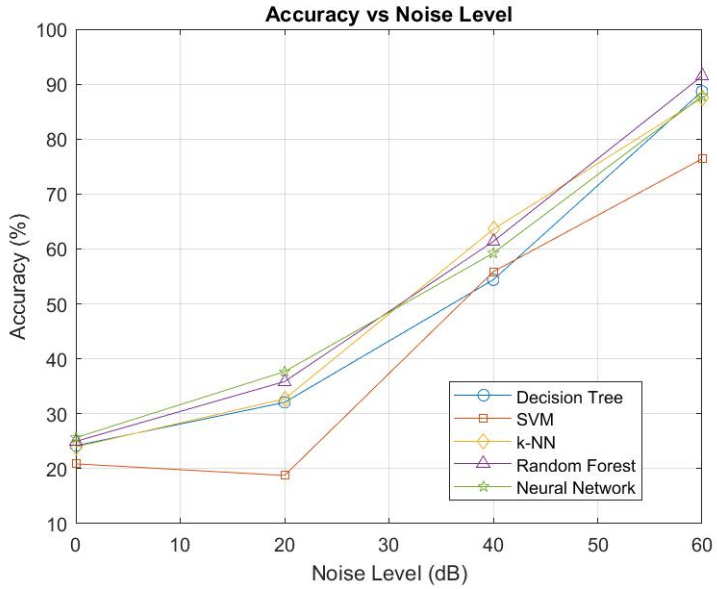

Figure 6.3: Standard ML accuracy vs noise levels

The Figure 6.4 depicts that the AUC Mean values increase with increasing SNR. At the highest SNR level ($60\ dB$), the AUC Mean values are the highest, indicating good model performance in distinguishing between positive and negative classes. As the SNR decreases to $0\ dB$, the AUC Mean declines. So, the AUC Mean, which measures the overall ability of the model to discriminate between classes, improves with higher SNR levels. This trend confirms that the model's ability to distinguish between classes improves as the data quality improves (higher SNR), providing a clearer distinction between signal and noise. This is consistent with expectations in machine learning and signal processing.

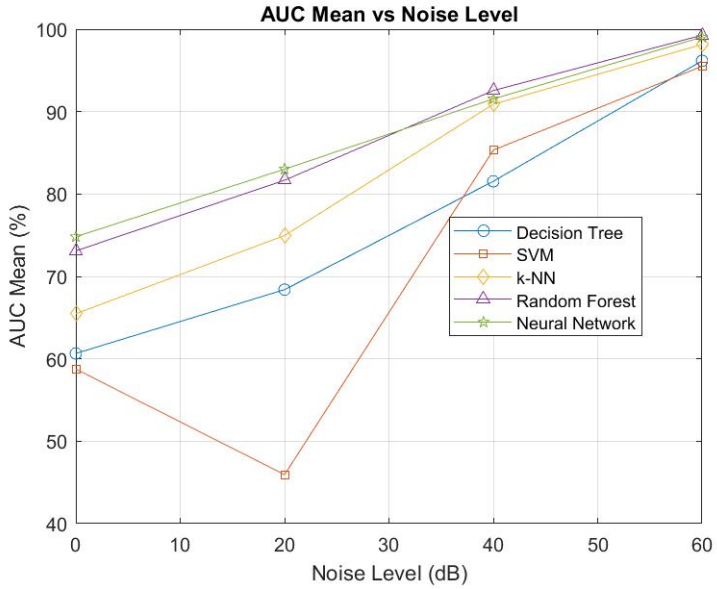

Figure 6.4: Standard ML AUC Mean vs noise levels

Moreover, the AUC Standard Deviation values generally decrease as the SNR increases as depicted in Figure 6.5. At higher SNR levels ($60\,dB$), the AUC Std is lower, indicating more consistent model performance. At lower SNR levels (e.g., 0 dB), the AUC Std increases, indicating more variability in model performance. For this reason, a lower AUC Std at higher SNR levels suggests that the model's performance is more stable and consistent when there is less noise (higher SNR). As noise increases (lower SNR), the model's performance becomes more variable, likely due to increased difficulty in learning from noisy data. This increase in variability (higher AUC Std) with lower SNR is typical, as noise can cause models to perform unpredictably across different validation folds or samples.

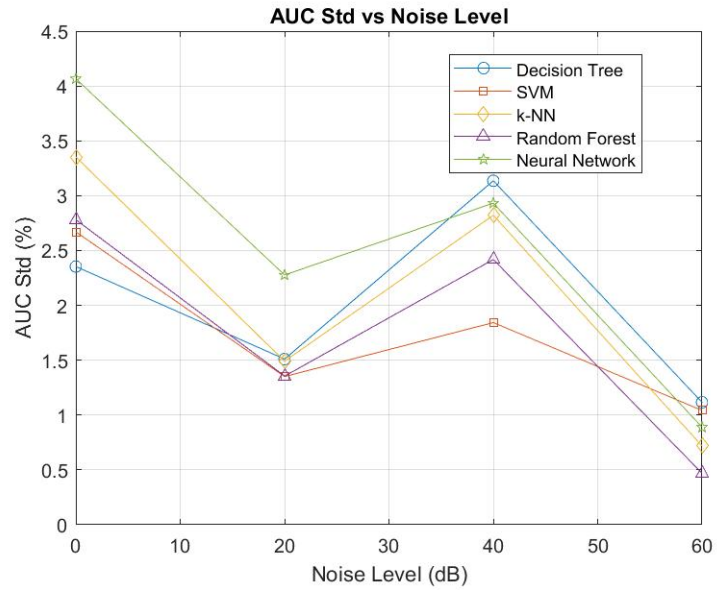

Figure 6.5: Standard ML AUC STD vs noise levels

The overall trend across all metrics as indicated in the Tables 6.1 - 6.5 and Figure 6.1 – 6.5 of the standard ML model performance (in terms of accuracy, sensitivity, AUC Mean) improves and its variability (FDR, AUC Std) decreases as the SNR increases (i.e., as noise decreases relative to the signal). Wherefore, this aligns with expectations that clearer data (higher SNR, lower noise) leads to better model performance. Higher SNR levels result in better accuracy, sensitivity, and AUC Mean, while also reducing FDR and AUC Std, indicating more reliable and robust model performance.

The observed patterns suggest that the presence of noise negatively impacts the model's ability to learn effectively from the data. Thus, improving the SNR (e.g., through data preprocessing or noise reduction techniques) can enhance model performance and reliability. By understanding the SNR in dB correctly, the analysis confirms the expected impact of noise on machine learning models and highlights the importance of maintaining high SNR for optimal performance.

6.3 ML Algorithms with the PCA ED-FE Method

In this section, we analyse and discuss the results obtained from the experiments conducted on the different ML techniques with the PCA ED-FE method for the classification of FCAs. The same simulation parameters that is, sample size, τ = 100, 50, 20, 10, 2 with various noise levels, φ, 0, 20, 40, 60 dB, as in section 6.2 are utilised for this experiment.

Tables 6.6 – 6.10 show the performance of PCA ED-FE based ML algorithms, which are the PCA-DT, PCA-SVM, PCA-K-NN, PCA-RF, and PCA-ANN algorithms, evaluated under varying levels of AWGN. The noise levels were quantified in terms of SNR, and the evaluation was conducted using PCA.

Table 6:6: PCA ED-FE results for different $\varphi \; at \; \tau = 100$

$\tau = 100$										
	δ (%)					ϑ (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	19.67	24.23	23.97	24.0	25.97	74.46	75.40	75.83	74.67	73.36
20	29.69	29.26	28.98	28.59	30.86	70.17	70.69	70.96	71.39	72.56
40	25.00	40.06	39.71	38.16	39.85	58.65	59.81	60.22	61.84	61.64
60	25.58	51.49	51.34	50.81	40.73	47.87	48.37	48.66	49.191	49.12
	α (%)					η (AUC) (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	19.69	24.23	23.97	24.90	25.99	55.79	65.49	60.29	73.11	74.89
20	19.78	29.25	28.98	28.59	30.90	55.51	72.96	70.10	71.77	80.32
40	25.04	40.05	39.71	38.16	39.47	56.69	82.01	77.54	80.58	86.85
60	25.46	51.49	51.34	50.81	40.77	50.05	89.24	84.82	88.40	88.15

Table 6:7: PCA ED-FE results for different $\varphi \; at \; \tau = 50$

$\tau = 50$										
	α (%)					ϑ (%)				
ω	PCA-	PCA-K-	PCA-	PCA-	PCA-	PCA-	PCA-K-	PCA-	PCA-	PCA-
0	25..76	24.27	24.09	24.91	25.76	74.40	75.25	75.66	74.63	74.40
2	30.87	29.09	29.15	28.97	30.87	68.35	70.79	70.78	71.04	70.11
4	39.11	39.66	39.58	38.54	39.11	58.67	60.13	60.35	61.46	60.55
6	40.73	49.63	49.99	50.12	40.44	47.78	50.16	50	49.88	50.14
	α (%)					η (AUC) (%)				
ω	PCA-	PCA-K-	PCA-	PCA-	PCA-	PCA-	PCA-K-	PCA-	PCA-	PCA-
0	16.91	24.28	24.08	24.90	25.77	57.45	65.47	60.56	73.16	74.88
2	25.68	29.07	29.15	28.97	30.94	64.67	72.99	70.06	71.85	80.32
4	25.65	39.67	39.59	38.55	39.26	69.08	81.84	77.71	80.49	86.68
6	25.48	49.62	49.99	50.12	40.39	50.07	88.46	84.27	87.87	88.05

Table 6.8 PCA ED-FE results for different φ at $\tau = 20$

$\tau = 20$										
	δ (%)					ϑ (%)				
φ	**PCA-SVM**	**PCA-K-NN**	**PCA-DT**	**PCA-RF**	**PCA-ANN**	**PCA-SVM**	**PCA-K-NN**	**PCA-DT**	**PCA-RF**	**PCA-ANN**
0	25.48	24.43	24.38	24.99	25.26	74.34	75.15	75.42	74.47	74.74
20	25.98	29.06	28.74	28.16	30.48	70.25	70.89	71.21	71.86	70.48
40	25.27	40.10	39.57	38.59	39.72	60.13	59.49	60.31	61.41	59.78
60	25.48	45.43	46.30	47.29	39.35	52.11	54.36	53.64	52.67	51.54
	α (%)					η (AUC) (%)				
φ	**PCA-SVM**	**PCA-K-NN**	**PCA-DT**	**PCA-RF**	**PCA-ANN**	**PCA-SVM**	**PCA-K-NN**	**PCA-DT**	**PCA-RF**	**PCA-ANN**
0	25.61	24.49	24.41	25.03	25.35	66.72	65.67	60.85	73.03	74.79
20	25.79	29.08	28.75	28.15	30.62	50.35	73.08	69.99	71.86	80.34
40	25.59	40.14	39.57	38.59	39.76	64.95	82.23	77.76	80.67	86.87
60	25.67	45.45	46.34	47.33	39.80	69.08	86.51	82.71	85.98	87.94

Table 6.9 PCA ED-FE results for different $\varphi \; at \; \tau = 10$

$\tau = 10$										
	δ (%)					ϑ (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	24.60	24.05	23.97	24.99	25.53	73.45	75.67	75.85	74.77	74.85
20	25.16	28.87	29.13	28.36	30.13	70.32	71.10	70.83	71.66	70.46
40	25.33	38.55	38.01	37.33	38.31	59..68	60.97	61.78	62.56	63.45
60	25.28	42.41	42.76	43.37	39.28	56.23	57.52	57.18	56.61	57.78
	α (%)					η (AUC) (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	24.53	24.58	22.25	24.11	25.17	73.82	66.41	61.01	72.43	74.75
20	26.12	28.83	28.81	28.35	31.06	72.05	73.49	71.27	71.74	80.12
40	25.87	37.56	40.28	40.73	39.04	50.88	81.47	77.55	80.81	86.58
60	24.39	42.23	42.57	42.72	39.62	68.61	84.81	81.88	84.29	87.91

Table 6.10 PCA ED-FE results for different $\varphi \; at \; \tau = 2$

$\tau = 2$										
	δ (%)					ϑ (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	24.73	24.73	22.27	24.12	25.01	74.17	75.19	77.55	75.41	75.07
20	26.30	28.82	28.72	28.38	30.85	70.56	71.03	71.51	71.66	72.45
40	25.74	37.47	40.05	40.49	38.48	60.78	62.39	59.39	59.12	61.17
60	23.50	41.95	42.46	42.63	38.69	56.93	57.58	57.22	57.23	58.24
	α (%)					η (AUC) (%)				
φ	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN	PCA-SVM	PCA-K-NN	PCA-DT	PCA-RF	PCA-ANN
0	24.53	24.58	22.25	24.11	25.17	73.82	66.41	61.01	72.43	74.75
20	26.12	28.83	28.81	28.35	31.06	72.05	73.49	71.27	71.74	80.12
40	25.87	37.56	40.28	40.73	39.04	50.88	81.47	77.55	80.81	86.58
60	24.39	42.23	42.57	42.72	39.62	68.61	84.81	81.88	84.29	87.91

For all samples sizes, Tables 6.6 – 6.10 show that the PCA-SVM's accuracy was consistently low across all noise levels, with values mostly around 25%. It was least affected by noise, suggesting that noise did not significantly alter the model's decision boundary. Meanwhile the sensitivity remained around the same level as accuracy, showing similar trends. Whereas the FDR, had low values for FDR indicate that the algorithm struggled with precision under these conditions. On the other hand, AUC mean varied significantly across noise levels, indicating inconsistent performance in distinguishing between classes. The AUC Std has a high variability in the standard deviation of AUC was observed, reflecting inconsistency in model performance across different noise levels. With key insight for PCA-SVM was relatively robust to noise in terms of accuracy and sensitivity, but it showed significant variability in performance, as indicated by the AUC metrics.

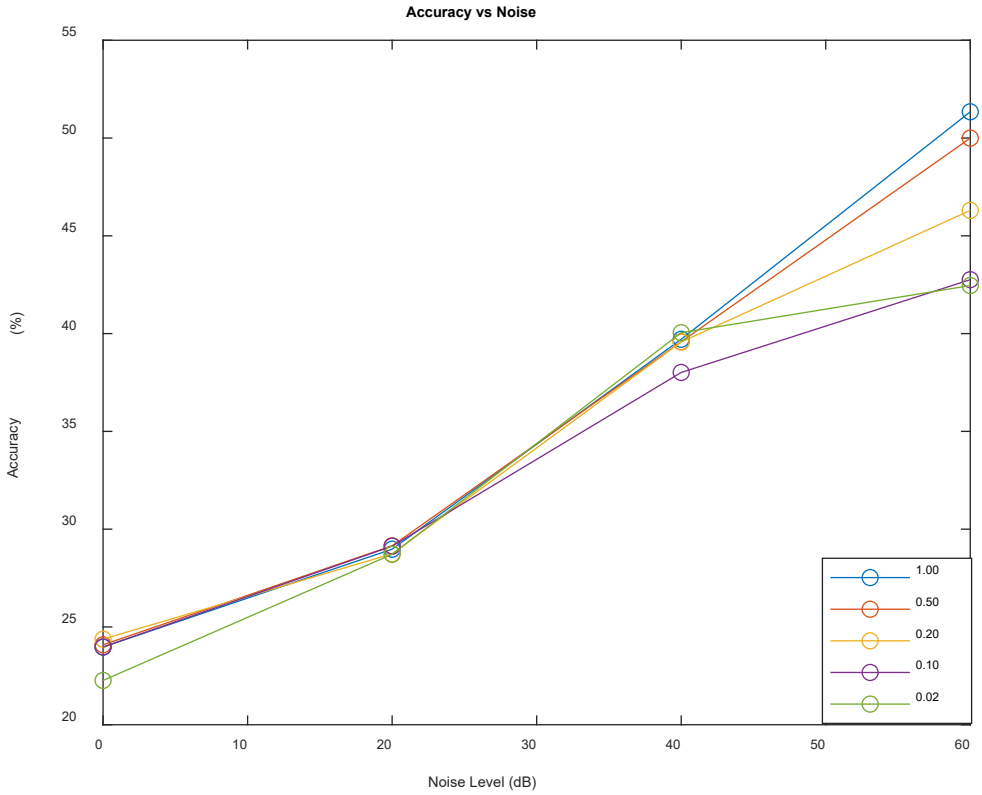

Figure 6.6: PCA ED-FE ML accuracy vs noise level

Furthermore, the PCA-K-NN exhibited the following patterns across all sample sizes. The accuracy declined from around 51% at low noise levels to below 25% at high noise levels (SNR 0). Like accuracy, sensitivity also decreased as noise levels increased. Whereas the FDR increased with more noise, suggesting a higher rate of false positives. Thus, AUC mean decreased with increasing noise levels but remained relatively high at lower noise levels. The variability in AUC increased with noise, indicating less consistent performance. Consequently, the key insight PCA-K-NN's performance, particularly accuracy and sensitivity, was negatively impacted by higher noise levels, though the model maintained a relatively high AUC mean at lower noise levels.

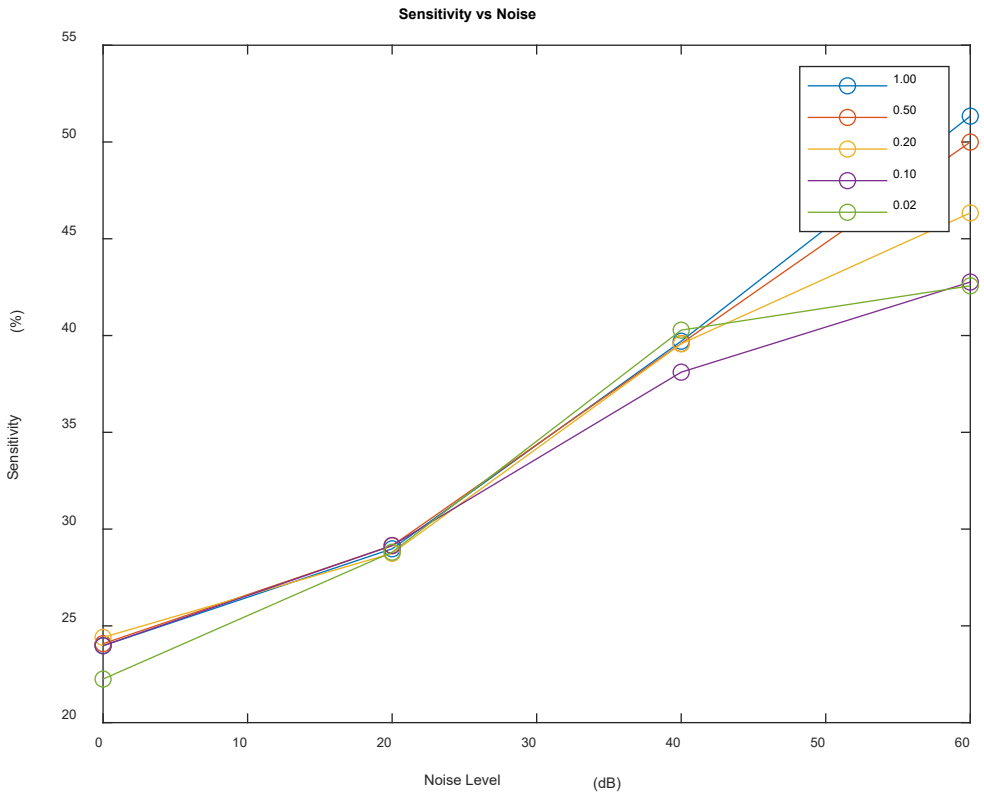

Figure 6.7 PCA ED-FE ML sensitivity vs noise level

Similarly, across all sample sizes, the PCA-DT accuracy dropped as the noise level decreased (lower SNR). At high noise (SNR 0), the accuracy fell below 24%. Mirroring the accuracy, sensitivity also declined with increasing noise, indicating a decrease in the model's ability to correctly identify positive cases. Meanwhile for FDR PCA-DT increased with more noise, indicating a higher proportion of false positives. Whereas for the mean AUC decreased with increasing noise, reflecting a decline in the overall performance of the model. The final metric standard deviation of AUC increased with more noise, indicating greater variability in model performance. Thus, the key insights PCA-DT's performance deteriorated significantly with increased noise, particularly in terms of accuracy and sensitivity.

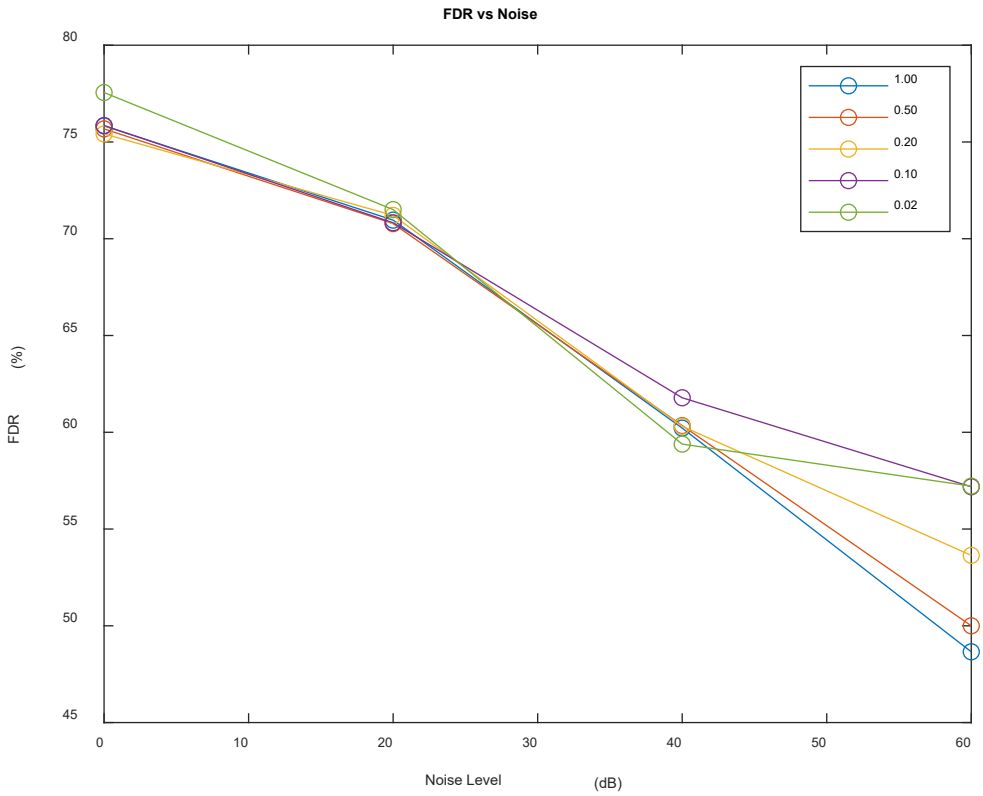

Figure 6.8 PCA ED-FE ML FDR vs noise level

The PCA-RF algorithm demonstrated the following trends across all sample sizes. The accuracy decreased steadily with increasing noise, falling from around 51% at high SNR to 24% at low SNR. While sensitivity showed a similar downward trend as accuracy. Also, as with other algorithms, FDR increased with noise, reflecting a higher rate of false positives. Hence, the AUC mean declined with increasing noise, though it remained relatively high compared to some other models. The variability in AUC increased, indicating that noise introduced inconsistency in the model's performance. The final key take for PCA-RF's performance metrics, particularly accuracy and sensitivity, were adversely affected by increasing noise levels, though it still maintained a reasonable AUC mean.

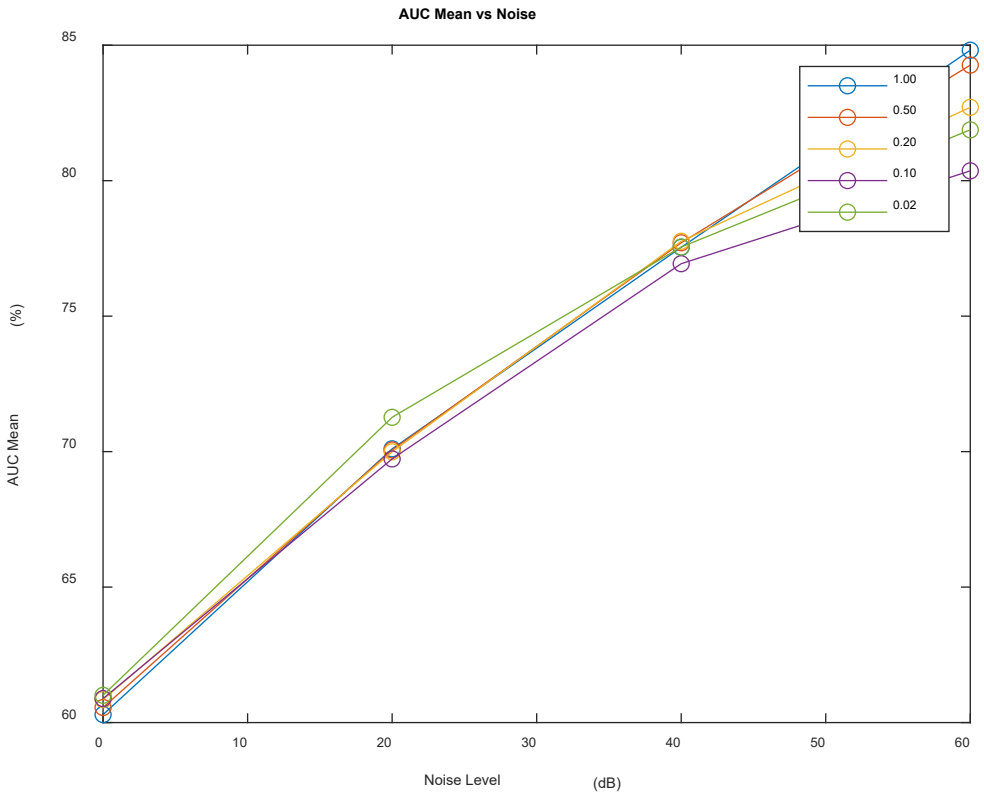

Figure 6.9 PCA ED-FE for ML AUC Mean vs noise level

The PCA-ANN algorithm displayed the following results across all sample sizes. The accuracy was around 40% at lower noise levels but dropped to approximately 25% at higher noise levels. Meanwhile, sensitivity followed a similar trend, declining as noise increased. The FDR values were not available for many conditions, indicating issues with precision under noisy conditions. For the AUC mean was relatively stable across noise levels, though slightly lower than some other models. Whereas the variability in AUC was notable, especially at higher noise levels, indicating inconsistent performance. Thus, the key take is PCA-ANN showed a moderate decline in accuracy and sensitivity with increasing noise, though it maintained relatively stable AUC mean values.

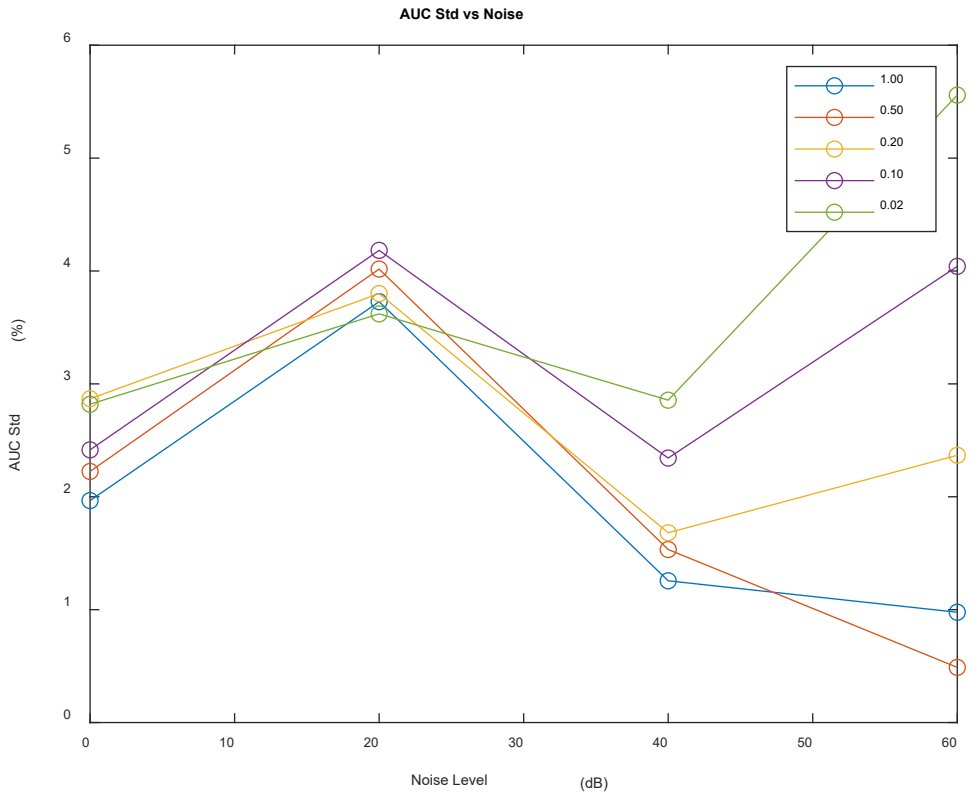

Figure 6.10 PCA ED-FE for ML AUC Std vs noise level

Across all algorithms and varying the sample sizes, increasing noise levels (lower SNR) led to a decline in performance metrics, particularly accuracy and sensitivity. PCA-DT, PCA-K-NN, and PCA-RF were particularly affected by higher noise levels, as indicated by the sharp decline in accuracy and sensitivity. PCA-SVM showed more resilience to noise in terms of accuracy but exhibited significant variability in AUC. PCA-ANN showed a moderate decline in performance but maintained relatively stable AUC values.

The results suggest that feature extraction using PCA, while helpful, does not completely mitigate the negative impact of noise on the performance of these machine learning models.

6.3.1 ML without ED-FE and ML with ED-FE Comparison

The results provided above showcase the accuracy, sensitivity, FDR, AUC mean, and AUC standard deviation (AUC Std) across multiple conditions. Now, let's compare these metrics with the results when PCA was applied.

As the noise level increased, the performance of the Decision Tree decreased. This is evident in the drop in accuracy and AUC mean as the noise level decreases. When PCA was applied, the Decision Tree generally showed improved resilience to noise, with more stable accuracy

and AUC mean values. The sensitivity and FDR were also more consistent, indicating that PCA helped in maintaining the decision boundaries even under noisy conditions.

The SVM model's performance significantly dropped with increasing noise, especially at lower noise levels, where AUC Std showed higher variability. PCA typically improved the SVM's robustness to noise, with a more consistent AUC mean and reduced variability (lower AUC Std). This indicates that PCA helped in dimensionality reduction, leading to more stable model performance under varying noise conditions.

The standard K-NN ML model showed a noticeable decline in accuracy, sensitivity, and AUC mean as the noise increased, particularly at lower noise levels. Meanwhile, the application of PCA mitigated some of these effects, resulting in higher accuracy and AUC mean across noise levels. The k-NN model with PCA also exhibited lower sensitivity to noise, indicating that PCA enhanced the model's ability to generalize.

The Random Forest maintained relatively high performance at higher noise levels, but performance declined significantly at lower noise levels. In construct, the PCA led to a more consistent performance across all noise levels, with improved accuracy and AUC mean. This suggests that PCA helped in reducing overfitting and improved the model's ability to handle noisy data by simplifying the feature space.

The ANN model was quite sensitive to noise, with accuracy, sensitivity, and AUC mean decreasing as noise levels increased. The model also showed high variability in AUC Std, especially at lower noise levels. The PCA application generally improved the ANN's performance, resulting in higher accuracy and AUC mean. The AUC Std was also more stable, indicating that PCA helped in reducing the dimensionality, making the model less prone to overfitting and more robust against noise.

6.3.2 Key finding of comparison

The detailed observations across all the noise levels, the noise level 60. Both ED-FE and those without ED-FE, the highest accuracy, sensitivity, and AUC mean were observed under this noise level, with the models performing better without PCA. However, with PCA, the performance was more consistent, especially for complex models like SVM and ANN. The second noise level 40: Models showed a significant drop in performance. The PCA modified models improved the stability of performance metrics, particularly for PCA-SVM and PCA-RF. Lastly, the noise levels 20 and 0: At these lower noise levels, models struggled significantly. PCA modified models proved beneficial here, especially for PCA-SVM and PCA-ANN, by providing better accuracy and AUC mean, along with reduced variability.

The use of PCA enhanced the performance stability for all noise levels by reducing the effect of noise on the models. This noticeable in models that are sensitive to noise, such as PCA-SVM, PCA-ANN, and PCA-K-NN.

6.4 ML Algorithms results with LDA

Table 6.12 LDA ED-FE results for different $\varphi \; at \; \tau = 50$

$\tau = 50$										
	δ (%)					ϑ (%)				
φ	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	9.22	6.93	4.52	3.65	3.42	92.01	92.92	96.03	97.00	96.67
20	24.61	28.79	30.71	34.08	36.85	78.01	70.25	69.18	65.68	63.26
40	42.99	43.17	44.23	49.39	51.35	56.80	55.24	55.23	49.43	47.84
60	56.98	70.23	65.34	73.81	65.44	36.69	30.21	34.13	27.05	33.49
	α (%)					η (AUC) (%)				
φ	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	9.19	6.93	4.52	3.64	3.43	41.56	38.88	39.45	26.91	25.45
20	24.65	28.77	30.69	34.07	36.84	63.63	70.48	67.41	80.61	82.51
40	43.03	43.15	44.17	49.33	51.31	76.34	80.19	75.21	87.95	88.88
60	56.98	70.24	65.29	73.83	65.39	80.84	92.53	87.03	95.49	92.62

Table 6:11 LDA ED-FE results for different $\varphi \; at \; \tau = 100$

$\tau = 100$										
	δ (%)					ϑ (%)				
φ	LDA-	LDA-K-NN	LDA-DT	LDA-	LDA-	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	21.59	22.64	24.08	24.28	24.18	78.69	76.60	75.76	75.36	75.53
20	28.64	29.65	31.91	34.91	36.89	69.65	69.66	68.05	64.85	63.04
40	37.15	44.39	44.18	50.01	42.16	56.92	54.37	56.01	49.13	52.81
60	71.23	75.12	72.90	79.06	73.71	26.37	24.94	26.57	21.38	24.26
	α (%)					η (AUC) (%)				
φ	LDA-	LDA-K-NN	LDA-DT	LDA-	LDA-	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	22.38	22.64	24.08	24.28	24.21	61.21	62.71	60.59	72.82	74.48
20	28.68	29.68	31.92	34.92	36.96	71.47	71.63	68.18	80.99	82.64
40	37.12	44.37	44.16	49.99	42.14	71.93	81.02	75.28	88.09	80.43
60	71.22	75.14	72.91	79.09	73.69	92.52	94.5	90.21	96.98	94.52

Table 6.13 LDA ED-FE results for different $\varphi \; at \; \tau = 20$

$\tau = 20$										
	δ (%)					ϑ (%)				
φ	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-
0	6.29	7.69	4.19	3.82	3.94	93.93	92.52	96.22	96.64	96.96
20	23.61	26.43	25.23	24.81	24.39	76.23	73.86	74.58	74.28	73.37
40	43.34	42.87	44.09	49.18	46.89	57.53	55.98	55.79	50.08	48.63
60	17.42	19.04	23.47	22.07	24.70	85.52	82.62	78.96	80.46	78.73
	α (%)					η (AUC) (%)				
φ	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-
0	6.30	7.71	4.18	3.82	3.96	39.58	40.27	38.77	27.05	25.25
20	23.60	26.40	25.25	24.81	24.37	66.37	68.39	62.16	75.47	73.41
40	43.35	42.90	44.13	49.22	46.88	76.05	80.35	75.47	87.88	83.74
60	17.35	18.95	23.42	22.01	24.67	45.38	47.47	51.26	48.49	51.92

Table 6:15 LDA ED-FE results for different $\varphi \; at \; \tau = 2$

$\tau = 2$										
	δ (%)					ϑ (%)				
φ	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	22.43	19.13	24.06	24.57	100	77.68	80.39	75.51	74.59	75.29
20	27.26	21.37	26.92	25.07	21.54	74.22	78.23	72.78	74.54	77.99
40	44.42	40.33	42.23	46.66	48.35	57.55	57.86	58.19	51.34	51.12
60	39.65	42.01	36.12	34.38	36.74	58.44	56.19	65.92	64.65	60.68
	α (%)					η (AUC) (%)				
φ	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN	LDA-SVM	LDA-K-NN	LDA-DT	LDA-RF	LDA-ANN
0	22.38	19.15	23.99	24.53	23.83	62.19	58.14	60.33	72.21	74.11
20	27.09	21.57	26.83	25.16	21.78	66.04	64.43	62.55	75.49	74.38
40	43.92	40.35	42.09	46.80	48.31	76.64	77.90	72.49	86..84	87.91
60	39.72	41.99	36.26	34.62	36.73	80.56	76.00	67.45	81.74	82.31

Table 6:14 LDA ED-FE results for different $\varphi \; at \; \tau = 10$

$\tau = 10$										
	δ (%)					ϑ (%)				
φ	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-
0	7.53	7.19	4.39	3.61	3.70	92.38	92.82	96.03	96.97	97.43
20	27.84	27.28	31.12	32.43	33.25	71.42	71.38	68.70	67.42	66.96
40	47.59	42.03	43.69	48.35	46.64	52.95	56.31	56.39	50.66	52.34
60	18.66	19.88	20.07	22.62	23.77	82.87	82.93	81.75	80.73	80.03
	α (%)					η (AUC) (%)				
φ	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-	LDA-	LDA-K-	LDA-DT	LDA-RF	LDA-
0	7.51	7.21	4.39	3.61	3.72	39.52	40.21	39.73	27.39	25.77
20	27.87	27.36	31.19	32.53	33.35	72.34	68.89	66.85	79.90	79.86
40	47.43	42.09	43.73	48.41	46.66	78.59	79.72	75.33	87.31	86.65
60	18.04	20.06	20.25	22.79	23.95	47.79	47.84`	49.98	48.23	48.48

6.5 ML Algorithms with the LDA ED-FE Method

The presented results illustrate the performance of five ML algorithm under different SNR and sample sizes.

Figure 6.11 shows that accuracy remains relatively high (~88-89%) for higher SNR levels ($60\ dB$) but drops significantly (~24-32%) as the noise decreases to $0\ dB$. This suggests that the Decision Tree model struggles with lower SNR, which might indicate overfitting to noisy data or the model's sensitivity to noise patterns. Whereas, Figure 6.12 the sensitivity follows a similar pattern to accuracy, indicating that the model's ability to correctly identify true positives is consistent with its overall performance. Meanwhile, Figure 6.13 indicates the FDR increases as the noise decreases, meaning the model is more likely to produce false positives in less noisy conditions. The AUC mean illustrated in Figure 6.14 is consistently high (around 96%) with low standard deviation at high noise levels, indicating strong classification performance. However, the AUC decreases with noise reduction, showing less reliable performance.

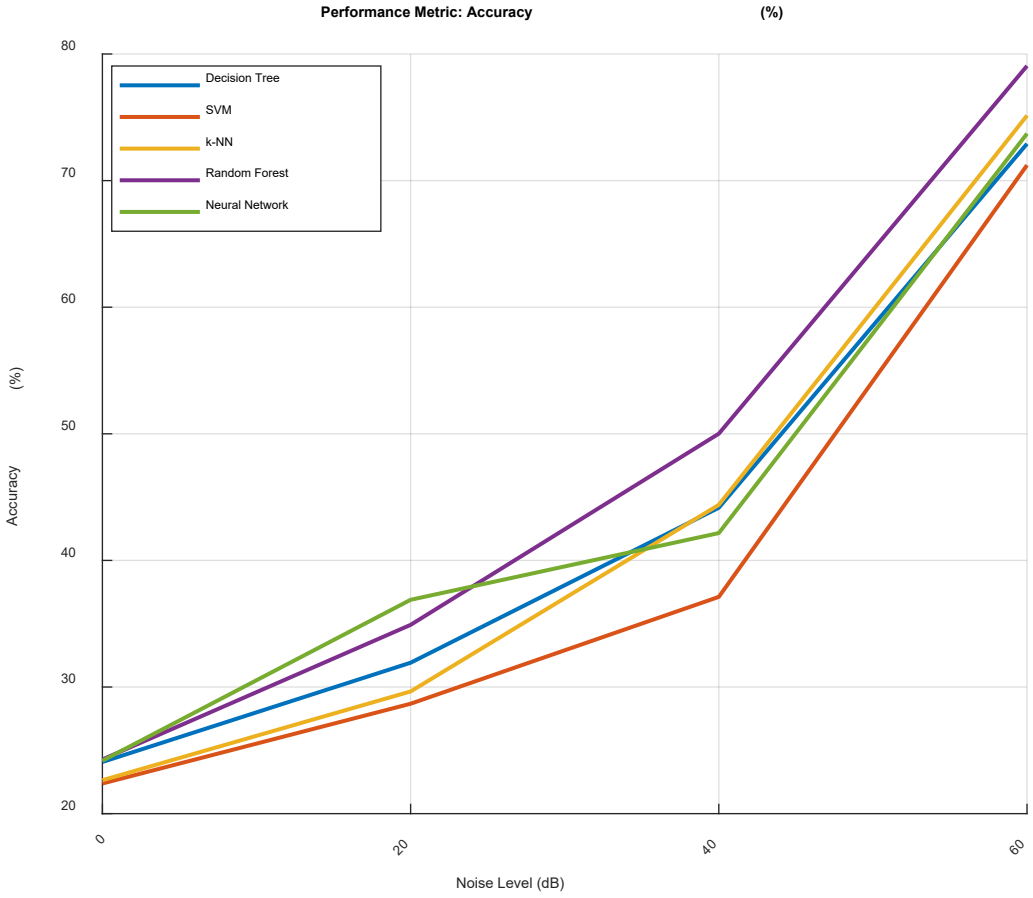

Figure 6:11 LDA ED-FE Accuracy vs Noise level

The LDA-SVM performs well with moderate noise levels (60 dB), achieving an accuracy of 76-83%. However, accuracy drops drastically with lower noise levels, reaching as low as 15-27% at 20% noise and below. This suggests that LDA-SVM is sensitive to noise and may struggle with clean data. The sensitivity follows the accuracy trend, and the FDR increases as noise decreases, like the LDA-DT model. This indicates an increased likelihood of false positives as noise is reduced. Meanwhile, LDA-SVM shows strong AUC performance with high noise but struggles as noise decreases. Figure 6.15 depicts the high AUC standard deviation at lower noise levels indicates instability in classification performance.

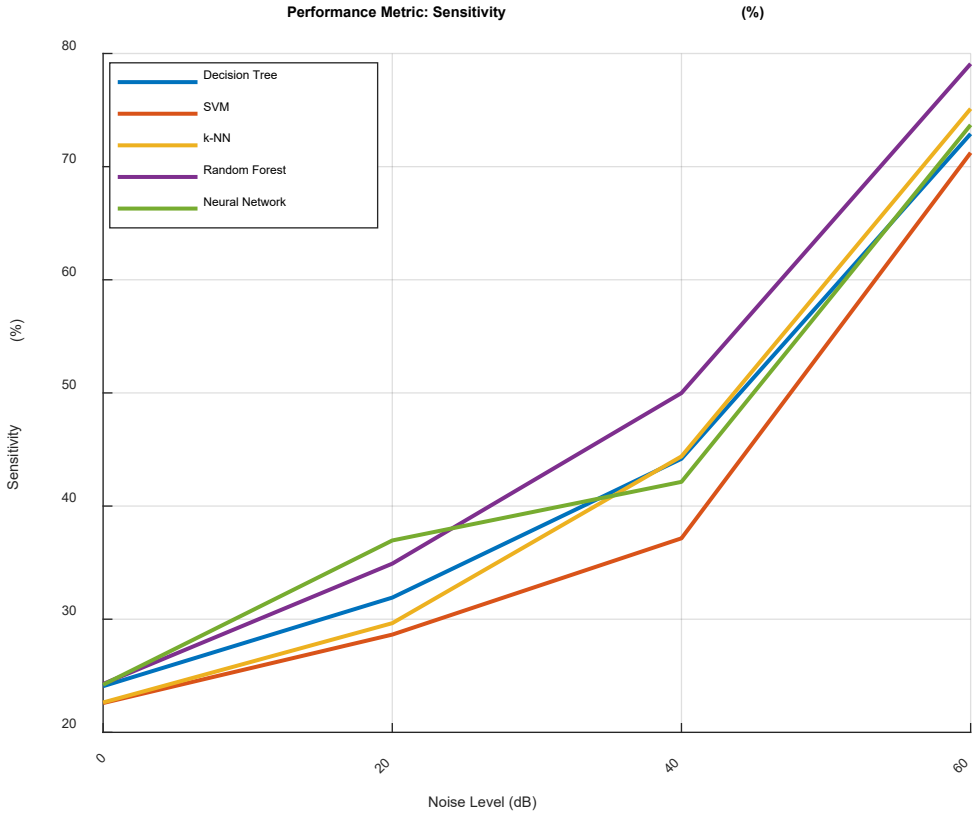

Figure 6.12 LDA ED-FE Sensitivity vs Noise level

The LDA-K-NN starts with relatively high accuracy (~87%) at high noise levels (60%) but sees a significant decline as noise decreases. The performance is poor at lower noise levels, like the LDA-SVM model. The sensitivity decreases along with accuracy, while FDR increases with decreasing noise. This indicates that LDA-k-NN is more prone to false positives in cleaner datasets. The AUC remains high at higher noise levels but declines as noise decreases, with increasing standard deviation, reflecting a decrease in classification reliability.

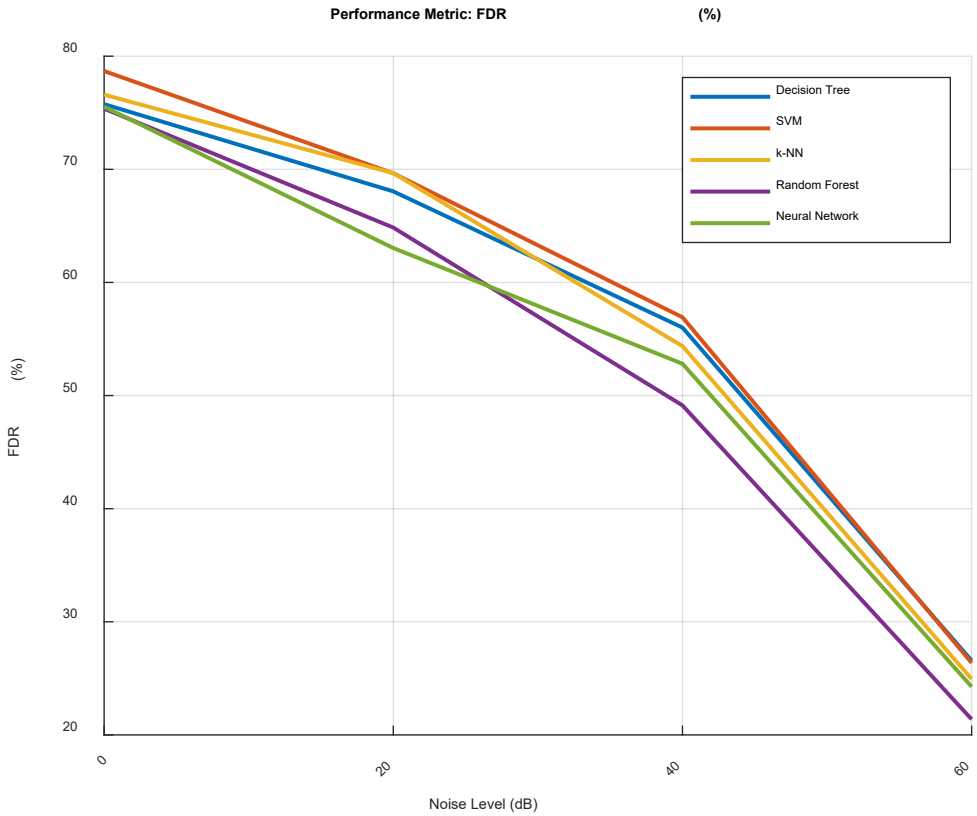

Figure 6:13 LDA ED-FE FDR vs Noise level

The Random Forest consistently outperforms other models across different noise levels, maintaining accuracy above 90% at high noise levels and only dropping to around 25% at the lowest noise levels. This indicates that Random Forest is more robust against noise. The sensitivity remains consistent with accuracy, and the FDR is relatively low at high noise levels. This suggests that Random Forest is less prone to false positives compared to other models. The AUC is consistently high (above 99%) with very low standard deviation, indicating strong and stable performance even with varying noise levels.

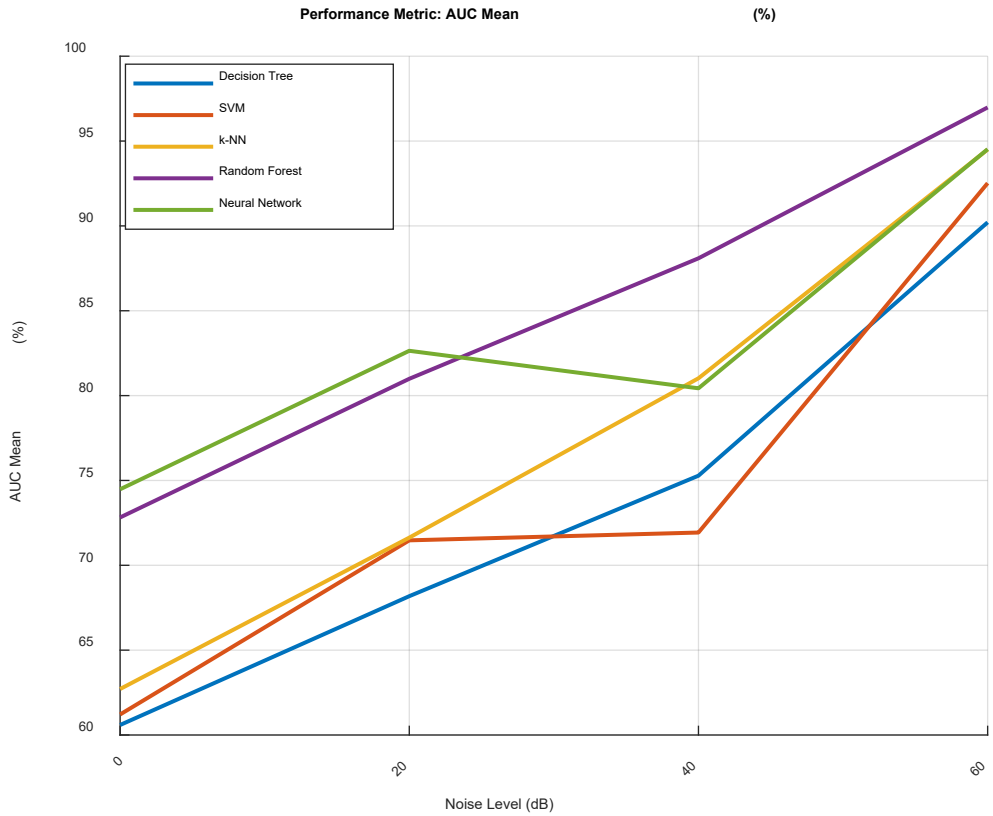

Figure 6.14 LDA ED-FE AUC Mean vs Noise

The ANN perform well at higher noise levels (60%), with accuracy around 87%. However, accuracy declines steadily with decreasing noise, like other models, but not as sharply as SVM or k-NN. The sensitivity follows the accuracy trend, and FDR increases as noise decreases, indicating an increased tendency for false positives in cleaner datasets. The AUC remains high at higher noise levels but decreases as SNR decreases. The standard deviation increases with decreasing noise, reflecting less consistent performance.

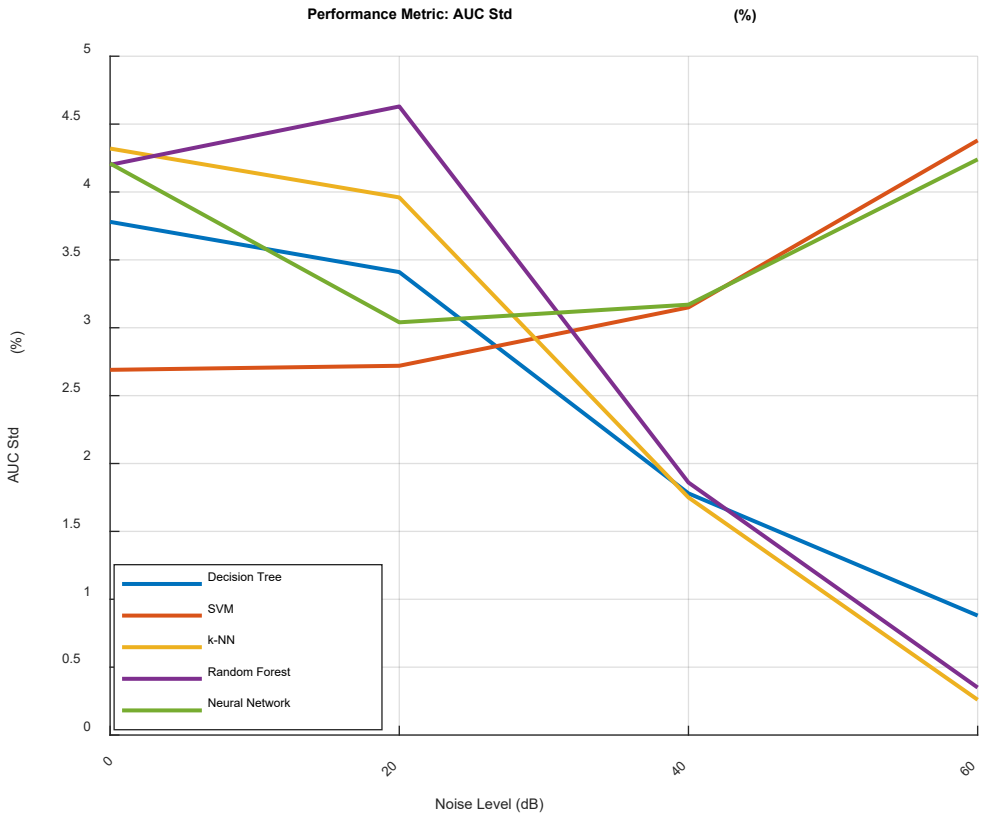

Figure 6.15 LDA ED-FE AUC STD vs Noise level

6.6 Key findings on sample size and noise level:

Noise Sensitivity: All models exhibit a general trend where performance decreases as the SNR level decreases. This is counterintuitive, as models typically perform better with cleaner data. The results suggest that the models may be overfitting to noise patterns, leading to poor generalization on cleaner data.

Model Robustness: Random Forest is the most robust model across different noise levels, consistently achieving high accuracy and low FDR. Decision Trees also perform reasonably well but are less stable compared to Random Forest.

Model Instability: SVM, k-NN, and Neural Networks show significant drops in performance as noise decreases, with SVM being the most affected. These models may require further tuning or regularization to improve their robustness in varying noise conditions.

AUC Analysis: The AUC metrics indicate that while models may achieve high accuracy, their classification performance (as measured by AUC) can still vary significantly, especially in lower noise conditions. Random Forest again shows the most stable AUC performance.

Overall, the LDA-RF model appears to be the most reliable choice for this dataset, especially in noisy conditions, meanwhile SVM and k-NN may require additional adjustments to handle varying noise levels effectively.

6.7 Summary and Key Results

6.7.1 ML Algorithms with Feature Extraction using LDA

The LDA ED-based FE application as a FE method, it attempts to reduce the dimensionality of the dataset while maintaining as much of the class discriminatory information as possible. The analysis of the performance of various machine learning algorithms (Decision Tree, SVM, k-NN, Random Forest, Neural Network) across different noise levels and sample sizes after applying LDA provides insights into how well each algorithm can generalize with reduced feature sets in noisy environments.

The Decision Tree model's accuracy showed a decreasing trend with lower sample sizes and higher noise levels, ranging from 72.90% to 4.19%. However, at lower noise levels, the model struggles significantly, indicating that the Decision Tree is highly sensitive to noise after FE reduction using LDA. The sensitivity metric follows a similar trend, showing the model's struggle to correctly identify positive instances under high noise and low sample conditions. The FDR increases with higher noise levels, suggesting that the model makes more false positive predictions as the quality of the data degrades. The AUC Mean indicates a significant drop at higher noise levels, reflecting a reduced ability of the Decision Tree to distinguish between classes. The standard deviation also increases, indicating inconsistency in model performance under different conditions.

The SVM performs relatively well with clean data (NoiseLevel_60_Sample_1_00) but struggles significantly as noise increases, with accuracy dropping from 71.22% to as low as 6.29%. Like accuracy, sensitivity declines with higher noise levels, indicating that SVM has difficulty correctly classifying instances when the data is noisy. The FDR increases substantially with noise, showing that SVM is prone to false positives in noisy environments. The SVM has a high AUC Mean in low-noise scenarios, but this performance deteriorates rapidly as noise increases. The high AUC Std at certain noise levels indicates that SVM's performance is inconsistent when dealing with reduced and noisy features.

The K-NN shows reasonable accuracy at higher noise levels compared to other models but still suffers a significant drop as noise increases, with accuracy ranging from 75.11% to 6.92%. The model's sensitivity also decreases with noise, highlighting that k-NN is less effective in identifying true positives as the data quality diminishes. The FDR indicates that k-NN makes more false positives under noisy conditions, but it handles moderate noise better than some

other models. The K-NN maintains a relatively higher AUC Mean compared to SVM and Decision Trees under similar noise conditions. However, the model's performance becomes less consistent with a higher AUC Std as noise increases.

The LDA-Random Forest generally performs better than other models in noisy environments, with the highest accuracy recorded at 79.06% under the cleanest conditions. However, its accuracy drops significantly in low sample, high-noise conditions. The sensitivity remains relatively higher for Random Forest compared to other models, showing its robustness in identifying true positives even with noisy data. The FDR is lower in LDA-Random Forest compared to other models, indicating fewer false positives, especially in less noisy scenarios. The LDA-RF consistently shows high AUC Mean values across various noise levels, reflecting its strong classification performance. The AUC Std is also relatively low, indicating consistent performance.

The ANN show competitive accuracy in clean environments but suffer a steep decline in accuracy under noisy conditions, with the lowest accuracy recorded at 3.42%. Likewise, accuracy, sensitivity drops as noise increases, highlighting that Neural Networks struggle to correctly classify instances when the data is noisy. The FDR increases with noise, indicating that Neural Networks become more prone to making false positive predictions as data quality diminishes. The ANN exhibit high AUC Mean values in less noisy environments, indicating good discrimination capability. However, the AUC Std increases with noise, reflecting inconsistency in model performance under varying noise levels.

The key takeaways on the impact of LDA, it effectively reduces dimensionality and preserves class discrimination in the data, but its effectiveness varies across algorithms. Random Forest and k-NN show more robustness to noise with LDA, while SVM and ANN are more affected by the noise after LDA. Whereas Random Forest superiority among the models tested, Random Forest consistently performs better across most metrics, showing the highest accuracy, sensitivity, and AUC Mean while maintaining lower FDR and AUC Std, indicating stable and reliable performance even in noisy conditions. Meanwhile all models exhibit a decline in performance as noise levels increase, but the degree of sensitivity varies. Decision Trees and SVMs are particularly sensitive, while k-NN and Random Forest show more resilience. The increase in AUC Std for most models at higher noise levels suggests that performance becomes more variable and less predictable, making them less reliable in noisy environments.

In conclusion, while LDA helps in dimensionality reduction, the choice of the machine learning model significantly impacts the overall performance, especially in noisy environments. Random Forest and k-NN emerge as more robust choices when applying LDA for feature extraction in scenarios with varying noise levels.

6.7.2 Key Observations from the comparison of LDA-ML and PCA-ML models

6.7.2.1 Decision Tree:

Accuracy performs well across the board for LDA, peaking at ~80.98% for specific conditions. **AUC Mean** is slightly lower than with PCA but still robust.

6.7.2.2 SVM:

Similar behavior in both PCA and LDA, but with LDA having slightly better performance at lower noise levels. LDA offers higher AUC Mean and more stability across different noise conditions.

6.7.2.3 k-NN:

For LDA, the accuracy and sensitivity increase over PCA. **FDR** is better managed compared to PCA, especially at higher noise levels, which indicates fewer false positives. Random Forest: **LDA** significantly boosts performance, with accuracy peaking at 82.21%. **AUC Mean** is slightly better, with LDA offering more stable performance.

6.7.2.4 Artificial Neural Networks:

LDA provides much better accuracy and sensitivity. **FDR** is better controlled under LDA, while PCA struggles at higher noise levels.

Table 6.16 Comparative Summary

Metric	PCA Performance	LDA Performance	Comparison
Accuracy	Ranges between 4.19% to 79.06%	Range between 24% to 82.21%	LDA shows better accuracy for Random Forest and k-NN.
Sensitivity	High variable, especially at higher noise levels	More stable across noise conditions	LDA performs better in handling sensitivity.
FDR	Increase significantly with higher noise	Better managed under LDA	LDA superior in reducing false positives.
AUC Mean	Range between 39% to 96%	Slightly better, ranging up to ~97%	LDA offers slight better AUC Mean values.
AUC Std	Moderate to high at lower noise level (PCA struggles at high noise levels)	LDA maintains lower AUC Std across noise levels	LDA shows less variability in AUC Std scores.

6.7.3 Summary of ED-FE (PCA and LDA)

LDA generally performs better than PCA across the models in terms of stability and accuracy, particularly under conditions of higher noise. LDA's ability to manage FDR and provide more consistent accuracy and sensitivity makes it a superior feature extraction technique for the fog computing application dataset in this context.

LDA's strength lies in reducing false positives (better FDR) and maintaining higher accuracy and sensitivity, especially for models like Random Forest, k-NN, and Neural Networks. On the other hand, PCA tends to be more sensitive to noise and shows higher variability in its AUC performance.

Thus, for FCAs, LDA is the better choice for ED-based FE compared to PCA.

6.8 Introduction

In this section, we compare the performance of the suggested eigen decomposition machine learning (ED-FE ML) models of specifically PCA and LDA. The comparison is based on various performance metrics such as accuracy, sensitivity, FDR, AUC mean, and AUC standard deviation across different noise levels and sample conditions.

6.8.1 ED-FE PCA and LDA

The comparison between PCA and LDA for feature extraction in machine learning models, we can assess the performance metrics such as accuracy, sensitivity, FDR (False Discovery Rate), AUC mean, and AUC standard deviation across various noise levels and sample conditions.

1. PCA vs LDA Analysis ML algorithms

6.8.1.1 Decision Tree

Accuracy ranges from ~4.19% to 72.90%. Sensitivity follows a similar trend as accuracy, with significant drops at lower noise levels.

FDR increases significantly as noise increases, indicating higher false positives at higher noise levels. AUC Mean ranges from ~39% to ~90%.

Standard deviation of AUC is relatively low at lower noise levels (~1.91% to ~6.76%).

6.8.1.2 SVM

Accuracy ranges from 6.29% to 71.23%. AUC Mean ranges from 39.52% to ~92.52%. At higher noise levels, the model struggles with performance (high FDR and lower accuracy).

6.8.1.3 k-NN

Accuracy drops significantly with noise, ranging from 7.19% to 75.11%. AUC Mean at higher noise levels (60% noise) is significantly lower (~47% to 94%).

6.8.1.4 Random Forest:

Among the highest performers for PCA-based feature extraction, with accuracy reaching up to 79.06%. AUC Mean is also strong, peaking at 96.98%.

Noise levels, especially at 0%, cause a significant drop in performance.

6.8.1.5 Neural Networks:

Accuracy ranges between ~3.42% and ~73.7%. AUC Mean peaks at ~94.52%. There is a significant drop in performance as noise levels increase.

6.9 Conclusion

The performance comparison reveals that different machine learning models exhibit varying levels of effectiveness when combined with PCA and LDA for feature extraction. Random Forest and Neural Networks generally perform better with PCA, showing higher accuracy and AUC mean values. However, noise levels significantly impact the performance of all models, with higher noise leading to increased FDR and decreased accuracy. These findings highlight the importance of selecting appropriate feature extraction techniques and considering noise levels in the development of robust machine learning models.

CHAPTER:7 : CONCLUSION AND FUTURE WORK

The distributed or multi-edge RAN in the form of 5G-FRAN utilise centralization and virtualization of CRAN, HCRAN, which brought various advantages such as reduced CAPEX/OPEX, enhanced mobility management, resource sharing, enhanced performance, system level efficiency, enhanced energy, and spectral efficiency. Throughout this thesis, we have analysed diverse methods that can benefit from distributed and joint decision-making in the novel RAN architectures, more precisely, in FRAN, and AI/ML incorporated architectures.

7.1 Research Summary

The contribution of this thesis detailed in the description of the study's aim and achievement. Subsequently, the results of each chapter are summarized to give an overview of the precise contribution of each chapter. Also, this chapter provide recommendations and ideas for future research possibilities. The chapter provides a summary of the findings and limitations of the thesis. In addition, recommendations for future research are presented. This section elaborates on the general contributions of this thesis. It begins with a summary of the study's aim and achievements. Subsequently, the key findings and contributions from different sections are presented to offer a comprehensive overview. The section also provides recommendations and ideas for future research, along with a summary of the thesis's findings and limitations.

5G necessitates a complete rethinking of mobile network design, particularly the RAN architecture, because to the rapidly growing subscriber population, huge connectivity, massive data, and new expectations for extremely low latency and higher data rates. An evaluation of the current RAN designs for 5G mobile networks, especially CRAN, HCRAN, and FRAN, was undertaken as a response. The goal was to explore the benefits of CRAN and HCRAN while not disregarding the challenges that each network design encounters. Consequently, current CRAN and HCRAN solutions were investigated. For energy efficiency, fronthaul capacity, latency, and other factors, detailed summaries were supplied in tabular style. In the case of FRAN, a different method was followed, with the focus being on resource categories within the design layers and then an analysis of resource management system.

The present research aims to clarify the overarching aim of this study, as delineated by the thesis named "5G New Radio and Fog Computing Scalability and QoS." The primary objective of this work is to devise an improved approach for resource management in the context of 5G-FRAN, using the capabilities of a machine learning algorithm. Provides a comprehensive overview of the study and examines the challenges encountered by the existing RAN architecture in managing resources for 5G. Thus, the thesis explains the primary hypotheses

employed for the purpose of investigating and addressing the research topic. Furthermore, highlights the significance of this study and underscores its practical implications.

A detailed background of fundamental principles and related literature are presented in thesis. The major contribution of opening section was to undertake a comprehensive evaluation of resource management for 5G-FRAN. Thus, various resources within the four layers the FRAN system were categorised. The main objective was to identify the novelty presented by the 5G Architecture that in corporate aspects of Cloud RAN, Heterogenous Cloud RAN edge computing of Fog Computing. The unique architecture of 5G-FRAN has the potential to optimise resource management using machine learning techniques for the 5G-FRAN system.

A full assessment of the 5G communication network, covering service classifications, preferred network topology, benefits, applications, and implementation concerns. In addition, the performance requirements, worldwide deployment of 5G, and enablement of numerous vertical sectors using network slicing and network virtualization were also considered.

The development of RAN architectures in the past and present is examined. Consequently, several RAN implementations which have been published in the literature are fully discussed. We also go over the present state of the art, ongoing standardisation initiatives, and why existing RAN systems need be rebuilt and remodelled to suit the demands of next-generation mobile networks.

A detailed comparative analysis of CRAN, HCRAN, and FRAN designs in this work is provided, with the goal of evaluating these RAN architectures from several perspective such as enhanced energy efficiency, enhanced energy efficiency, reduced CAPEX/OPEX, resource allocation and resource sharing, and so on.

In addition, we provide an in-depth analysis and overview of the essential enabling technologies for 5G RAN. We also look at 5G RAN improvements, which are projected to improve spectrum efficiency, lower CAPEX/OPEX, and meet end-user expectations as well as vertical industrial requirements Besides, extensive related work on the 5G-FRAN resource management is discussed in Chapter 2 to support the aim and achievement of this thesis.

7.2 Research Limitations

The availability of real-world data for this research posed a significant challenge. Consequently, certain aspects of the study relied on synthetic data. However, this does not compromise the credibility of the research. Careful attention was devoted to the creation of the synthetic dataset to ensure its relevance and accuracy. Various statistical measures and validation techniques were employed to meticulously design and vet the synthetic data,

ensuring it closely mirrors the characteristics and patterns found in real data. This rigorous approach maintains the integrity of the analysis and the conclusions drawn from the study, and ensures that the findings remain robust and applicable. This strategic methodology not only reinforces the research's reliability but also highlights its adaptability in the face of data constraints.

7.3 Future Research Directions

5G communications network resources present endless possibilities and various challenges. This section explores some issues that have not been addressed in detail in this work.

However, for future work like other 5G RAN systems, the (FRAN) still has several major research problems to overcome. By lessening traffic congestion at the cloud server and facilitating faster content access for (F-UEs), edge caching enhances FRAN performance. However, unlike centralised caching solutions, FRAN restricts the caching capability of individual (F-APs) and F-UEs. Enhancing overall caching performance requires optimising edge caching tactics, such as selecting which material to store and when to distribute caches across various edge devices.

REFERENCES

[1] GSMA, 2018. 2018 Mobile Industry Impact Report: Sustainable Development Goals. pp. 1-136, [Online] Available at: https://www.gsma.com/solutions-and-impact/connectivity-for-good/external-affairs/wp-content/upload/2020/05/2018-SDG-Impact-Report.pdf [Accessed 20 January 2025].

[2] United Nations Conference on Trade and Development (UNCTAD), 2019. Digital Economy Report 2019: Value creation and capture - implications for developing countries. Geneva: United Nations. [Online] Available at: https://unctad.org/webflyer/digital-economy-report-2019 [Accessed 21 January 2025].

[3] International Telecommunication Union, “Measuring digital development Facts and figures 2019,” *ITUPublications*, pp. 1–15, 2019, [Online]. Available: https://www.itu.int/en/mediacentre/Documents/MediaRelations/ITU Facts and Figures 2019 - Embargoed 5 November 1200 CET.pdf

[4] W., Coetzee, L., Wydeman, F., Mekuria, and Z., du Toit, 2018. Making 5G a Reality for Africa. Council for Scientific and Industrial Research (CSIR) and Ericsson.

[5] GSM Intelligence, 2018. The Mobile Economy. No. 35, p. 53 [Online] Available at: https://www.gsma.com/mobeeconomy/ [Accessed 21 January 2025]. doi: 10.5121/ijcsit.2015.7409.

[6] International Telecommunication Union (ITU), 2015. IMT, 2015. IMT Vision - Framework and overal objectives of the future development of IMT for 2020 and beyond. Recommendation ITU-R M.2083-0. [Online] Available at: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf [Accessed 20 January 2025].

[7] GSMA, 2020. The Mobile Economy Sub-Saharan Africa 2020. [Online] Available at: https://www.gsma.com/mobileeconomy/sub-saharan-africa/ [Accessed 21 January 2025].

[8] L., Vailshery, 2021. Internet of Things (IoT) connected devices worldwide in 2019 and 2030, by technology. [Online] Available at: https://www.idc.com/getdoc.jsp?containerId=IDC_P24793 [Accessed 21 Jauary 2025].

[9] M. Roggero, (2020) 'Technical Principles and Simulations of the 5G NR Physical Layer Standard'. Available at: https://www.mathworks.com/content/dam/mathworks/conference-or-academic-paper/technical-principles-simulations-5g-nr-physical-layer-standard.pdf

[10] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, “Cloud radio access network (C-RAN): A primer,” *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, 2015, doi: 10.1109/MNET.2015.7018201.

[11] J. L. C. MacGillivray, S. Crooka, M. Torchia, K. Chinta, S. Kotagi and and N. W. G. Roberti, H. Roman, A. Rotaru, Y. Torisu, *"Worldwide internet of things forecast update 2020–2024,” 2021.* 2021.

[12] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,” *IEEE Wirel. Commun.*, vol. 23, no. 5, pp. 60–67, 2016, doi: 10.1109/MWC.2016.7721743.

[13] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015, doi: 10.1016/j.bushor.2015.03.008.

[14] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, “Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies,” *IEEE Wirel. Commun.*, vol. 21, no. 6, pp. 126–135, 2014, doi: 10.1109/MWC.2014.7000980.

[15] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. C. M. Leung, “Fog radio access networks: Mobility management, interference mitigation, and resource optimization,” *IEEE Wirel. Commun.*, vol. 24, no. 6, pp. 120–127, 2017, doi: 10.1109/MWC.2017.1700007.

[16] CAICT and Huawei, “Virtual Reality / Augmented Reality White Paper,” 2017. [Online]. Available: https://www.huawei.com/au/press-events/events/ubbf2017/ar-vr-white-paper

[17] J. Jang, Y. Ko, and W. O. N. S. U. G. Shin, “Augmented Reality and Virtual Reality for Learning : An Examination Using an Extended Technology Acceptance Model,” vol. 9, 2021, doi: 10.1109/ACCESS.2020.3048708.

[18] P. D. A Doumanoglou, D Griffin, J Serrano, N Zioulis, T Khoa Phan, D Jiménez, D Zarpalas, F Alvarez, M Rio, “Quality of experience for 3-d immersive media streaming,” *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 379–391, 2018, doi: 10.1109/TBC.2018.2823909.

[19] ICT-317669-METIS, “Updated scenarios, requirements and KPIs for 5G mobile and wireless system with recommendations for future investigations (DeliverableD1.5),” *Mob. Wirel. Commun. Enablers Twenty-twenty Inf. Soc. V1.0*, pp. 1–57, 2015.

[20] ITU-R, “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,” *Itu-R M.2083-0*, vol. 0, p. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M, 2015.

[21] L. Alessio, P. Apostoli, I. Cortesi, and L. Lucchini, “Introduzione: Scopi e finalità del progetto multicentrico valutazione degli effetti conseguenti a basse dosi di mercurio inorganico da esposizioni ambientali ed occupazionali: Studio dei meccanismi di tossicità specifica in vitro e nell’uomo,” 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12197265

[22] Broadband Forum, “Cloud Central Office Reference Architectural TR-384 Framework,” 2018.

[23] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, “Toward an Efficient C-RAN Optical Fronthaul for the Future Networks: A Tutorial on Technologies, Requirements, Challenges, and Solutions,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp.

708–769, 2018, doi: 10.1109/COMST.2017.2773462.

[24] GSMA, “5G Implementation Guidelines : NSA Option 3,” *GSMA Futur. Networks*, no. February, pp. 1–42, 2020, [Online]. Available: https://www.gsma.com/futurenetworks/wiki/5g-implementation-guidelines/

[25] 5GMF, “5GMF White Paper, 5G Mobile Communications Systems for 2020 and beyond,” *Fifth Gener. Mob. Commun. Promot. Forum. Ver. 1.1*, 2017.

[26] S. Khatibi, L. Caeiro, L. S. Ferreira, L. M. Correia, and N. Nikaein, “Modelling and implementation of virtual radio resources management for 5G Cloud RAN,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, pp. 1–16, 2017, doi: 10.1186/s13638-017-0908-1.

[27] D. Pliatsios, P. Sarigiannidis, S. Goudos, and G. K. Karagiannidis, “Realizing 5G vision through Cloud RAN: technologies, challenges, and trends,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2018, no. 1, pp. 1–15, 2018, doi: 10.1186/s13638-018-1142-1.

[28] T. Taleb, A. Ksentini, and B. Sericola, “On Service Resilience in Cloud-Native 5G Mobile Systems,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 483–496, 2016, doi: 10.1109/JSAC.2016.2525342.

[29] M. Awais, A. Ahmed, S. A. Ali, M. Naeem, W. Ejaz, and A. Anpalagan, “Resource management in multicloud IoT radio access network,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3014–3023, 2019, doi: 10.1109/JIOT.2018.2878511.

[30] Y. Zhang and M. Chen, *Cloud Based 5G Wireless Networks*. 2016. doi: 10.1007/978-3-319-47343-7.

[31] R. S. Alhumaima, M. Khan, and H. S. Al-Raweshidy, “Component and parameterised power model for cloud radio access network,” *IET Commun.*, vol. 10, no. 7, pp. 745–752, 2016, doi: 10.1049/iet-com.2015.0752.

[32] R. Bassoli, M. Di Renzo, and F. Granelli, “Analytical energy-efficient planning of 5G cloud radio access network,” *IEEE Int. Conf. Commun.*, pp. 2–5, 2017, doi: 10.1109/ICC.2017.7996871.

[33] H. Zhang, Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, “Fronthauling for 5G LTE-U ultra dense cloud small cell networks,” *IEEE Wirel. Commun.*, vol. 23, no. 6, pp. 48–53, 2016, doi: 10.1109/MWC.2016.1600066WC.

[34] Y. Zhang and M. Chen, *Cloud Based 5G Wireless Networks*. 2016. doi: 10.1007/978-3-319-47343-7.

[35] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, “Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems,” *J. Commun. Networks*, vol. 18, no. 2, pp. 135–149, 2016, doi: 10.1109/JCN.2016.000023.

[36] V. Suryaprakash, P. Rost, and G. Fettweis, “Are Heterogeneous Cloud-Based Radio Access Networks Cost Effective?,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2239–2251, 2015, doi: 10.1109/JSAC.2015.2435275.

[37] T.-M. T. . F. L Chen, L Liu, X Fan, J Li, C Wang, G Pan, J Jakubowicz, “Complementary base station clustering for cost-effective and energy-efficient cloud-RAN,” *2017 IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017* -, pp. 1–7, 2018, doi: 10.1109/UIC-ATC.2017.8397526.

[38] O. Arouk, T. Turletti, N. Nikaein, and K. Obraczka, “Cost Optimization of Cloud-RAN Planning and Provisioning for 5G Networks,” *IEEE Int. Conf. Commun.*, vol. 2018-May, pp. 0–5, 2018, doi: 10.1109/ICC.2018.8422744.

[39] M. De Andrade, M. Tornatore, A. Pattavina, A. Hamidian, and K. Grobe, “Cost models for BaseBand Unit (BBU) hotelling: From local to cloud,” *2015 IEEE 4th Int. Conf. Cloud Networking, CloudNet 2015*, pp. 201–204, 2015, doi: 10.1109/CloudNet.2015.7335306.

[40] C. Mobile, “C-RAN: the road towards green RAN,” *White Pap. ver 3.0*, vol. 0, pp. 15–16, 2015, [Online]. Available: http://labs.chinamobile.com/cran/wp-content/uploads/2015/09/NGFI-Whitepaper_EN_v1.0_201509291.pdf

[41] U. Karneyenka, K. Mohta, and M. Moh, “Location and mobility aware resource management for 5G cloud radio access networks,” *Proc. - 2017 Int. Conf. High Perform. Comput. Simulation, HPCS 2017*, pp. 168–175, 2017, doi: 10.1109/HPCS.2017.35.

[42] G. Kardaras and C. Lanzani, “Advanced multimode radio for wireless & mobile broadband communication,” *Eur. Microw. Week 2009 Sci. Prog. Qual. Radiofreq. Conf. Proc. - 2nd Eur. Wirel. Technol. Conf. EuWIT 2009*, no. September, pp. 132–135, 2009.

[43] F. Tian, P. Zhang, and Z. Yan, “A Survey on C-RAN Security,” *IEEE Access*, vol. 5, pp. 13372–13386, 2017, doi: 10.1109/ACCESS.2017.2717852.

[44] M. Tahir, M. H. Habaebi, M. Dabbagh, A. Mughees, A. Ahad, and K. I. Ahmed, “A Review on Application of Blockchain in 5G and beyond Networks: Taxonomy, Field-Trials, Challenges and Opportunities,” *IEEE Access*, vol. 8, pp. 115876–115904, 2020, doi: 10.1109/ACCESS.2020.3003020.

[45] J. You, Z. Zhong, G. Wang, and B. Ai, “Security and Reliability Performance Analysis for Cloud Radio Access Networks with Channel Estimation Errors,” *IEEE Access*, vol. 2, pp. 1348–1358, 2014, doi: 10.1109/ACCESS.2014.2370391.

[46] B. Niu, Y. Zhou, H. Shah-Mansouri, and V. W. S. Wong, “A Dynamic Resource Sharing Mechanism for Cloud Radio Access Networks,” *IEEE Trans. Wirel. Commun.*, vol. 15, no. 12, pp. 8325–8338, 2016, doi: 10.1109/TWC.2016.2613896.

[47] P. Patil and W. Yu, “Generalized Compression Strategy for the Downlink Cloud Radio Access Network,” *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6766–6780, 2019, doi: 10.1109/TIT.2019.2930568.

[48] P. Patil, B. Dai, and W. Yu, “Hybrid data-sharing and compression strategy for

downlink cloud radio access network," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5370–5384, 2018, doi: 10.1109/TCOMM.2018.2842758.

[49] H. Yu and J. Joung, "Optimization of Frame Structure and Fronthaul Compression for Uplink C-RAN under Time-Varying Channels," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 2, pp. 1278–1292, 2021, doi: 10.1109/TWC.2020.3032432.

[50] O. Simeone, O. Somekh, H. V. Poor, and S. Shamai, "Downlink multicell processing with limited-backhaul capacity," *EURASIP J. Adv. Signal Process.*, vol. 2009, 2009, doi: 10.1155/2009/840814.

[51] S. H. Park, O. Simeone, O. Sahin, and S. Shamai, "Joint precoding and multivariate backhaul compression for the downlink of cloud radio access networks," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5646–5658, 2013, doi: 10.1109/TSP.2013.2280111.

[52] S. H. Park, O. Simeone, O. Sahin, and S. Shamai Shitz, "Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 69–79, 2014, doi: 10.1109/MSP.2014.2330031.

[53] R. G. Stephen and R. Zhang, "Joint millimeter-wave fronthaul and OFDMA resource allocation in ultra-dense CRAN," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1411–1423, 2017, doi: 10.1109/TCOMM.2017.2649519.

[54] S. Ahn, S. I. Park, J. Y. Lee, N. Hur, and J. Kang, "Fronthaul Compression and Precoding Optimization for NOMA-Based Joint Transmission of Broadcast and Unicast Services in C-RAN," *IEEE Trans. Broadcast.*, vol. 66, no. 4, pp. 786–799, 2020, doi: 10.1109/TBC.2019.2960929.

[55] J. Kim, S. H. Park, O. Simeone, I. Lee, and S. S. Shitz, "Joint Design of Fronthauling and Hybrid Beamforming for Downlink C-RAN Systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4423–4434, 2019, doi: 10.1109/TCOMM.2019.2903142.

[56] Y. Zhou and W. Yu, "Fronthaul Compression and Transmit Beamforming," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4138–4151, 2016.

[57] Y. Zhang, X. He, C. Zhong, L. Meng, and Z. Zhang, "Fronthaul Compression and Beamforming Optimization for Uplink C-RAN with Intelligent Reflecting Surface-Enhanced Wireless Fronthauling," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1979–1983, 2021, doi: 10.1109/LCOMM.2021.3062861.

[58] T. X. Vu, H. D. Nguyen, T. Q. S. Quek, and S. Sun, "Fronthaul compression and optimization for cloud radio access networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6. doi: 10.1109/ICC.2016.7511263.

[59] E. Heo, O. Simeone, and H. Park, "Optimal fronthaul compression for synchronization in the uplink of cloud radio access networks," *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, 2017, doi: 10.1186/s13638-017-0804-8.

[60] O. Taghizadeh, T. Yang, and R. Mathar, “Private Uplink Communication in C-RAN with Untrusted Radios,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 8034–8039, 2020, doi: 10.1109/TVT.2020.2995603.

[61] L. Liu, S. Bi, and R. Zhang, “Joint Power Control and Fronthaul Rate Allocation for Throughput Maximization in OFDMA-Based Cloud Radio Access Network,” *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4097–4110, 2015, doi: 10.1109/TCOMM.2015.2469687.

[62] L. Liu and W. Yu, “Cross-Layer Design for Downlink Multihop Cloud Radio Access Networks with Network Coding,” *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1728–1740, 2017, doi: 10.1109/TSP.2017.2649486.

[63] Y. Li, T. Jiang, K. Luo, and S. Mao, “Green Heterogeneous Cloud Radio Access Networks: Potential Techniques, Performance Trade-offs, and Challenges,” *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 33–39, 2017, doi: 10.1109/MCOM.2017.1600807.

[64] H. Dahrouj, A. Douik, O. Dhifallah, T. Y. Al-Naffouri, and M.-S. S. Alouini, “Resource allocation in heterogeneous cloud radio access networks: Advances and challenges,” *IEEE Wirel. Commun.*, vol. 22, no. 3, pp. 66–73, Jun. 2015, doi: 10.1109/MWC.2015.7143328.

[65] M. Elhattab, M. A. Arfaoui, and C. Assi, “CoMP Transmission in Downlink NOMA-Based Heterogeneous Cloud Radio Access Networks,” *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7779–7794, 2020, doi: 10.1109/TCOMM.2020.3021145.

[66] M. Matinmikko, H. Okkonen, M. Palola, S. Yrjola, P. Ahokangas, and M. Mustonen, “Spectrum sharing using licensed shared access: the concept and its workflow for LTE-advanced networks,” *IEEE Wirel. Commun.*, vol. 21, no. 2, pp. 72–79, Apr. 2014, doi: 10.1109/MWC.2014.6812294.

[67] R. E. Learned, S. E. Johnston, and N. J. Kaminski, “Cognitive coexistence: A throughput study of MUD-enhanced opportunistic spectrum access,” *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, pp. 1455–1462, 2013, doi: 10.1109/ACSSC.2013.6810537.

[68] C. Liang and F. R. Yu, “Wireless Network Virtualization: A Survey, Some Research Issues and Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 358–380, 2015, doi: 10.1109/COMST.2014.2352118.

[69] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, “Radio access network virtualization for future mobile carrier networks,” *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 27–35, 2013, doi: 10.1109/MCOM.2013.6553675.

[70] J. Demestichas, P Georgakopoulos, A Karvounas, D Tsagkaris, K Stavroulaki, V Lu, J Xiong, C Yao, “5G on the Horizon: Key challenges for the radio-access network,” *IEEE Veh. Technol. Mag.*, vol. 8, no. 3, pp. 47–53, 2013, doi: 10.1109/MVT.2013.2269187.

[71] J. C. Bernardos, Carlos J.; De La Oliva, A.; Serrano, P.; Banchs, A.; Contreras, L. M.; Jin, Hao.; Zúniga, J. C.; Zúñiga, “An architecture for software defined wireless networking,” *IEEE Wirel. Commun.*, vol. 21, no. 3, pp. 52–61, Jun. 2014,

doi: 10.1109/MWC.2014.6845049.

[72] C. B. Marotta, M. A; Kaminski, N, Gomez-Miguelez, I; Granville, L. Z; Rochol, J; Da Silva, L; Both, “Resource sharing in heterogeneous cloud radio access networks,” *IEEE Wirel. Commun.*, vol. 22, no. 3, pp. 74–82, 2015, doi: 10.1109/MWC.2015.7143329.

[73] A. Onireti, O; Zoha, A; Moysen, J; Imran, A; Giupponi, L; Ali; Imran, M; Abu-Dayya, “A cell outage management framework for dense heterogeneous networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2097–2113, 2016, doi: 10.1109/TVT.2015.2431371.

[74] W. Wang, J. Zhang, and Q. Zhang, “Cooperative cell outage detection in Self-Organizing femtocell networks,” *Proc. - IEEE INFOCOM*, pp. 782–790, 2013, doi: 10.1109/INFCOM.2013.6566865.

[75] W. Xue, H. Zhang, Y. Li, D. Liang, and M. Peng, “Cell outage detection and compensation in two-tier heterogeneous networks,” *Int. J. Antennas Propag.*, vol. 2014, 2014, doi: 10.1155/2014/624858.

[76] L. Wang, W. Huang, Y. Fan, and X. Wang, “Priority-based cell selection for mobile equipments in heterogeneous cloud radio access networks,” *2015 Int. Conf. Connect. Veh. Expo, ICCVE 2015 - Proc.*, pp. 62–67, 2016, doi: 10.1109/ICCVE.2015.53.

[77] N. Chen, B. Rong, X. Zhang, and M. Kadoch, “Scalable and flexible massive MIMO precoding for 5G H-CRAN,” *IEEE Wirel. Commun.*, vol. 24, no. 1, pp. 46–52, 2017, doi: 10.1109/MWC.2017.1600139WC.

[78] Q. Liu, T. Han, N. Ansari, and G. Wu, “On Designing Energy-Efficient Heterogeneous Cloud Radio Access Networks,” *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 3, pp. 721–734, 2018, doi: 10.1109/TGCN.2018.2835451.

[79] L. Chen, H. Jin, H. Li, J. B. Seo, Q. Guo, and V. Leung, “An Energy efficient implementation of c-ran in HetNet,” *IEEE Veh. Technol. Conf.*, 2014, doi: 10.1109/VTCFall.2014.6965870.

[80] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. S. Dhillon, “An overview of load balancing in hetnets: old myths and open problems,” *IEEE Wirel. Commun.*, vol. 21, no. 2, pp. 18–25, Apr. 2014, doi: 10.1109/MWC.2014.6812287.

[81] C. Ran, S. Wang, and C. Wang, “Balancing backhaul load in heterogeneous cloud radio access networks,” *IEEE Wirel. Commun.*, vol. 22, no. 3, pp. 42–48, 2015, doi: 10.1109/MWC.2015.7143325.

[82] M. Gastpar, “The Wyner-Ziv problem with multiple sources,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2762–2768, 2004, doi: 10.1109/TIT.2004.836707.

[83] Y. Qi, M. Z. Shakir, M. A. Imran, A. Quddus, and R. P. Tafazolli, “How to Solve the Fronthaul Traffic Congestion Problem in H-CRAN?,” *2016 IEEE Int. Conf. Commun. Work. ICC 2016*, pp. 240–245, 2016, doi: 10.1109/ICCW.7501907.

[84] R. Balakrishnan and B. Canberk, “Traffic-aware QoS provisioning and admission

control in OFDMA hybrid small cells," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 802–810, 2014, doi: 10.1109/TVT.2013.2280124.

[85] D. C. Chen, T. Q. S. Quek, and M. Kountouris, "Backhauling in Heterogeneous Cellular Networks: Modeling and Tradeoffs," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 6, pp. 3194–3206, 2015, doi: 10.1109/TWC.2015.2403321.

[86] S. C. Hung, H. Hsu, S. M. Cheng, Q. Cui, and K. C. Chen, "Delay Guaranteed Network Association for Mobile Machines in Heterogeneous Cloud Radio Access Network," *IEEE Trans. Mob. Comput.*, vol. 17, no. 12, pp. 2744–2760, 2018, doi: 10.1109/TMC.2018.2815702.

[87] S. Lien, S. Hung, K. Chen, and Y. Liang, "Ultra-Low-Latency Ubiquitous Connections In Heterogeneous Cloud Radio Access Networks," no. June, pp. 22–31, 2015.

[88] H. Xiang, Y. Yu, Z. Zhao, Y. Li, and M. Peng, "Tradeoff between energy efficiency and queues delay in heterogeneous cloud radio access networks," *2015 IEEE Int. Conf. Commun. Work. ICCW 2015*, pp. 2727–2731, 2015, doi: 10.1109/ICCW.2015.7247591.

[89] Y. Qi and H. Wang, "Interference-aware User Association under Cell Sleeping for Heterogeneous Cloud Cellular Networks," *IEEE Wirel. Commun. Lett.*, vol. 6, no. 2, pp. 242–245, 2017, doi: 10.1109/LWC.2017.2665556.

[90] H. Zhang, C. Jiang, J. Cheng, and V. C. M. M. Leung, "Cooperative interference mitigation and handover management for heterogeneous cloud small cell networks," *IEEE Wirel. Commun.*, vol. 22, no. 3, pp. 92–99, Jun. 2015, doi: 10.1109/MWC.2015.7143331.

[91] I. Al-Samman, R. Almesaeed, A. Doufexi, M. Beach, and A. Nix, "User weighted probability algorithm for heterogeneous C-RAN interference mitigation," *IEEE Int. Conf. Commun.*, pp. 0–6, 2017, doi: 10.1109/ICC.2017.7997213.

[92] M. Peng, X. Xie, Q. Hu, J. Zhang, and H. V. Poor, "Contract-based interference coordination in heterogeneous cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1140–1153, 2015, doi: 10.1109/JSAC.2015.2416985.

[93] M. Peng, H. Xiang, Y. Cheng, S. Yan, and H. V. Poor, "Inter-tier interference suppression in heterogeneous cloud radio access networks," *IEEE Access*, vol. 3, no. 6, pp. 2441–2455, 2015, doi: 10.1109/ACCESS.2015.2497268.

[94] OpenFog Consortium Architecture Working Group and OpenfogConsortium, "OpenFog Reference Architecture for Fog Computing," *OpenFogConsortium*, no. February, pp. 1–162, 2017, doi: OPFRA001.020817.

[95] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.

[96] K. Liang, L. Zhao, X. Zhao, Y. Wang, and S. Ou, "Joint resource allocation and

coordinated computation offloading for fog radio access networks," *China Commun.*, vol. 13, no. 2, pp. 131–139, 2016, doi: 10.1109/CC.2016.7833467.

[97] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, 2016, doi: 10.1109/MNET.2016.7513863.

[98] Y. Y. Shih, W. H. Chung, A. C. Pang, T. C. Chiu, and H. Y. Wei, "Enabling Low-Latency Applications in Fog-Radio Access Networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, 2017, doi: 10.1109/MNET.2016.1500279NM.

[99] A. C. Ku, Y.J. Lin, D. Y. Lee, C.F. Hsieh, P.J. Wei, H.Y. Chou, C.T. Pang, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 46–52, 2017, doi: 10.1109/MCOM.2017.1600893.

[100] S. Jia, Y. Ai, Z. Zhao, M. Peng, and C. Hu, "Hierarchical content caching in fog radio access networks: Ergodic rate and transmit latency," *China Commun.*, vol. 13, no. 12, pp. 1–14, 2016, doi: 10.1109/CC.2016.7897534.

[101] S. Kim, "Fog radio access network system control scheme based on the embedded game model," *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, 2017, doi: 10.1186/s13638-017-0900-9.

[102] S. H. Park, O. Simeone, and S. Shamai, "Joint cloud and edge processing for latency minimization in Fog Radio Access Networks," *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, vol. 2016-Augus, pp. 1–5, 2016, doi: 10.1109/SPAWC.2016.7536737.

[103] S. H. Park, O. Simeone, and S. S. Shitz, "Joint Optimization of Cloud and Edge Processing for Fog Radio Access Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 11, pp. 7621–7632, 2016, doi: 10.1109/TWC.2016.2605104.

[104] X. Wang, S. Leng, and K. Yang, "Social-Aware Edge Caching in Fog Radio Access Networks," *IEEE Access*, vol. 5, pp. 8492–8501, 2017, doi: 10.1109/ACCESS.2017.2693440.

[105] G. M. Shafiqur Rahman, M. Peng, K. Zhang, and S. Chen, "Radio Resource Allocation for Achieving Ultra-Low Latency in Fog Radio Access Networks," *IEEE Access*, vol. 6, pp. 17442–17454, 2018, doi: 10.1109/ACCESS.2018.2805303.

[106] C. Dao, N. N., Lee, J., Vu, D. N., Paek, J., Kim, J., Cho, S., Keum, "Adaptive Resource Balancing for Serviceability Maximization in Fog Radio Access Networks," *IEEE Access*, vol. 5, pp. 14548–14559, 2017, doi: 10.1109/ACCESS.2017.2712138.

[107] M. Peng and K. Zhang, "Recent Advances in Fog Radio Access Networks: Performance Analysis and Radio Resource Allocation," *IEEE Access*, vol. 4, pp. 5003–5009, 2016, doi: 10.1109/ACCESS.2016.2603996.

[108] K. Liang, L. Zhao, X. Chu, and H. H. Chen, "An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing," *IEEE Netw.*, vol. 31, no. 1, pp. 80–87, 2017, doi:

10.1109/MNET.2017.1600027NM.

[109] D. Chen, H. Al-Shatri, T. Mahn, A. Klein, and V. Kuehn, “Energy Efficient Robust F-RAN Downlink Design for Hard and Soft Fronthauling,” *IEEE Veh. Technol. Conf.*, vol. 2018-June, pp. 1–5, 2018, doi: 10.1109/VTCSpring.2018.8417561.

[110] Y. Qiu, H. Zhang, K. Long, Y. Huang, X. Song, and V. C. M. Leung, “Energy-Efficient Power Allocation with Interference Mitigation in MmWave-Based Fog Radio Access Networks,” *IEEE Wirel. Commun.*, vol. 25, no. 4, pp. 25–31, 2018, doi: 10.1109/MWC.2018.1700409.

[111] D. Wang, Z. Liu, X. Wang, and Y. Lan, “Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks,” *IEEE Access*, vol. 7, pp. 43356–43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

[112] S. Yan, M. Peng, M. A. Abana, and W. Wang, “An evolutionary game for user access mode selection in fog radio access networks,” *arXiv*, vol. 5, 2017.

[113] H. Xiang, M. Peng, Y. Cheng, and H. H. Chen, “Joint mode selection and resource allocation for downlink fog radio access networks supported D2D,” *Proc. 11th EAI Int. Conf. Heterog. Netw. Qual. Reliab. Secur. Robustness, QSHINE 2015*, no. Qshine, pp. 177–182, 2015, doi: 10.4108/eai.19-8-2015.2260167.

[114] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez, “WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul,” *IEEE Trans. Mob. Comput.*, vol. 17, no. 10, pp. 2452–2466, 2018, doi: 10.1109/TMC.2018.2793859.

[115] H. Hu and R. Wang, “User-centric local mobile cloud-assisted D2D communications in heterogeneous cloud-RANs,” *IEEE Wirel. Commun.*, vol. 22, no. 3, pp. 59–65, 2015, doi: 10.1109/MWC.2015.7143327.

[116] 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, doi: 10.1109/TWC.2020.2965927.

[117] H. Xiang, W. Zhou, M. Daneshmand, and M. Peng, “Network Slicing in Fog Radio Access Networks: Issues and Challenges,” *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 110–116, 2017, doi: 10.1109/MCOM.2017.1700523.

[118] P. Rost *et al.*, “Network slicing to enable scalability and flexibility in 5G mobile networks,” *arXiv*, no. May, pp. 72–79, 2017.

[119] A. Zappone, M. Di Renzo, and M. Debbah, “Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?,” *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019, doi: 10.1109/TCOMM.2019.2924010.

[120] O. Simeone, “A Very Brief Introduction to Machine Learning with Applications to Communication Systems,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, 2018, doi: 10.1109/TCCN.2018.2881442.

[121] M. Xia, Y. Owada, M. Inoue, and H. Harai, “Optical and wireless hybrid access

networks: Design and optimization," *J. Opt. Commun. Netw.*, vol. 4, no. 10, pp. 749–759, 2012, doi: 10.1364/JOCN.4.000749.

[122] R. C. Qiu *et al.*, "Cognitive radio network for the smart grid: Experimental system architecture, control algorithms, security, and microgrid testbed," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 724–740, 2011, doi: 10.1109/TSG.2011.2160101.

[123] H. Nguyen, G. Zheng, R. Zheng, and Z. Han, "Binary inference for primary user separation in cognitive radio networks," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 4, pp. 1532–1542, 2013, doi: 10.1109/TWC.2013.022213.112260.

[124] B. K. Donohoo, C. Ohlsen, S. Pasricha, Y. Xiang, and C. Anderson, "Context-aware energy enhancements for smart mobile devices," *IEEE Trans. Mob. Comput.*, vol. 13, no. 8, pp. 1720–1732, 2014, doi: 10.1109/TMC.2013.94.

[125] V. Sen Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 505–512, 2012, doi: 10.1109/TPDS.2011.156.

[126] P. Gandotra and R. K. Jha, "Adaptive Resource Block Allocation for Green 5G Wireless Communication Networks," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, vol. 2018-Decem, pp. 1–6, 2018, doi: 10.1109/ANTS.2018.8710144.

[127] A. Abrol, R. K. Jha, S. Jain, and P. Kumar, "Joint power allocation and relay selection strategy for 5G network: a step towards green communication," *Telecommun. Syst.*, vol. 68, no. 2, pp. 201–215, 2018, doi: 10.1007/s11235-017-0385-1.

[128] I. A. M. Balapuwaduge and F. Y. Li, "Hidden Markov Model Based Machine Learning for mMTC Device Cell Association in 5G Networks," *IEEE Int. Conf. Commun.*, vol. 2019-May, pp. 1–6, 2019, doi: 10.1109/ICC.2019.8761913.

[129] G. M. D. Santana, R. S. Cristo, J. P. Diguet, C. Dezan, O. P. M. Diana, and B. R. L. J. C. Kalinka, "A Case Study of Primary User Arrival Prediction Using the Energy Detector and the Hidden Markov Model in Cognitive Radio Networks," *Proc. - IEEE Symp. Comput. Commun.*, vol. 2019-June, pp. 2019–2022, 2019, doi: 10.1109/ISCC47284.2019.8969632.

[130] S. DASH, S. MAHESHWARI, and S. MAHAPATRA, "Traffic Prediction in Future Mobile Networks using Hidden Markov Model," no. October, 2019.

[131] M. J. Piran, N. H. Tran, D. Y. Suh, J. Bin Song, C. S. Hong, and Z. Han, "QoE-Driven Channel Allocation and Handoff Management for Seamless Multimedia in Cognitive 5G Cellular Networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6569–6585, 2017, doi: 10.1109/TVT.2016.2629507.

[132] T. Report, "O-RAN Working Group 2 AI / ML workflow description and requirements," pp. 1–51, 2020.

[133] H. Lee, J. Cha, D. Kwon, M. Jeong, and I. Park, "Hosting AI/ML Workflows on O-RAN RIC Platform," *2020 IEEE Globecom Work. GC Wkshps 2020 - Proc.*, 2020, doi: 10.1109/GCWkshps50303.2020.9367572.

[134] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, “Applicability domains of machine learning in next generation radio access networks,” *Proc. - 6th Annu. Conf. Comput. Sci. Comput. Intell. CSCI 2019*, pp. 1066–1073, 2019, doi: 10.1109/CSCI49370.2019.00203.

[135] A. Aprem, C. R. Murthy, and N. B. Mehta, “Transmit power control policies for energy harvesting sensors with retransmissions,” *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 5, pp. 895–906, 2013, doi: 10.1109/JSTSP.2013.2258656.

[136] O. Onireti *et al.*, “A cell outage management framework for dense heterogeneous networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2097–2113, 2016, doi: 10.1109/TVT.2015.2431371.

[137] S. Maghsudi and S. Stanczak, “Channel Selection for Network-Assisted D2D Communication via No-Regret Bandit Learning with Calibrated Forecasting,” *IEEE Trans. Wirel. Commun.*, vol. 14, no. 3, pp. 1309–1322, 2015, doi: 10.1109/TWC.2014.2365803.

[138] A. T. Nassar and Y. Yilmaz, “Reinforcement-learning-based resource allocation in fog radio access networks for various IoT environments,” *arXiv Prepr. arXiv1806.04582*, 2018.

[139] J. Zhao, M. Kong, Q. Li, and X. Sun, “Contract-Based Computing Resource Management via Deep Reinforcement Learning in Vehicular Fog Computing,” *IEEE Access*, vol. 8, pp. 3319–3329, 2020, doi: 10.1109/ACCESS.2019.2963051.

[140] J. Moon, O. Simeone, S. H. Park, and I. Lee, “Online reinforcement learning of X-haul content delivery mode in fog radio access networks,” *arXiv*, vol. 26, no. 10, pp. 1451–1455, 2019.

[141] J. Yu, R. Wang, and J. Wu, “QoS-Driven Resource Optimization for Intelligent Fog Radio Access Network: A Dynamic Power Allocation Perspective,” pp. 1–12, 2021, [Online]. Available: http://arxiv.org/abs/2103.02134

[142] N. N. Khumalo, O. O. Oyerinde, and L. Mfupe, “Reinforcement learning-based resource management model for fog radio access network architectures in 5G,” *IEEE Access*, vol. 9, pp. 12706–12716, 2021, doi: 10.1109/ACCESS.2021.3051695.

[143] T. Wei, Y. Sun, Y. Zhang, Z. Wang, W. Wu, and J. Gao, “Energy Efficient User Access and Computation Offloading Strategy for Fog Radio Access Network with Uplink/Downlink Decoupling,” *2019 IEEE 5th Int. Conf. Comput. Commun. ICCC 2019*, pp. 894–900, 2019, doi: 10.1109/ICCC47050.2019.9064102.

[144] L. Mai, N. N. Dao, and M. Park, “Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing,” *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18092830.

[145] J. Yao and N. Ansari, “Task Allocation in Fog-Aided Mobile IoT by Lyapunov Online Reinforcement Learning,” *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 556–565, 2020, doi: 10.1109/TGCN.2019.2956626.

[146] Y. Zhou, M. Peng, S. Yan, and Y. Sun, “Deep Reinforcement Learning Based Coded Caching Scheme in Fog Radio Access Networks,” *2018 IEEE/CIC Int. Conf. Commun. China, ICCC Work. 2018*, pp. 309–313, 2019, doi: 10.1109/ICCChinaW.2018.8674478.

[147] M. Seid, “Machine Learning Based Unmanned Aerial Vehicle enabled Fog-Radio Access Network and Edge Computing 《 ZTE Communications 》 网络首发论文,” vol. 17, no. March, pp. 33–45, 2020, doi: 10.12142/ZTECOM.201904006.

[148] Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, and ..., “Cloud-based multi-agent cooperation for IoT devices using workflow-nets,” *J. Grid …*, 2019, doi: 10.1007/s10723-019-09485-z.

[149] Z. Ren, T. Lu, X. Wang, W. Guo, G. Liu, and ..., “Resource scheduling for delay-sensitive application in three-layer fog-to-cloud architecture,” *Peer-to-Peer Netw. …*, 2020, doi: 10.1007/s12083-020-00900-x.

[150] G. C. Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*. books.google.com, 2011. [Online]. Available: https://books.google.com/books?hl=en%5C&lr=%5C&id=h6q-e4Q_rzgC%5C&oi=fnd%5C&pg=PR3%5C&dq=hard+%22real+time%22+computing+systems+predictable+scheduling+algorithms+and+applications%5C&ots=jAB3jPNLx8%5C&sig=CAjlZrhEQgQuuMHOkUTlFzZHFxk

[151] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” *… Mob. cloud Comput.*, 2012, doi: 10.1145/2342509.2342513.

[152] A. Kertész, T. Pflanzner, and T. Gyimóthy, “A mobile IoT device simulator for IoT-Fog-Cloud systems,” *J. Grid Comput.*, 2019, doi: 10.1007/s10723-018-9468-9.

[153] W. Wang, C. Feng, B. Zhang, and H. Gao, “Environmental Monitoring Based on Fog Computing Paradigm and Internet of Things,” *IEEE Access*, vol. 7, pp. 127154–127165, 2019, doi: 10.1109/ACCESS.2019.2939017.

[154] Riya, N. Gupta, and S. K. Dhurandher, “Efficient caching method in fog computing for internet of everything,” *Peer-to-Peer Netw. …*, 2021, doi: 10.1007/s12083-020-00952-z.

[155] A. Čolaković and M. Hadžialić, “Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues,” *Comput. networks*, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618305243

[156] B. M. Nguyen, H. T. T. Binh, T. T. Anh, and D. B. Son, “Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment,” *Appl. Sci.*, 2019, [Online]. Available: https://www.mdpi.com/452378

[157] A. S. Abohamama, M. F. Alrahmawy, and M. A. Elsoud, “Improving the dependability of cloud environment for hosting real time applications,” *Ain Shams*

Eng. …, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447917301466

[158] X. Q. Pham, N. D. Man, N. D. T. Tri, and ..., "A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing," *… J. Distrib. …*, 2017, doi: 10.1177/1550147717742073.

[159] V. S. Pana, O. P. Babalola, and V. Balyan, "5G radio access networks: A survey," *Array*, vol. 14, p. 100170, 2022, doi: https://doi.org/10.1016/j.array.2022.100170.

[160] V. S. Pana, O. P. Babalola, and V. Balyan, "5G radio access networks : A survey," *Array*, vol. 14, no. April, p. 100170, 2022, doi: 10.1016/j.array.2022.100170.

[161] S. Li, M. A. Maddah-Ali, and ..., "Coding for distributed fog computing," *IEEE Commun.* …, 2017, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7901473/

[162] I. Petri, J. Diaz-Montes, O. Rana, Y. Rezgui, and ..., "Coordinating data analysis and management in multi-layered clouds," *Internet Things. IoT …*, 2016, doi: 10.1007/978-3-319-47063-4_37.

[163] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv Prepr. arXiv …*, 2015, [Online]. Available: https://arxiv.org/abs/1502.01815

[164] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018, doi: 10.1109/COMST.2018.2814571.

[165] L. Liu, D. Qi, N. Zhou, and Y. Wu, "A task scheduling algorithm based on classification mining in fog computing environment," *Wirel. Commun. Mob. …*, 2018, [Online]. Available: https://www.hindawi.com/journals/wcmc/2018/2102348/

[166] M. M. E. Mahmoud, J. Rodrigues, K. Saleem, and ..., "Towards energy-aware fog-enabled cloud of things for healthcare," *Comput. \&Electrical …*, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790618300399

[167] H. T. T. Binh, T. T. Anh, D. B. Son, P. A. Duc, and ..., "An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment," *Proc. 9th …*, 2018, doi: 10.1145/3287921.3287984.

[168] J. F. Kurose and K. W. Ross, "Computer networking: A top down approach sixth edition," *Citado na*, 2012.

[169] M. Alhamad, T. Dillon, and E. Chang, "Conceptual SLA framework for cloud computing," *4th IEEE Int. Conf. Digit. Ecosyst. Technol. - Conf. Proc. IEEE-DEST 2010, DEST 2010*, pp. 606–610, 2010, doi: 10.1109/DEST.2010.5610586.

[170] L. Wu, S. K. Garg, R. Buyya, C. Chen, and S. Versteeg, "Automated SLA

negotiation framework for cloud computing," *Proc. - 13th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2013*, pp. 235–244, 2013, doi: 10.1109/CCGrid.2013.64.

[171] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, "Quality of Experience (QoE)-aware placement of applications in Fog computing environments," *J. Parallel Distrib. Comput.*, vol. 132, pp. 190–203, 2019, doi: 10.1016/j.jpdc.2018.03.004.

[172] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," *Proc. - 2017 IEEE 1st Int. Conf. Fog Edge Comput. ICFEC 2017*, pp. 89–96, 2017, doi: 10.1109/ICFEC.2017.12.

[173] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," *Proc. 2010 Int. Conf. High Perform. Comput. Simulation, HPCS 2010*, pp. 48–54, 2010, doi: 10.1109/HPCS.2010.5547150.

[174] V. C. Emeakaroha, T. C. Ferreto, M. A. S. Netto, I. Brandic, and C. A. F. De Rose, "CASViD: Application level monitoring for SLA violation detection in clouds," *Proc. - Int. Comput. Softw. Appl. Conf.*, pp. 499–508, 2012, doi: 10.1109/COMPSAC.2012.68.

[175] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE management for cloud applications," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 28–36, 2012, doi: 10.1109/MCOM.2012.6178831.

[176] S. Yang, "IoT Stream Processing and Analytics in The Fog," pp. 1–8.

[177] V. Cardellini, V. Grassi, F. Lo Presti, and M. Nardelli, "On QoS-Aware scheduling of data stream applications over fog computing infrastructures," *Proc. - IEEE Symp. Comput. Commun.*, vol. 2016-Febru, pp. 271–276, 2016, doi: 10.1109/ISCC.2015.7405527.

[178] M. Aazam, M. St-Hilaire, C. H. Lung, and I. Lambadaris, "MeFoRE: QoE based resource estimation at Fog to enhance QoS in IoT," *2016 23rd Int. Conf. Telecommun. ICT 2016*, pp. 23–27, 2016, doi: 10.1109/ICT.2016.7500362.

[179] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "Dynamic Service Placement for Mobile Micro-Clouds with Predicted Future Costs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1002–1016, 2017, doi: 10.1109/TPDS.2016.2604814.

[180] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and Y. F. Sun, "QoE-Driven Joint Resource Allocation for Content Delivery in Fog Computing Environment," *IEEE Int. Conf. Commun.*, vol. 2018-May, pp. 1–6, 2018, doi: 10.1109/ICC.2018.8422843.

[181] M. Aazam, M. St-Hilaire, C. H. Lung, and ..., "PRE-Fog: IoT trace based probabilistic resource estimation at Fog," *2016 13th IEEE …*, 2016, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7444724/

[182] V. B. C. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, and G. Tashakor, “Handling service allocation in combined Fog-cloud scenarios,” in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–5. doi: 10.1109/ICC.2016.7511465.

[183] M. Aazam and E. N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, vol. 2015-April, pp. 687–694, 2015, doi: 10.1109/AINA.2015.254.

[184] J. C. Guevara, R. da S. Torres, and N. L. S. da Fonseca, “On the classification of fog computing applications: A machine learning perspective,” *J. Netw. Comput. Appl.*, vol. 159, 2020, doi: 10.1016/j.jnca.2020.102596.

[185] M. A. Ali, A. Esmailpour, and N. Nasser, “Traffic density based adaptive QoS classes mapping for integrated LTE-WiMAX 5G networks,” *2017 IEEE Int. …*, 2017, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7996488/

[186] R. Choen, N. L. S. Fonseca, and M. Zukerman, “Traffic management and network dimensioning,” *Multimed. Commun. Networks …*, 1998.

[187] A. Callado, C. Kamienski, G. Szabó, and ..., “A survey on internet traffic identification,” *… Surv. \&tutorials*, 2009, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5208732/

[188] M. Finsterbusch, C. Richter, E. Rocha, and ..., “A survey of payload-based traffic classification approaches,” *… Surv. \&Tutorials*, 2013, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6644335/

[189] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, “Analyzing software measurement data with clustering techniques,” *IEEE Intell. Syst.*, 2004, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1274907/

[190] M. A. Bouke and A. Abdullah, “An empirical assessment of ML models for 5G network intrusion detection: A data leakage-free approach,” *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 8, no. May, p. 100590, 2024, doi: 10.1016/j.prime.2024.100590.

[191] M. Alsenwi, K. Kim, and C. S. Hong, “Radio Resource Allocation in 5G New Radio: A Neural Networks Approach,” *J. KIISE*, vol. 46, no. 9, pp. 961–967, 2019, doi: 10.5626/jok.2019.46.9.961.

[192] M. Chen, U. Challita, W. Saad, C. Yin, and ..., “Artificial neural networks-based machine learning for wireless networks: A tutorial,” *… Surv. \&Tutorials*, 2019, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8755300/

[193] E. Alpaydin, *Introduction to Machine Learning*, 3rd Editio. The MIT Press Cambridge, Massachusetts London, England, 2014.

[194] V. Vapnik, “The nature of statistical learning theory, volume Spinger-Verlag,” *New York*, 1995.

[195] C. Burges, “A tutorial on support vector machines for pattern recognition

(Knowledge discovery and data mining, Usama Fayyad)," *Kluwer*, 2000.

[196] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, *Large Margin DAGs for Multiclass Classification. Advances in Neural Lnformation Processing System 12*. MIT Press, 2000.

[197] K. Rpbert and S. Mika, "An introduction to kernel based learning algorithms [J]," *IEEE Trans. Neural Networks*, 2001.

[198] B. Schölkopf, A. J. Smola, and F. Bach, "Learning with kernels: support vector machines, Regularization," *Optim. Beyond. MIT Press*, 2002.

[199] R. Hebrich, *Learning kernel classifiers*. Mit. Press. Cambridge, MA, 2002.

[200] P. Flach, "Performance Evaluation in Machine Learning: The Good, the Bad, the Ugly, and the Way Forward," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pp. 9808–9814, 2019, doi: 10.1609/aaai.v33i01.33019808.

[201] J. Labory, E. Njomgue-Fotso, and S. Bottini, "Benchmarking feature selection and feature extraction methods to improve the performances of machine-learning algorithms for patient classification using metabolomics biomedical data," *Comput. Struct. Biotechnol. J.*, vol. 23, pp. 1274–1287, 2024, doi: https://doi.org/10.1016/j.csbj.2024.03.016.

[202] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[203] M. Aqib, D. Kumar, and S. Tripathi, "Classification of Edge Applications using Decision Tree, K-NN, & SVM Classifier," *2022 IEEE Students Conf. Eng. Syst. SCES 2022*, pp. 1–6, 2022, doi: 10.1109/SCES55490.2022.9887690.

[204] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.

[205] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, 2009, [Online]. Available: http://scholarpedia.org/article/K-Nearest_Neighbor

[206] Autoridad Nacional de Servicio Civil, 2021. Manual de Buenas Practicas en la Genstion Publica. Lima: Autoridad Nacional del Servicio Civil.

[207] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification Second Edition, Jone Wiley \&Son*. Inc, 2001.

[208] V. Vapnik and A. Y. Chervonenkis, "A class of algorithms for pattern recognition learning," *Avtomat. i Telemekh*, 1964, [Online]. Available: https://www.mathnet.ru/eng/at/v25/i6/p937

[209] B. Boser, I. Guyon, and V. Vapnik, "Pattern recognition system using support vectors," *US Pat. 5,649,068*, 1997, [Online]. Available: https://patents.google.com/patent/US5649068A/en

[210] …, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines,"

Adv. neural …, 1996, [Online]. Available: https://proceedings.neurips.cc/paper/1996/hash/d38901788c533e8286cb6400b40b386d-Abstract.html

[211] V. Vapnik, “The Nature of Statistical Learning Theory Springer, 2000,” *Google Sch. Google Sch. Digit. Libr. Digit. …*, 1995.

[212] V. N. Vapnik, “The support vector method,” *Int. Conf. Artif. neural networks*, 1997, doi: 10.1007/BFb0020166.

[213] V. Vapnik, *The nature of statistical learning theory*. books.google.com, 2013. [Online]. Available: https://books.google.com/books?hl=en%5C&lr=%5C&id=EqgACAAAQBAJ%5C&oi=fnd%5C&pg=PR7%5C&dq=vapnik%5C&ots=g5F0gB6X26%5C&sig=2pos-t6JNBwZ2HH7qYVdx_jLMa0

[214] H. K. Bharadwaj *et al.*, “A Review on the Role of Machine Learning in Enabling IoT Based Healthcare Applications,” *IEEE Access*, vol. 9, pp. 38859–38890, 2021, doi: 10.1109/ACCESS.2021.3059858.

[215] D. Liu *et al.*, “User Association in 5G Networks: A Survey and an Outlook,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2. Institute of Electrical and Electronics Engineers Inc., pp. 1018–1044, Apr. 01, 2016. doi: 10.1109/COMST.2016.2516538.

[216] Y. Sun, G. Feng, S. Qin, and S. Sun, “Cell association with user behavior awareness in heterogeneous cellular networks,” *IEEE Trans. Veh. …*, 2018, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8265220/

[217] W. Zhu, P. Xu, H. Jiang, and Y. He, “A resource allocation and cell association algorithm for energy efficiency in HetNets,” *Int. J. …*, 2019, doi: 10.1002/dac.4123.

[218] P. Han, Z. Zhou, and Z. Wang, “User association for load balance in heterogeneous networks with limited CSI feedback,” *IEEE Commun. Lett.*, 2020, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8994050/

[219] H. Z. Khan, M. Ali, I. Rashid, A. Ghafoor, and ..., “Cell association for energy efficient resource allocation in decoupled 5G heterogeneous networks,” *2020 IEEE 91st …*, 2020, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9129442/

[220] A. Mohamed, O. Onireti, M. A. Imran, and ..., “Predictive and core-network efficient RRC signalling for active state handover in RANs with control/data separation,” *IEEE Trans. …*, 2016, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7797249/

[221] V. Sharma, I. You, and R. Kumar, “Resource-based mobility management for video users in 5G using catalytic computing,” *Comput. Commun.*, vol. 118, no. October 2017, pp. 120–139, 2018, doi: 10.1016/j.comcom.2017.10.009.

[222] H. Farooq and A. Imran, “Spatiotemporal mobility prediction in proactive self-organizing cellular networks,” *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 370–373,

2017, doi: 10.1109/LCOMM.2016.2623276.

[223] A. Al-Molegi, M. Jabreel, and A. Martínez-Ballesté, “Move, attend and predict: An attention-based neural model for people’s movement prediction,” *Pattern Recognit. Lett.*, 2018, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786551830182X

[224] M. Ozturk, M. Gogate, O. Onireti, A. Adeel, A. Hussain, and ..., “A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks: The case of the Control/Data Separation Architecture (CDSA),” *Neurocomputing*, 2019, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219300438

[225] H. Wei, H. Luo, and Y. Sun, “Mobility-aware service caching in mobile edge computing for internet of things,” *Sensors*, 2020, [Online]. Available: https://www.mdpi.com/624302

[226] O. P. Babalola and V. Balyan, “Vertical Handover Prediction Based on Hidden Markov Model in Heterogeneous VLC-WiFi System,” *Sensors*, vol. 22, no. 7, 2022, doi: 10.3390/s22072473.

[227] A. Mohamed, O. Onireti, M. A. Imran, S. Member, and A. Imran, “Predictive and Core-Network Efficient RRC Signalling for Active State Handover in RANs With Control / Data Separation,” vol. 16, no. 3, pp. 1423–1436, 2017.

[228] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and ..., “Human mobility patterns in cellular networks,” *IEEE Commun. …*, 2013, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6589296/

[229] H. Abu-Ghazaleh and A. S. Alfa, “Application of mobility prediction in wireless networks using markov renewal theory,” *IEEE Trans. Veh. …*, 2009, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5342480/

[230] R. Jain, A. Shivaprasad, D. Lelescu, and X. He, “Towards a model of user mobility and registration patterns,” *ACM SIGMOBILE Mob. …*, 2004, doi: 10.1145/1052871.1052877.

[231] S. Z. Yu and H. Kobayashi, “Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden Markov model,” *IEEE Trans. Signal Process.*, 2006, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1621425/

[232] S. Manzoor, A. N. Mian, and S. Mazhar, “An LSTM-based cell association scheme for proactive bandwidth management in 5G fog radio access networks,” *Int. J. …*, 2021, doi: 10.1002/dac.4943.

[233] A. Rahmati and L. Zhong, *CRAWDAD data set rice/context (v. 2007-05-23)*. May, 2007.

[234] O. M. Ogundile, A. A. Owoade, O. O. Ogundile, and O. P. Babalola, “Linear discriminant analysis based hidden Markov model for detection of Mysticetes’ vocalisations,” *Sci. African*, vol. 24, no. December 2023, p. e02128, 2024, doi: 10.1016/j.sciaf.2024.e02128.

[235] O. O. Ogundile, A. M. Usman, O. P. Babalola, and ..., “Dynamic mode decomposition: A feature extraction technique based hidden Markov model for detection of Mysticetes’ vocalisations,” *Ecol. Inform.*, 2021, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000972

[236] G. Strang, *Linear Algebra and Its Applications 4th ed.* ttwrdcs.ac.in, 2012. [Online]. Available: https://ttwrdcs.ac.in/Mahabubabad/pdf/1701440179Strang G.-Linear algebra and its applications-Brooks (2005).pdf

[237] G. H. Golub and C. F. Van Loan, *Matrix computations*. books.google.com, 2013. [Online]. Available: https://books.google.com/books?hl=en%5C&lr=%5C&id=5U-l8U3P-VUC%5C&oi=fnd%5C&pg=PP1%5C&dq=matrix+computations%5C&ots=70DCNj0T6s%5C&sig=fs8aQZqgbmh_zUhMThDYNJrh1vs

[238] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, 1936, doi: 10.1007/BF02288367.

[239] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer (Long. Beach. Calif).*, 2009, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5197422/

[240] J. N. Kutz, *Data-driven modeling \&scientific computation: methods for complex systems \&big data*. books.google.com, 2013. [Online]. Available: https://books.google.com/books?hl=en%5C&lr=%5C&id=WZMeAAAAQBAJ%5C&oi=fnd%5C&pg=PP1%5C&dq=%22data+driven%22+modeling+scientific+computation+methods+for+complex+systems+big+data%5C&ots=BkqRzpg2Ym%5C&sig=pAAuMkSdjJOhTlsNeVC-X33qa34

[241] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[242] R. Ahmed, K. Jayasinghe, and T. Wild, “Comparison of explicit CSI feedback schemes for 5G new radio,” *2019 IEEE 89th Veh. …*, 2019, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8746469/

[243] J. D. Gonzalez-Franco, J. E. Preciado-Velasco, and ..., “Comparison of Supervised Learning Algorithms on a 5G Dataset Reduced via Principal Component Analysis (PCA),” *Futur. Internet*, 2023, [Online]. Available: https://www.mdpi.com/1999-5903/15/10/335

[244] O. P. Babalola and V. Balyan, “WiFi fingerprinting indoor localization based on dynamic mode decomposition feature selection with hidden Markov model,” *Sensors*, 2021, [Online]. Available: https://www.mdpi.com/1424-8220/21/20/6778

[245] A. Review and C. O. F. Hmm, “ICA AND NEURAL NETWORKS,” pp. 41–46, 2004.

[246] K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space,” *London, Edinburgh, Dublin Philos. …*, 1901, doi: 10.1080/14786440109462720.

[247] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear Component Analysis as a

Kernel Eigenvalue Problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998, doi: 10.1162/089976698300017467.

[248] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *… Trans. R. Soc. A …*, 2016, doi: 10.1098/rsta.2015.0202.

[249] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, pp. 169–190, 2017.

[250] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification 2nd edition," *New York, USA John Wiley\&Sons*, 2001.

[251] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 195–200, 2003, doi: 10.1109/TNN.2002.806647.

[252] J. Cui and Y. Xu, "Three Dimensional Palmprint Recognition Using Linear Discriminant Analysis Method," *2011 Second Int. Conf. Innov. Bio-inspired Comput. Appl.*, pp. 107–111, 2011, doi: 10.1109/IBICA.2011.31.

[253] M. C. Wu, L. Zhang, Z. Wang, D. C. Christiani, X. Lin, and H. Ave, "Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set / pathway and gene selection," vol. 25, no. 9, pp. 1145–1151, 2009, doi: 10.1093/bioinformatics/btp019.

[254] D. Coomans, D. L. Massart, and L. Kaufman, "Optimization by statistical linear discriminant analysis in analytical chemistry," *Anal. Chim. Acta*, 1979, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003267001835133

[255] A., Tharwat, T., Gaber, A., Ibrahim, & A.E., Hassanien, 2017 Linear Discriminant Analysis: A Detailed Tutorial. AI Communications, 30(2), pp. 169-190. doi: 10.3233/AIC-170720. [Online] Available at; https://content.iospress.com/articles/ai-communications/aic729 [Accessed 20 January 2025]

[256] J. Troya, A. Vallecillo, F. Durán, and S. Zschaler, "Model-driven performance analysis of rule-based domain specific visual models," *Inf. Softw. Technol.*, vol. 55, no. 1, pp. 88–110, 2013, doi: 10.1016/j.infsof.2012.07.009.

[257] H. Lee and S. Choi, "PCA+HMM+SVM for EEG pattern classification. Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings., Paris, France, 2003, vol. 1, pp. 541-544. doi: 10.119/ISSPA.2003.1224760. [Online] Available at: https://doi.org/10.1109/ISSPA.2003.1224760 [Accessed 20 January 2025]

[258] B. Li, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees (CART)," *Biometrics*, 1984, [Online]. Available: http://statweb.lsu.edu/faculty/li/IIT/tree1.pdf

[259] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, 1995, doi: 10.1007/Bf00994018.

[260] Y.Li, "Predicting materials properties and behavior using classification and regression trees," Materials Science and Engineering: A, vol.433, no. 1-2, pp. 261-272, 2006. [Online]. Available:

[261] L. Tsipi, M. Karavolos, G. Papaioannou, M. Volakaki, and D. Vouyioukas, “Machine learning-based methods for MCS prediction in 5G networks,” *Telecommun. Syst.*, vol. 86, no. 4, pp. 705–728, 2024, doi: 10.1007/s11235-024-01158-x.

[262] Cortes, C., Jackel, L. D., LeCun, Y. and Vapnik, V., 1994. Boosting and other ensemble methods. Neural Computation, 6(6), pp. 1289-1301. [Online] Available at: https://ieeexplore.ieee.org/document/67955389/[Access 20 January 2025].

[263] International Telecommunication Union, *Measuring the Information Society Report Volume 1 2018 ITUPublications Statistical reports*, vol. 1. 2018.