

https://dspace.org/

Deterministic fog computing architecture for 5G applications in underserved communities

Khumalo, Nosipho

2021

https://hdl.handle.net/10539/31888

Downloaded from a DSpace repository. https://dspace.org/

DETERMINISTIC FOG COMPUTING ARCHITECTURE FOR 5G APPLICATIONS IN UNDERSERVED COMMUNITIES

BY:

NOSIPHO KHUMALO

A dissertation submitted in fulfilment of the requirements for the degree of Master of Science in Engineering to the

Faculty of Engineering and the Built Environment

University of the Witwatersrand

SUPERVISOR: PROF. OLUTAYO OYERINDE – Wits University

CO-SUPERVISOR: PROF. LUZANGO MFUPE - CSIR

January 2021

Declaration

I declare that this dissertation is my own unaided work. It is being submitted to the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

.....

(Signature of Candidate)

……. day of …… year …….

Abstract

The need to cope with the continuously growing number of connected users and the increased demand for mobile broadband services in the Internet of Things has led to the notion of introducing the fog computing paradigm in fifth generation (5G) mobile networks in the form of fog radio access network (F-RAN). The F-RAN approach emphasises bringing the computation capability to the edge of the network so as to reduce network bottlenecks and improve latency. In addition, the F-RAN method is feasible enough for deployments in underserved regions. However, despite the potential, the management of computational resources remains a challenge in F-RAN architectures. Thus, this dissertation aims to overcome the shortcomings of conventional approaches to computational resource allocation in F-RANs. This research first investigates applications of machine learning (ML) techniques in fog computing and 5G networks, as well as present approaches to the resource allocation problem. The potential of ML in future wireless networks is highlighted along with the limitations of current resource allocation methods. Consequently, two resource allocation techniques are presented as a solution- a reactive algorithm based on the auto-scaling method in cloud virtualisation and a proactive algorithm based on Q-learning in reinforcement learning (RL) - along with their respective architectures for implementation. The effectiveness of the proposed resource management techniques is demonstrated through simulation modelling. The proposed reactive auto-scaling algorithm yields favourable performance results in terms of latency and throughput, compared with other systems in literature. The proposed Q-learning algorithm is more efficient than popular RL algorithms in literature and outperforms the reactive method in terms of CPU utilisation and virtual link utilisation, which demonstrates the potential of ML in 5G F-RAN architectures for resource management.

Dedication

To my mother

~ this is for you ~

Acknowledgements

Foremost, I would like to express the deepest appreciation to my research co-supervisors- Professor Luzango Mfupe and Professor Olutayo Oyerinde- for the unfailing guidance and assistance. The learning curve was steep - from telecoms and machine learning to writing and presentation skills – but their immense knowledge and mentorship styles enlightened me. I am grateful to the Council of Scientific and Industrial Research (CSIR) for providing technical and financial support for this research. I would also like to thank my colleagues for encouraging me to present my work; their insightful comments and hard questions improved my work.

Finally, I wish to acknowledge the continuous support and encouragement of my family and close friends. They kept me going and believed in me when it counted the most.

Table of Contents

Declaration ..... **i**

Abstract ..... **ii**

Dedication ..... **iii**

Acknowledgements ..... **iv**

Table of Contents ..... **v**

List of Figures ..... **ix**

List of Tables ..... **xi**

Glossary of Terms and Abbreviations ..... **xiii**

Chapter 1 – Introduction **1**

1.1 Overview 1

1.2 Problem Statement and Motivation 3

1.3 Study Objectives 4

1.4 Scope Delineation 5

1.5 Major Contributions 6

1.6 Research Outputs 7

1.6.1 Peer-Reviewed Conference Papers 7

1.6.2 Peer-Reviewed Journal Articles 8

1.7 Non Peer-Reviewed Research Outputs 8

1.8 Organisation of the Dissertation 9

Chapter 2 – Background and State of the Art **10**

2.1 Introduction 10

2.2 Fog computing Background 11

2.2.1 Fog Computing Architecture 14

2.3 5G Background 16
2.3.1 C-RAN 16
2.3.2 H-CRAN 17
2.3.3 F-RAN 18
2.4 Machine Learning Applications for Fog Computing 21
2.4.1 Supervised Learning 21
2.4.2 Unsupervised Learning 26
2.4.3 Reinforcement Learning 29
2.5 Machine Learning Applications for 5G 30
2.5.1 Supervised Learning 30
2.5.2 Unsupervised Learning 32
2.5.3 Reinforcement Learning 34
2.6 Resource Management Techniques in Fog Computing 35
2.6.1 Computing Optimisation 36
2.6.2 Decision Making 37
2.6.3 Resource Provisioning 38
2.7 Resource Management Techniques in 5G 40
2.8 Resource Allocation Techniques for 5G F-RAN 42
2.9 Conclusion 45

Chapter 3 – Mathematical Model for Fog Radio Access Network Architecture in 5G 46

3.1 Introduction 46
3.2 System Model 46
3.2.1 Delay Model 48
3.2.2 Throughput and Utilisation Model 49
3.3 Problem Formulation and Optimisation 51

3.4 Evaluation and Results 54
3.4.1 Impact of the Number of Users 56
3.4.2 Impact of User Applications 59
3.4.3 Impact of Computation Capacity 61
3.5 Conclusion 63
Chapter 4 – Reactive Auto-scaling Resource Allocation Technique 64
4.1 Introduction 64
4.2 Proposed Resource Management Architecture 65
4.3 Proposed Reactive Auto-scaling Algorithm 66
4.4 Description of Other Auto-Scaling Resource Management Frameworks 70
4.5 Evaluation Setup 71
4.5.1 Simulation 71
4.5.2 Performance Metrics 72
4.6 Performance Evaluation 73
4.6.1 Demonstration of Auto-Scaling 73
4.6.2 Impact of Latency Requirements 74
4.6.3 Data Transmitted 75
4.6.4 Latency 76
4.6.5 Throughput 77
4.6.6 User Satisfaction 78
4.7 Discussion 79
4.8 Conclusion 80
Chapter 5 – Proactive Auto-scaling Resource Allocation Technique based on Reinforcement Learning 81
5.1 Introduction 81

5.2 Proposed Reinforcement Learning Model 82

5.3 Proposed Proactive Auto-Scaling Algorithm 85

5.4 Description of Other Systems 87

5.5 Evaluation setup 87

5.5.1 Simulation Parameters 87

5.5.2 Performance Metrics 88

5.6 Performance Evaluation 89

5.6.1 Comparison with the Proposed Reactive Auto-Scaling Technique 89

5.6.2 Comparison with Other RL Techniques 92

5.7 Discussion 95

5.8 Conclusion 96

Chapter 6 – Concluding Remarks 97

6.1 Introduction 97

6.2 Statement of Initial Objectives 98

6.3 Achieved Objectives 99

6.4 Key Research Findings 99

6.5 Benefits of the Study 100

6.6 Recommendations and Future Work 101

References 102

List of Figures

Figure 2.1. Fog computing architecture 14

Figure 2.2. C-RAN architecture 17

Figure 2.3. H-CRAN architecture 18

Figure 2.4. F-RAN architecture 20

Figure 2.5. Basic principle of KNN 23

Figure 2.6. Basic principle of K-means clustering 27

Figure 3.1. F-RAN system model 47

Figure 3.2. Computation resource allocation pseudocode 54

Figure 3.3. Impact of the number of users on average latency 56

Figure 3.4. Impact of the number of users on dissatisfaction ratio 57

Figure 3.5. Impact of the number of users on average user throughput 58

Figure 3.6. Impact of the number of users on average resource utilization 59

Figure 3.7. Impact of user applications on average latency 60

Figure 3.8. Impact of user applications on dissatisfaction ratio 61

Figure 3.9. Impact of computation capacity on average latency 62

Figure 3.10. Impact of computation capacity on dissatisfaction ratio 62

Figure 4.1. Architecture of fog nodes in the proposed reactive framework 66

Figure 4.2. Reactive auto-scaling algorithm 68

Figure 4.3. Scaling procedure 69

Figure 4.4. Relationship between the number of active requests and free resources 74

Figure 4.5. Impact of latency requirements on cost efficiency 75

Figure 4.6. Average data transmitted vs number of connected users 76

Figure 4.7. Impact of network traffic load on latency 77

Figure 4.8. Impact of network traffic load on throughput 78

Figure 4.9. User satisfaction ratio comparison 79

Figure 5.1. Reinforcement learning model 82

Figure 5.2. CPU utilization comparison 90

Figure 5.3. Virtual link utilization comparison 91

Figure 5.4. Sum of latency experienced by users in proactive vs reactive system 92

Figure 5.5. Sum of latency comparison of RL systems 93

Figure 5.6. Cost efficiency comparison of RL systems 94

Figure 5.7. CPU utilisation comparison of RL systems 95

List of Tables

Table 2.1. Attribute comparison of cloud computing and fog computing 12

Table 2.2. Feature comparison of cloud computing and fog computing 13

Table 2.3. Attribute comparison of C-RAN, H-CRAN and F-RAN 20

Table 2.4. Supervised machine learning techniques for fog computing 26

Table 2.5. Unsupervised machine learning techniques for fog computing 29

Table 2.6. Supervised machine learning techniques for 5G networks 32

Table 2.7. Unsupervised machine learning techniques for 5G networks 33

Table 2.8. Reinforcement machine learning techniques for 5G networks 34

Table 2.9. Resource management techniques for computing optimisation in fog computing 37

Table 2.10. Resource management techniques for decision making in fog computing 38

Table 2.11. Resource management techniques for resource provisioning in fog computing 39

Table 2.12. Summary of resource management technique in F-RANs 44

Table 3.1: F-RAN architecture notation definitions 51

Table 3.2. User application parameters 55

Table 4.1. Notations used in the reactive auto-scaling algorithm 70

Table 4.2. Summary of resource management systems 71

Table 4.3: Simulation parameter settings 72

Table 5.1: State-action mapping 85

Table 5.2. Reinforcement learning system parameters 88

Glossary of Terms and Abbreviations

Abbreviation/Term	Description
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARPU	Average Revenue Per User
BBU	Baseband Unit
BS	Base Station
CC	Cloud Controller
CoMP	Coordinated Multi-Point
CPU	Central Processing Unit
C-RAN	Cloud Radio Access Network
CRBM	Conditional Restricted Boltzmann Machine
CRRM	Cooperative Radio Resource Management
CRSP	Collaboration Radio Signal Processing
CSI	Channel State Information
D2D	Device to Device
DL	Deep Neural Learning
DMN	Directional Mesh Network
DNN	Deep/Dense Neural Network
DRL	Deep Reinforcement Learning
DUDA	Deviation Update Decision Algorithm
EE	Energy Efficiency
EH	Energy Harvesting
EM	Expectation-Maximisation
eMBB	Enhanced Mobile Broadband
eNB/eNodeB	Evolved Node B
F-AP	Fog Computing-Based Access Point
FCD	Floating Car Data
F-RAN	Fog Radio Access Network
GAP	Gateway Access Point
GM	Gaussian Mixture Model
gNB	Gnodeb
GPU	Graphics Processing Unit
HAMRL	Heuristically-Accelerated Multi-Agent Reinforcement Learning
H-CRAN	Heterogeneous Cloud Radio Access Networks
HMM	Hidden Markov Model
HPN	High Power Node
H-SVMM	Hierarchical Support Vector Machine
IARR	Interference-Aware Radio Resource
ICA	Independent Component Analysis

IoT	Internet of Things
IP	Internet Protocol
KNN	K Nearest Neighbors
LAN	Local-Area Network
MAB	Multi-Armed Bandit
MAP	Mesh Access Point
MARLL	Multi-Agent Reinforcement Learning
MCL	Maximum Coupling Loss
MDP	Markov Decision Process
MIMO	Multiple-Input Multiple-Output
MIP	Mixed Integer Programming
MIRS	Mid Infrared Spectroscopy
ML	Machine Learning
mMTC	Massive Machine-Type Communications
MP-MAB	Multi-Player Multi-Armed Bandit
MQTT	Message Queuing Telemetry Transport
NFQ	Neural-Fitted Q-Iteration
NFV	Network Function Virtualisation
NGMN	Next Generation Mobile Networks
NIST	National Institute For Standards And Technology
NOMA	Non-Orthogonal Multiple Access
NR	New Radio
PCA	Principal Component Analysis
POMDP	Partially Observable Markov Decision Process
PU	Primary User
QCQP	Quadratically Constraint Quadratic Programming
QoE	Quality Of Experience
QoS	Quality Of Service
RL	Reinforcement Learning
RRH	Remote Radio Head
RRM	Radio Resources Management
RRS	Round-Robin Scheduling
RRU	Remote Radio Unit
SDN	Software-Defined Networking
SE	Spectral Efficiency
SIC	Successive Interference Cancellation
SINR	Signal to Interference Plus Noise Ratio
SL	Supervised Learning
SNR	Signal to Noise Ratio
SU	Secondary User
SVM	Support Vector Machine
TCA	Threshold Controlled Access
UE	User Equipment
URLLC	Ultra-Reliable And Low-Latency Communications
USL	Unsupervised Learning
VM	Virtual Machine

VN	Virtual Network
VNF	Virtual Network Function
WAN	Wide-Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local-Area Network

Chapter 1 – Introduction

1.1 Overview

The wireless technologies industry has enjoyed remarkable growth over the past decades. Since its initiation, mobile wireless communication has developed from analogue voice calls characterised by limited capacity and no security, to current fourth-generation (4G) technologies capable of providing high quality mobile broadband services with end-user data rates of up to tens of megabits per second [1].

The forthcoming ubiquity of the Internet of Things (IoT) in everyday life, has created a challenge for current cellular networks. This challenge, which is particularly eminent when considering the need to deal with the exponential amounts of data produced at the edge of the network, is further exacerbated by the current network state, which is both extremely heterogeneous and immensely fragmented [2]. Moreover, the continuously growing number of connected users and the increased demand for mobile broadband services necessitate an essential change in the way in which wireless networks are designed and modelled [3].

Fifth generation (5G) wireless network technologies are the next generation in mobile communications, beyond the current fourth generation (4G) and Long Term Evolution (LTE) mobile networks, and promise to play a crucial role in enabling a better-connected networked society. 5G is anticipated to provide new opportunities that enable us to deliver unprecedented applications and services that can support new users and devices. These applications encompass massive machine-type communications (mMTC)- also known as the Internet of Things (IoT), enhanced mobile broadband (eMBB) requiring high data rates over a wide coverage area, and ultra-reliable and low-latency communications (URLLC) with stringent requirements on latency and reliability [4], [5].

The proposed architecture for 5G, in an effort to deal with the expanding amount of user traffic and the increasing number of IoT devices, is the cloud radio access network (CRAN) architecture. In the C-RAN approach, the function of processing data is borne by a pool of centralised baseband units (BBU) inside the core network, which are characterised by a limited fronthaul[6]–[8]. In order for processing to take place in the centralised BBU pool, a high bandwidth fronthaul with low latency is required. However, the fronthaul in the C-RAN is prone to time-delay and capacity constraints, which presents several challenges for 5G applications, particularly when considering the massive traffic produced by IoT devices. Furthermore, CRAN does not exploit the storage and processing capacity of edge devices and may excessively burden the core network and consequently adversely affect the quality of service (QoS) experienced by the end users [9].

As a means to overcome the challenges in the CRAN effort, the notion of introducing fog computing in 5G RAN in the form of fog radio access network (F-RAN) has emerged as a promising architecture. The F-RAN approach emphasises bringing the computation capability to the edge of the network so as to enable a lower burden on the fronthaul and meet the demands of ultra-low-latency applications [10]. As a secondary advantage to reducing network bandwidth bottlenecks and improving latency, the F-RAN technique also bears great potential in very low Average Revenue Per User (ARPU) areas, particularly when the connection to the cloud is unavailable or limited [11]. These developing regions, which are characterised by a lack of adequate broadband infrastructure, are referred to as underserved areas.

Despite all these attempts to handle the growing demand of IoT applications, management of computational and network resources of processing entities in the 5G F-RAN (i.e. fog nodes) still remains a high priority goal for future network designs. Contrary to C-RAN resources, the resources at the network edge are inherently: (i) restricted in terms of computational resources - a constraint imposed by the limited processor size and power budget of edge devices, (ii) heterogeneous - processors with different architectures, and (iii) dynamic with

variable workloads and applications contending for the limited resources [12]. Therefore, managing resources is one of the key challenges in 5G F-RAN.

1.2 Problem Statement and Motivation

Conventional legacy approaches to computational resource allocation in virtualised networks, such as the 5G F-RAN, are static mechanisms in which a fixed resource size pool (including storage resources, computing resources, and bandwidth resource) is allocated to each fog node when the network is configured [13]. However, the dynamic nature of F-RAN resources coupled with the heterogeneity and increasing complexity of IoT applications deem static allocation mechanisms insufficient for satisfying the needs of future mobile networks and necessitate dynamic allocation approaches that can predict changes in the workload and autonomously adjust resources accordingly.

In order to design a 5G F-RAN system that is truly capable of autonomous and dynamic management, intelligent functions must be introduced across both the edge and core of the network. Such mobile edge and core intelligence can be realised by integrating fundamental notions of artificial intelligence (AI) and machine learning (ML) across the wireless infrastructure [14]. At its core, ML- as an application of AI- aims to develop models and algorithms that can learn to make decisions directly from data without following pre-defined rules [15].

Fog computing and 5G are both relatively novel research topics with growing popularity in the research community and telecommunications industry at large, and recent developments and applications of machine learning algorithms in other fields point out its great potential. Based on the literature review, there is no autonomous virtual resource allocation which allows each node to manage its compute power allocation independently, although learning-based resource allocation has been implemented in [16]–[18] through a centralised controller that makes decisions for the service provider. Most research efforts aimed at addressing the computing resource allocation problem in 5G F-RAN are commonly focused on offloading data from the resource-constrained F-RAN to the core network, in order to meet the data rate and latency demands of eMBB and

URLLC applications, respectively. For instance, the work in [19] considers offloading as a means to optimise latency. The major shortfall of the offloading approach is that it is ill-suited for networks in underserved areas, where the connection to the remote cloud is unavailable or limited. The issue of computing resource allocation for 5G F-RAN architectures in underserved regions is an area of research that has not been studied extensively in literature. Thus, this research seeks to fill the research gap.

In literature, a number of relevant works discussing 5G F-RAN architectures using LTE for eMBB and URLLC services have emerged [20]–[25]. However, mMTC applications are an area that is lesser-explored. Furthermore, complete specifications for mMTC services, which were initially planned for 3rd Generation Partnership Project (3GPP) Release 16 [26], have since been included in the scope for Release 17, which is scheduled for completion in 2021. There is a lack of studies in the area of utilising enhanced next-generation network features such as 5G New Radio (NR) to support deployment scenarios for mMTC services and applications. To that end, this research is motivated to make an effort to offer insight into tackling this issue.

The primary research questions to be addressed in this dissertation are formulated as:

- How does the integration of fog computing and ML techniques unto the 5G architecture improve mMTC services in underserved communities?
- How can machine learning techniques be utilised to efficiently address the resource allocation problem in F-RAN for 5G networks?

1.3 Study Objectives

This dissertation focuses on leveraging the capabilities of ML and fog computing in order to address the computing resource allocation problem in 5G F-RAN architectures for mMTC services in underserved communities. To this aim, the key objectives of this research are:

- To investigate how ML-based techniques have been utilised in fog computing and 5G networks to address various challenges, and their potential applications in 5G F-RAN architectures.
- To design a resource allocation architecture for mMTC applications in 5G F-RAN systems.
- To develop a ML algorithm to address the problem of dynamic and autonomous allocation of computing resources in 5G F-RAN architectures.

1.4 Scope Delineation

In this dissertation, underserved areas are defined as rural and remote areas of developing regions. Therefore, only rural and remote areas as areas that are far from large cities or towns and not heavily populated in comparison with urban and suburban areas are considered. Underserved areas may be characterised by the following [27]:

- Geographic access problems due to distance, terrain, poor quality of road/ transport network and remoteness of some rural communities.
- Lack of or inadequate basic enabling infrastructure such as regular electricity supply.
- Absence of adequate telecommunications infrastructure.
- High cost of physical access and equipment installation due to any combination of geographic issues.
- Low geographic density of target population (i.e. small village populations, in sparsely populated communities that are geographically separated from one another).
- Low income, lack of disposable income and relative poverty of rural population.
- High degrees of illiteracy in some areas.
- Low levels of awareness (if any) of the benefits of modern telecommunications leading to low current demand in some areas.
- Overall lack of funding (both public and private), etc.

This research aims to address the challenges related to inadequate basic enabling infrastructure as well as high installation and access costs. As a result, fog computing has been proposed as a potential solution. In particular, fog computing selectively moves cloud computational and communication functions close to the network edge to overcome the limitations in current infrastructure. Fog nodes leverage the potential of power-efficient protocols such as Bluetooth and Zigbee, Moreover, continuous Internet connectivity is not essential for fog-based services to work. That is, the services can work independently with low or no Internet connectivity and send necessary updates to the cloud whenever the connection is available. To address the cost barrier, the distribution and sharing of resources in the fog-computing paradigm can significantly reduce capital and operational network expenses [1]. Another distinguishing feature of fog computing is that any device with computation power can be used to perform computations. Finally, the use of multiple interconnected channels in fog computing can efficiently address challenges related to loss of connection.

1.5 Major Contributions

The main contributions of this dissertation are summarized as follows:

- **5G F-RAN resource allocation architecture:** A network architectural model for computation resource allocation in 5G F-RAN is proposed. The proposed architecture has been constructed to prevent performance degradation of the system due to one fog node's congestion affecting other nodes. An algorithm based on reactive auto-scaling is developed and implemented according to the architecture.
- **Dynamic and autonomous resource allocation model:** A reinforcement learning-based algorithm for dynamic resource management of virtualised cloud computation resources in a distributed fog computing network is devised. The proposed Q-learning algorithm decides how to allocate the limited F-RAN resources by upscaling or downscaling virtual machines based on the requirement of resources and the predicted future availability.

1.6 Research Outputs

The following peer-reviewed conference papers and journal articles have been produced from this research:

1.6.1 Peer-Reviewed Conference Papers

1. Nosipho Khumalo, O.O. Oyerinde and Luzango Mfupe. *A Review of Resource Management Advances in Fog Computing* – conference paper under preparation for submission to AFRICON 2021.

   The paper provides a comprehensive review of resource management techniques that have been proposed as a means to address various problems in fog computing.
2. N. Khumalo, O. Oyerinde and L. Mfupe, "Reinforcement Learning-based Computation Resource Allocation Scheme for 5G Fog-Radio Access Network," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, 2020, pp. 353-355, doi: 10.1109/FMEC49853.2020.9144787.

   This paper addressed the computing resource allocation problem in F-RAN by using reinforcement learning. An algorithm was proposed to optimise latency, energy consumption and cost. In this paper, the cost function is defined as the weighted sum of latency and energy consumption.
3. Nosipho Khumalo, O.O. Oyerinde and Luzango Mfupe, “A D2D-enabled Fog Computing Architecture for 5G and IoT in Underserved Areas,” in Proceedings of South Africa Telecommunication Networks and Applications Conference (SATNAC) 2019, Durban, South Africa, pp. 1-6, 1st - 4th September 2019.

   The paper proposed a fog-based architecture that exploits device-to-device communication as well as local computation, storage, and communication as a means to reduce costs and thus overcome the financial constraint in 5G deployment.
4. N. Khumalo, O. Oyerinde and L. Mfupe, "Fog Computing Architecture for 5G-Compliant IoT Applications in Underserved Communities," 2019

IEEE 2nd Wireless Africa Conference (WAC), Pretoria, South Africa, 2019, pp. 1-5, doi: 10.1109/AFRICA.2019.8843414.

This paper proposed fog computing as a means to reduce network costs in the effort to deploy 5G in underserved communities. The paper won the IEEE Best Paper Award.

1.6.2 Peer-Reviewed Journal Articles

1. 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 journal, 2021. doi: 10.1109/ACCESS.2021.3051695..

   This paper identifies the shortcomings of conventional approaches to computational resource allocation in F-RANs and presents reinforcement learning as a method for dynamic and autonomous resource allocation; an algorithm is proposed based on Q-learning.

1.7 Non Peer-Reviewed Research Outputs

1. Wireless World Research Forum. (2020).'Network Slicing to Bridge the Digital Divide'. *Network Slicing for 5G and Beyond Systems* [White Paper].
2. Nosipho Khumalo, Luzango Mfupe. *Deterministic Fog Node Architecture for 5G Applications in Underserved Communities*. Poster presented at: Emerging Researchers Symposium. 6th CSIR Conference. 28-29 June 2018; Pretoria, South Africa.
3. Nosipho Khumalo, Luzango Mfupe. *Deterministic Fog Node Architecture for 5G Applications in Underserved Communities*. Poster presented at: 40th Wireless World Research Forum. 31 May- 1 June 2018; Durban, South Africa.

1.8 Organisation of the Dissertation

This dissertation is divided into the following chapters. The literature is presented in Chapter 2 , along with the background and an overview of the related work. Chapter 3 defines the system model and formulates the resource allocation problem. A resource allocation mechanism based on reactive auto-scaling is presented in Chapter 4 along with a comparative analysis with other auto-scaling based resource allocation approaches. In Chapter 5 , a reinforcement learning model for autonomous resource allocation in 5G F-RAN is presented based on proactive auto-scaling, including the learning parameters, the proposed Q-learning algorithm and the performance evaluation. Finally, Chapter 6 concludes the dissertation with a review of the research objectives and major contributions, and a brief overview of future work.

Chapter 2 – Background and State of the Art

2.1 Introduction

The concept of incorporating the fog computing paradigm into the RAN architecture of 5G systems was driven by the need to overcome the inherent challenges presented by conventional radio access networks when confronted with the demands of applications in next generation systems [28]. In comparison to the centralised BBU pool in the C-RAN architecture, the F-RAN model significantly lowers the load on fronthaul. Therefore, the F-RAN is expected to be a feasible model for the provisioning of ultra-low latency services and applications. However, as a result of limited storage capabilities and computing of the fog nodes, the completion of delay-sensitive tasks becomes a significant research issue in terms of resource management when considering resource-constraint end-users.

This chapter presents the background and relevant work that investigates the issue of resource management in the 5G F-RAN architecture. First, an overview of fog computing is provided along with a summary of the different radio access network architectures in 5G networks. Then, the applications of machine learning techniques in the 5G and fog computing are surveyed. After discussing how different techniques are used to address resource management problems in 5G systems and fog computing, a survey of machine learning-based resource allocation techniques related to 5G F-FRAN architectures is presented. Finally, the chapter concludes with a brief discussion that serves to provide motivation for using reinforcement learning to resolve resource allocation problems in 5G F-RAN architectures.

2.2 Fog computing Background

Fog computing allows network nodes near IoT devices to provide computation, storage, data management, and networking capabilities, thus being the intermediary between the cloud and IoT devices. This way, networking, computing, data management, storage, and decision making is possible on the path from the IoT devices to the cloud, as data transmission occurs from the IoT devices to the cloud. The OpenFog Consortium [29] defines fog computing as "a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from Cloud to Things." Another term for fog computing is edge cloud computing, which is characterised by the movement of data, computing content and applications from centralised points to the logical limits of a network. In fog computing, the horizontal platform enables the distribution of computing functions between multiple application domains and industry verticals. As an extension to facilitating a horizontal architecture, fog computing provides a flexible platform to meet the data-driven needs of users and mobile network operators. The fog computing paradigm is envisioned to provide substantial support for the IoT.

This description of fog computing is congruent with the definition provided by the National Institute for Standards and Technology (NIST), IBM and Gartner. There are many other definitions of the fog computing model with slight variations, however, the key concept is that fog moves storage, compute, decision making, control, communication and nearer to the network edge where data generation takes place. It must be noted, however, that fog computing does not replace cloud computing but rather complements it to address the prominent limitations of cloud. These limitations include high latency, the requirement of high speed reliable internet connectivity with enough bandwidth, and security threats resulting from cloud services being located within the loosely controlled internet [30].

Table 2.1. Attribute comparison of cloud computing and fog computing

Attribute	Cloud computing	Fog computing
Operators	Cloud service providers	Users and cloud service providers
Availability of computing resources	High	Moderate
Distance from users	Far	Relatively close
Type of computation	High	High with low delays
Architecture	Centralised & hierarchical	Decentralised & hierarchical
Service availability	High	High
Latency	High	Low
Server installation location	Dedicated locations	Edge or in dedicated buildings
Power consumption	Relatively high	Low
Internet connectivity requirement	Throughout the duration of services	Can operate autonomously with little or no Internet connectivity
Hardware connectivity	Wide Area Network (WAN)	WAN, Local Area Network (LAN),Wireless Local Area Network (WLAN), Wireless Fidelity (Wi-Fi), cellular
Service access	Through the core	Through connected devices from the edge to the core

Table 2.1 and Table 2.2 compare the attributes and features of fog computing to cloud computing. From these visuals, it is evident that fog computing is superior to the conventional cloud model for energy-efficient applications that require low latency, real-time interactions and local server nodes distributed across wide geographical areas.

Table 2.2. Feature comparison of cloud computing and fog computing

Feature	Cloud computing	Fog computing
Heterogeneity support	Yes	Yes
Geographically distributed	No	Yes
Ultra-low latency provision	No	Yes
Support for mobility	No	Yes
Support for real-time applications	No	Yes
Support for large-scale applications	Yes	Yes
Virtualisation support	Yes	Yes
Multiple IoT applications	Yes	Yes

As a result of the characteristics presented in Table 2.1, fog computing is progressively attracting attentions to address the issues of practical usage hindering underserved regions. For developing countries such as South Africa, which are characterised by wide gaps of development between urban and rural areas [11], there is an urgent need for a solution that could address issues like tight distribution of current adjacent resources, limited infrastructural facilities, and providing a means for remote regions to connect to the outside world.

In comparison to cloud computing, fog computing focuses on the edges of network resources, which has some notable benefits [31]:

1. Speed of data transmission: A fog network is created within a specific area, which increases the data transmission rate between devices due to reduced communication distances.
2. Sharing of storage capacity: Fog computing enables users to safely store their data on nearby devices, thus prolonging the buffer capacity of each device if availability is guaranteed.
3. Resource friendliness and cost-effectiveness: The network costs for configuring a fog network are substantially lower in comparison to a cloud network. In addition, there is more flexible bandwidth utilisation which

can be reserved for specific needs, since not all the information is being processed at the same time and/or through the same node.

These distinctive qualities suggest that fog computing is a more appropriate approach for budget constrained, smaller scale coverage, and environments requiring real-time streaming. Furthermore, the effective usage of fog computing should allow underserved regions to progress at a higher speed and with reduced cost.

2.2.1 Fog Computing Architecture

Different architectural models have been recommended for fog computing in the past years, which are derived mostly from the basic three-tiered structure. Fog computing introduces an intermediate layer between IoT devices and the cloud, as a means to provide the network edge with cloud services [32]. An overview of the architecture of fog computing is shown in Figure 2.1.

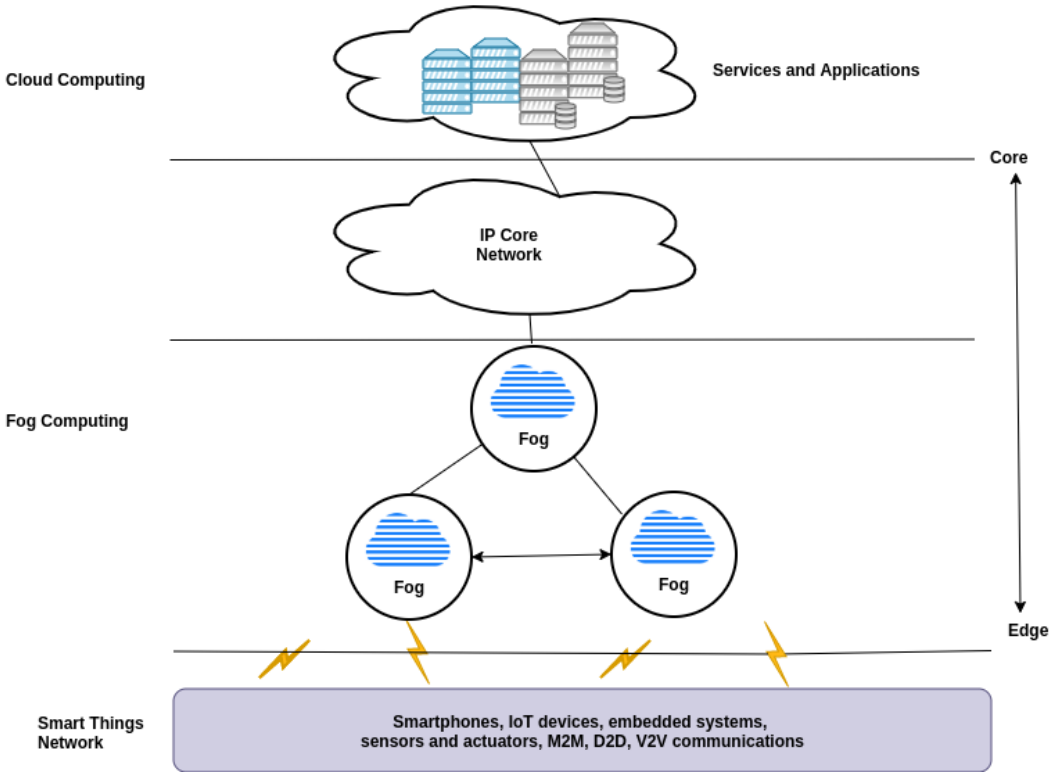


Figure 2.1. Fog computing architecture

The key elements of the architecture are discussed below [33]:

- At the bottom level of the architecture is a network of various smart end user devices such as IoT devices, smart phones, sensors, embedded systems, and actuators. These devices, which are usually distributed geographically, are responsible for collecting data from the sensors in the physical environment and communicating it to the fog nodes.
- The next level above is the fog computing layer that comprises multiple fog nodes such as switches, routers, base stations, gateways, access points, etc. that are widely distributed between the cloud and the IoT devices. Nodes located at or nearer to the edge are normally responsible for collecting sensor data from the IoT devices, data normalisation, and control/command of actuators and sensors.
- The following upper tier contains nodes that are responsible for data transformation, data compression, and data filtering. In addition, these nodes may also focus on some provision of latency-sensitive and real-time analysis applications. Nodes at the upper tiers or closest to the remote cloud normally assume the role of data aggregation and conversion of data into knowledge. The edge nodes are linked to the backend cloud via the Internet Protocol (IP) core network to attain more powerful processing and storage services.
- At the peak of the hierarchical architecture is the cloud computing tier which comprises numerous storage devices and high performance servers and storage devices that primarily focus on performing batch storage and extensive computing of enormous amounts of data. Contrary to the traditional cloud-only model where all processing takes place on the cloud, some storage and computing tasks do not go through the cloud. Instead, substantial amounts of computation are carried out by edge devices on the local area network, while computationally intensive or delay tolerant tasks can be offloaded to the cloud.

2.3 5G Background

The Next Generation Mobile Networks (NGMN) Alliance defines 5G as “an end-to-end ecosystem to enable a fully mobile and connected society. It empowers value creation towards customers and partners, through existing and emerging use cases, delivered with consistent experience, and enabled by sustainable business models” [34]. 5G mobile networks are expected to handle six challenges which are not adequately addressed by current mobile networks: massive device connectivity, reduced end-to-end latency, higher data rate, higher capacity, reliable Quality of Experience (QoE) provisioning, and lower capital and operations cost [35].

2.3.1 C-RAN

As a means to realise the abovementioned objectives, the idea of integrating cloud computing into radio access networks (RANs) has emerged to form the cloud radio access network (C-RAN) [36]. However, the C-RAN architecture introduces its own challenges in the baseband unit (BBU) pool, which is centralised and characterised by limited fronthaul. In order for processing to take place in the centralised BBU pool, a high bandwidth fronthaul with low latency is required. However, the fronthaul in the C-RAN is prone to time-delay and capacity constraints, which is significantly detrimental to energy efficiency (EE) and spectral efficiency (SE) gains.

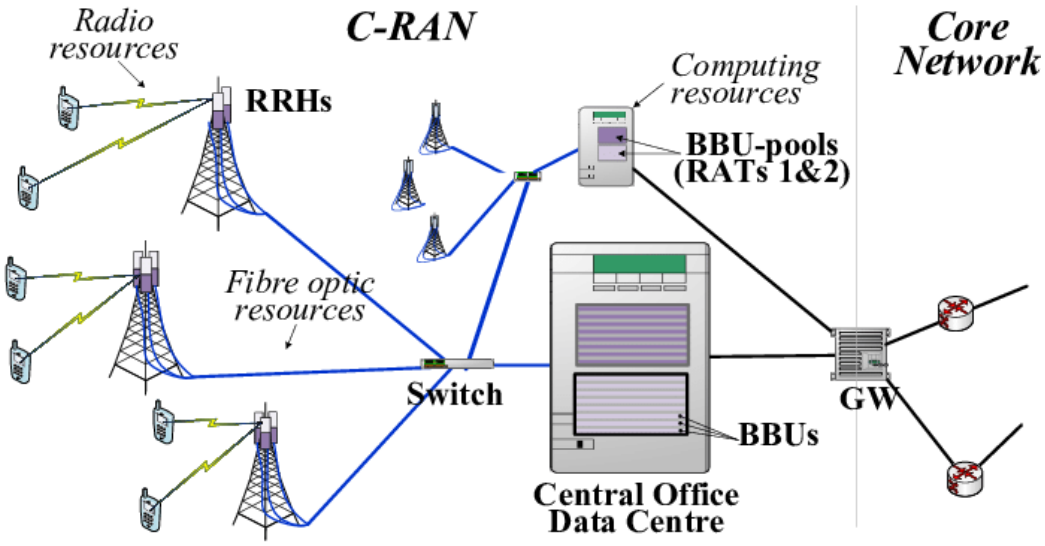


Figure 2.2. C-RAN architecture

2.3.2 H-CRAN

As a means to tackle the inherent challenges of the C-RAN architecture posed by fronthaul restraints, heterogeneous cloud radio access networks (H-CRANs) are the recommended method [37]. H-CRANs are characterised by the decoupled control and user planes implemented through the use of high power nodes (HPNs) that are responsible for performing control plane functions and enabling continuous coverage. Furthermore, H-CRANs deploy remote radio heads (RRHs) in the user plane, which are focused on providing the high speed data rates required for data packets to be transmitted. Backhaul links are used to connect HPNs to the BBU pool in order to facilitate interference coordination. However, the implementation of H-CRANs introduces several challenges. Firstly, the excessive amount of traffic data repetition between the centralised BBU pool and RRHs further aggravates the fronthaul constraints. Furthermore, making use of the storage and processing abilities of user equipment (UEs) and RRHs offers advantage towards alleviating the load of the BBU pool and the fronthaul, however H-CRANs do not fully exploit this. Finally, the need for mobile network operators to install a large number of fixed HPNs and RRHs for handling peak capacity requirements is deemed extremely uneconomical when the traffic is not necessarily huge. The need to address these challenges necessitates the

investigation of innovative methods including advanced technologies and new radio access network architectures.

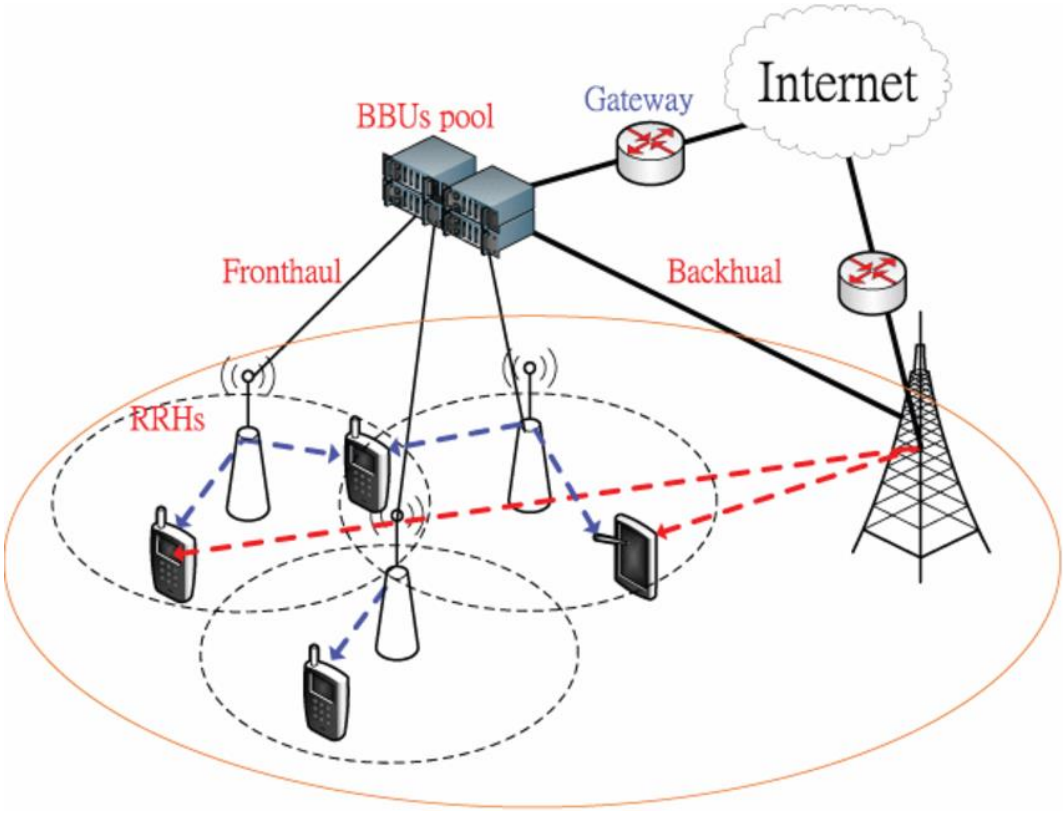


Figure 2.3. H-CRAN architecture

2.3.3 F-RAN

The fog computing approach is based on introducing the collaboration radio signal processing (CRSP) to RRHs and IoT devices or smart UEs, contrary to the H-CRAN approach of implementing it exclusively in the centralised BBU pool. Another characteristic of fog computing is that it takes full advantage of the on-device processing and cooperative radio resource management (CRRM) of IoT devices, along with distributed storage capabilities. Motivated by these specialties, the F-RAN architecture has developed, which aims to overcome the current barriers of H-CRANs and exploit CRRM of IoT devices at the edge, CRSP capabilities and local caching.

There are many evident benefits to the F-RAN approach, such as the reduced fronthaul and BBU pool load, adaptive CRRM functions at the IoT devices and

real-time CRSP functions. In addition, F-RANs leverage device to device (D2D), centralised collaboration, and distributed management in order to achieve reliable QoE and QoS. In order to integrate the fog paradigm in edge devices, the functions of the conventional RRH are developed by embedding CRRM, CRSP, and local caching capabilities, thus transforming it into a fog computing access point (F-AP)

The development of the F-RAN system architecture from C-RAN is presented in Figure 2.2 and Figure 2.3, as adapted from [38] and [39], respectively. The C-RAN and H-CRAN architectures are characterised by centralised CRSP and storage functions. On the other hand, the two architectures are differentiated by the evolution of the control function from the centralised BBU pool in the C-RAN approach to the HPNs in the H-CRAN model. The C-RAN and H-CRAN architectures are both prone to significant transmission delays and an overburdened fronthaul, which can be resolved by performing some processing at the RRHs and UEs, thereby ceasing redundant transmission of data to the BBU pool. In the meantime, the transmission delays can be successfully reduced by exploiting the local caching capabilities of RRHs.

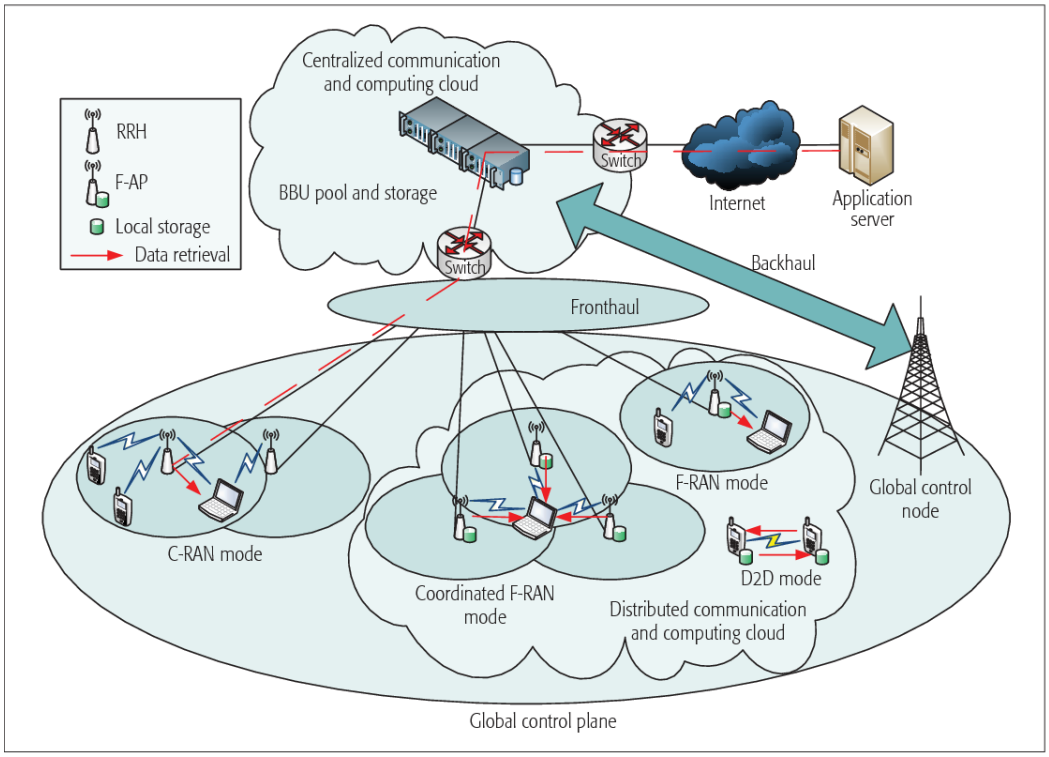


Figure 2.4. F-RAN architecture

As shown in Figure 2.4 [9], the logical fog computing tier includes certain distributed communication and computing capabilities. The proposed F-RAN model consists of four types of clouds that are either centralised or distributed and global or logical. In particular, these are the distributed logical communication cloud (performs CRSP and CRRM, situated in F-APs), global centralised communication and computing cloud (identical to the centralised cloud in C-RANs), distributed logical storage cloud (local caching, located in edge devices), and centralised control cloud (performs control plane functions in HPNs).

Table 2.3 compares the attributes of the C-RAN, H-CRAN and F-RAN architectures.

Table 2.3. Attribute comparison of C-RAN, H-CRAN and F-RAN

Attribute	**C-RAN**	**H-CRAN**	**F-RAN**
Fronthaul and BBU pool load	Heavy	Moderate	Low
Traffic Characteristic	Packet service	Packet service, real-time	Packet service, real-time

		voice service	voice service
Latency	High	High	Low
Complexity	Low in RRHs and UEs, high in the BBU pool,	Low in RRHs and UEs, high in the BBU pool	Moderate in the BBU pool, F-APs, and UEs
CRSP and caching	Centralised	Centralised	Both distributed and centralised
User and control plane decoupling	No	Yes	Yes
Constraint	Fronthual	Fronthual and backhaul	Backhaul
CRRM	Centralised	Centralization, and Distribution between the BBU pool and HPNs	Both distributed and centralised

2.4 Machine Learning Applications for Fog Computing

Machine Learning (ML) is a type of Artificial Intelligence that enables algorithms and models to become more accurate at making decisions without being explicitly programmed by rules [40]. ML tasks can be classified by their learning type as either supervised or unsupervised, or regression, classification, dimensionality reduction and clustering, depending on the learning model. Furthermore, the learning model applied to task execution can be used for classification.

In this section, the state-of-the-art of how ML techniques have been applied to fog computing is presented. The literature is classified according to the three ML categories: namely unsupervised learning (USL), supervised learning (SL), and reinforcement learning (RL).

2.4.1 Supervised Learning

Supervised learning (SL) is a learning type in ML whose objective is to learn a function that maps an input to an output based on a labelled data set. In particular, the tasks in supervised learning primarily generate two kinds of outputs: classification and regression [41]. The following supervised learning techniques are implemented in reviewed literature.

2.4.1.1 Techniques

a) **Markov model:** A Markov model is used to define the probabilities of states and the transitioning rates between them. The Markov approach is based on the presupposition that the probabilities of states are only affected by the history of states visited in the past. According to a First-Order Markov model, the probability of a future transition relies only on the current state.

b) **Support Vector Machine:** The support vector machine (SVM) model is a method for linear classification or regression that uses a $p - 1$ hyperplane to perform data transformations and then separate data. In other words, given labelled training data, the SVM model outputs an optimal hyperplane that categorises new data points. The objective of the SVM model is to find a hyperplane in a p-dimensional space, where p represents the number of features, that classifies the data points distinctly. The ideal plane is the one with the maximum margin, i.e. the maximum distance between data points between the two classes.

c) **Cascading** is a type of ensemble learning that integrates numerous classifiers to train a model. The key characteristic of this model, as pointed out by its name, is the cascading nature of learning and training. In other words, cascading consists of several stages, where each stage is an ensemble of weak learners. The output from a certain stage is used as additional information as part of the input for the next stage [42].

d) **K Nearest Neighbours:** K Nearest Neighbours (KNN) is a regression and classification model classifies data points based on similarity with other points. KNN is considered a non-parametric (i.e. makes no assumptions about the data structure) and lazy learning (makes no generalisations and involves minimal training) algorithm. Considering the classification instance, which is shown in Figure 2.5, the premise of the KNN model is to make an educated guess on a data point's class based on its K nearest neighbours' majority voting.

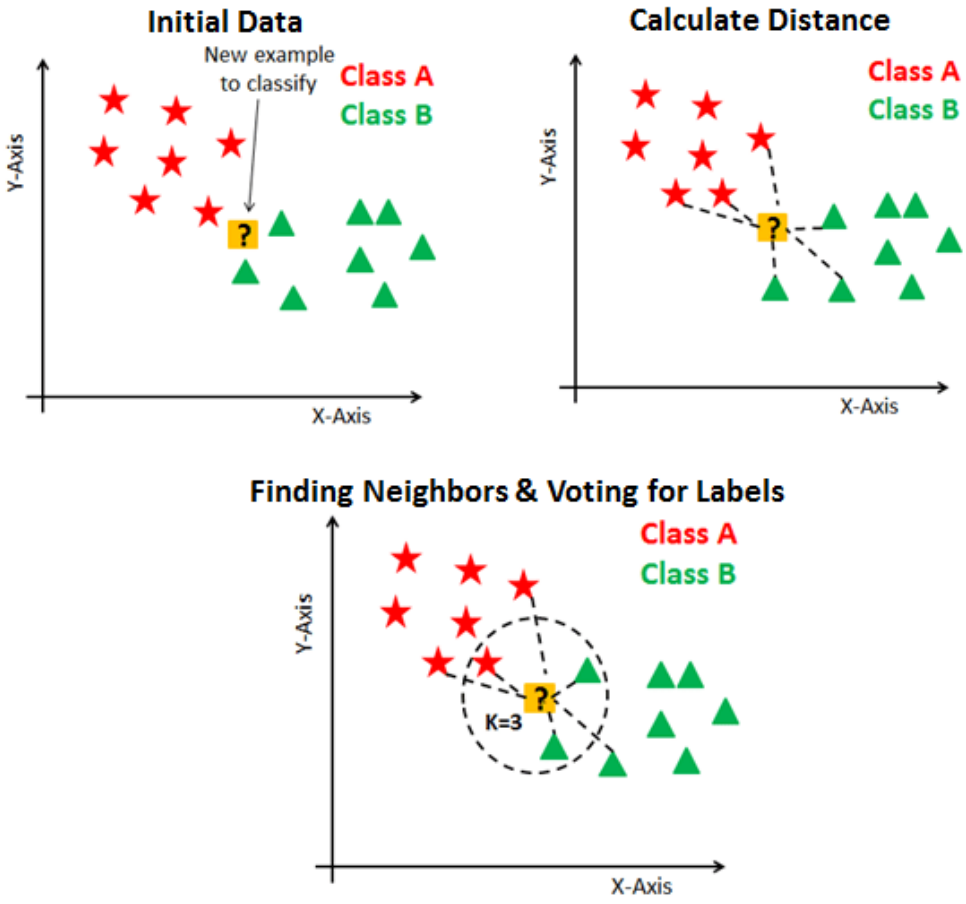


Figure 2.5. Basic principle of KNN

e) **Artificial Neural Networks:** Artificial Neural Networks (ANN) describe the method for information processing and are modelled based on how the biological nervous system creates patterns, learns, makes decisions and perceives information. In other terms, they are a simple mathematical model of the human brain which is used to process non-linear relationships between inputs and outputs. ANNs can be applied for both classification and regression tasks.

f) **Regression:** Regression analysis is used to examine relationships between variables. The main objective of the regression model is to employ statistical methods to find data patterns or estimate the value of the independent variables, given the dependent variables.

g) **Bayesian network:** The principle of Bayesian networks is to determine the probabilities and relationships between random variables. Bayesian

networks are a category of graphical models where the relationships between the nodes and edges are modelled as a directed acyclic graph.

2.4.1.2 Applications

Regression models are used to estimate or predict network parameters. For instance, the work in [43] presented a method for privacy protection of sensor data in a fog network inspired by linear regression. The mechanism proposed successfully attains high precision while maintaining the privacy of users. Similarly, regression was used in [44] for predictive analytics to determine the viability of deploying big data analytics near the edge of the network. By employing regression techniques in a fog computing context, one can realise substantial improvements in terms of prediction accuracy, and training and prediction times, with the latter being an indication that regression models are suitable for deployment in latency-sensitive environments.

ANNs are well-known for their ability to learn and adapt. The work in [45] attested to the efficiency and effectiveness of ANNs in fog-based wireless sensor networks. Here, ANNs were used to accurately track the location of sensor nodes in closed indoor and outdoor environments in a cost-efficient manner. Deep ANNs, also known as deep neural networks (DNNs) or deep learning, describe a relatively novel domain of ML that allows information processing models consisting of numerous computing layers to perform difficult data representations through the aid of many layers of abstraction. In [46], DNNs were deployed in a decentralised fog-based infrastructure to provide traffic modelling and forecasting capabilities. Distributing algorithms in the fog system has significant gains over the centralised cloud approach. More specifically, the model exhibits better behaviour in the fog network, especially in scenarios with connectivity issues or long and infrequent power outages by protecting the system against backhaul issues. This could be especially beneficial in rural network deployments where network performance is constrained by lack of power or environmental barriers. As a result of the theory of deep neural networks and the enhancement of Graphics Processing Unit (GPU) hardware, a novel mechanism inspired by deep neural learning (DL) presented in [47] has become increasingly popular. The

proposed DL-inspired scheme performs better than conventional mechanisms, as illustrated by the experimentation results.

The work in [48] appraised image recognition application in edge networks. Through the Markov model technique, prediction was used to pre-fetch certain segments of the trained classifiers used for identification and used smaller models to accelerate recognition. Results showed that Markov models yield significant improvements in terms of recognition latency, scalability, network utilisation and accuracy. The cascade classifier is also effective for image recognition application in fog computing, as demonstrated in [49].

End-to-end delay forecasting and estimation for edge traffic engineering were investigated in [50] via Bayesian Networks. In this work, the issue of traffic engineering path selection at the edge was formulated through a risk minimisation method based on mean-variance analysis in economics. Furthermore, machine learning techniques were leveraged to determine the risks of path selection. The results suggested that in scenarios where there are relationships between the data points representing path delay, then the Bayesian network performs well in terms of peak latency estimation.

The Support Vector Machines (SVM) and K-Nearest Neighbour (KNN) models are mainly utilised for the classification of points or objects. In [51], SVM was applied to the identification of diseases, while [52] investigated anomaly detection. Both works demonstrated the efficacy of SVM for detection applications in fog computing. The work in [53] successfully adopted KNN for QoS prediction.

In Table 2.4, the applications of supervised learning techniques in fog computing are summarised.

Table 2.4. Supervised machine learning techniques for fog computing

Learning technique	**Key characteristic**	**Application in fog computing**
Regression models	Estimate variables' relationships	Predictive analytics
K-Nearest Neighbour	Majority vote of neighbours	Estimation or detection of network parameters
Support Vector Machines	Non-linear classification	Anomaly or fault intrusion detection Disease detection
Bayesian learning	***a*** posteriori probability	Traffic engineering
Artificial Neural Networks	- interconnected processing units - DNN	- User location forecasting - Anomaly or fault intrusion detection

2.4.2 Unsupervised Learning

Unsupervised learning (USL) is a learning type in machine learning whose goal is to identify hidden patterns in unlabelled data. The following unsupervised learning algorithms were applied in literature.

2.4.2.1 Techniques

a) **K-Means Clustering Algorithm:** The basic philosophy of is to divide the data points into *K* clusters, such that every data point is a member of the cluster containing the nearest mean. The fundamentals of the K-means clustering algorithm, which is grounded in multiple stages of iterative modification, are shown in Figure 2.6. K means are first initialised randomly at the start of the process. Thereafter, every data point is iteratively allocated to a cluster until the members in every cluster are constant.

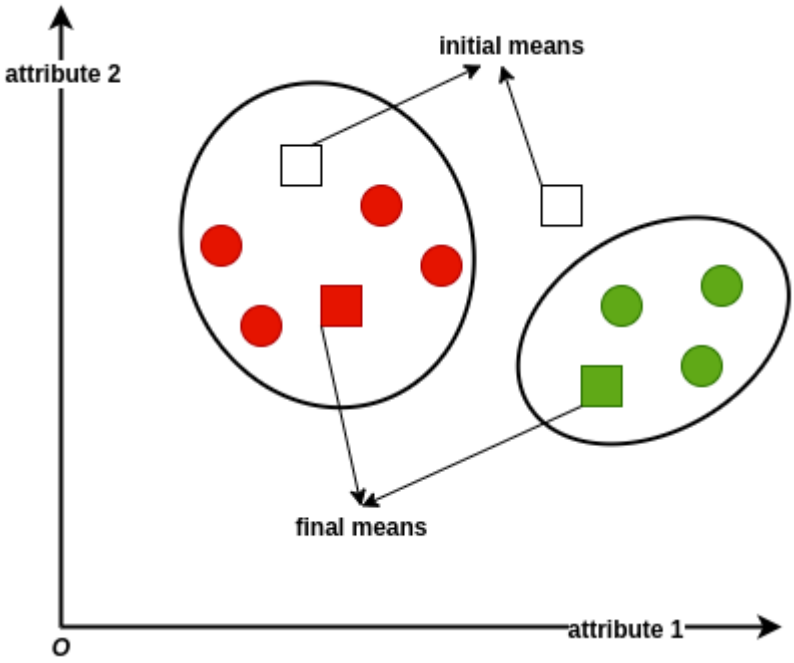


Figure 2.6. Basic principle of K-means clustering

b) **Auto-encoder:** An auto-encoder describes a type of unsupervised neural network that maps an input x into an output x' using a representation of the data. The objective of the auto-encoder is to minimise a loss through adequate training.

c) **Deep neural network:** A deep/dense neural network (DNN) describes a type of artificial neural network (ANN) characterised by numerous levels between the input layer and the output layer. DNNs can be applied in supervised and unsupervised learning. In supervised learning, where labelled data is available, DNNs are generally applied to classification and regression problems. In unsupervised learning, there is no labelled data set, therefore the neural network does not learn based on feedback [54].

d) **Spectral clustering** is an unsupervised learning method that uses the eigenvalues of the similarity matrix $S_{ij} = s(x_i, x_j)$of the data points x_i, x_j. Spectral clustering first executes dimension reduction, then uses basic clustering methods such as k-means clustering to group the similar data into clusters according to the low-dimensional space. The well-known and

most applied spectral clustering measures of similarity are inspired by the Gaussian kernel and the Euclidean distance [55].

2.4.2.2 Applications

One of the common problems in heterogeneous scenarios associated with diverse network resources and end user devices is clustering. The work in [56] used K-means clustering for smart telehealth monitoring to detect and analyse speech disorders in patients with Parkinson's disease. The proposed approach, which translated big data handling and processing from the backend cloud to fog devices, was illustrated to be promising for resource-constrained environments in terms of device power consumption, central processing unit (CPU) usage and memory usage. In [57], a soft clustering algorithm was derived for determining the locations of edge nodes in the work. The presented approach performed marginally superior to the Voronoi tessellation model, however, the performance improved with bandwidth up to a certain point, after which the latency saturated and did not decrease further. Despite the simplicity and efficiency of the clustering algorithms, the major drawback is that it is not clear how one should determine the value of K (the optimal number of fog nodes), and the solution relies on heuristics rather than mathematical analysis [58]. The issue of defining the optimal number of fog nodes dynamically using learning techniques is an open area of research.

Spectral clustering is an appropriate technique for identifying similarities among data points through the application of graph similarity. Therefore, it can be adapted to divide network graphs into smaller and smaller sub-graphs. In [59], this concept was explored by grouping logically similar fog devices into individual groups optimised for special kinds of IoT application, such as compute or memory intensive. The clustering processing, which facilitated the formation of functional areas, was efficient in terms of achieving high scalability at low latencies.

A federated approach to anomaly detection that uses auto-encoders and specialised deep learning neural networks, deployed on edge devices to perform

analytics and identify anomalous observations in a distributed manner, was proposed in [60]. The auto-encoders simultaneously learned from the new observations in order to identify new trends. The proposed approach showed encouraging performance improvements in terms of reduced bandwidth and reduced connectivity requirements.

The applications of unsupervised learning techniques for fog computing are summarised in Table 2.5.

Table 2.5. Unsupervised machine learning techniques for fog computing

Learning technique	**Key characteristic**	**Application in fog computing**
K-means clustering	K partition clustering	Disease identification
Spectral clustering	Similarity-based sub-graph	Fog node clustering
Neural networks	- Auto-encoders - DNN	Anomaly detection

2.4.3 Reinforcement Learning

Reinforcement learning (RL) is an iterative process in machine learning whose goal is to maximise some long-term objective. In RL, learning is achieved through interactions with the environment [61]. The following reinforcement learning techniques are utilised in surveyed works.

2.4.3.1 Techniques

In reinforcement learning, one of the most popular and widely-used algorithms is Q-learning, which is based on an agent interacting with the environment in order to learn the Q-values, which is the discounted cumulative reward. After learning the Q-values, the agent can make a decision about which action to take in the current state, which is often the one with the maximum Q-value.

2.4.3.2 Applications

The work in [62] proposed a load balancing mechanism as a means to tackle the unpredictability of task demands. As a solution, a reinforcement learning based decision-making approach was applied to determine the optimum offloading decision with an unknown reward and transition function. The proposed Q-learning algorithm surpassed least-queue, nearest and random offloading selection mechanisms to minimise the computing delay and the overall overloading probability while guaranteeing convergence in polynomial time.

The utilisation of machine learning techniques in fog computing is summarised below:

- Machine learning has great potential in fog computing systems. Specifically, machine learning techniques are valuable for a variety of problems ranging from anomaly detection, predictive analytics and traffic engineering, to application-specific issues such as disease identification in health services.
- Supervised and unsupervised learning have been used extensively in fog systems, while reinforcement learning is a domain that is much lesser explored. The application of reinforcement learning techniques for decision making demonstrates its potential for resource management solutions.

2.5 Machine Learning Applications for 5G

In this section, the state-of-the-art of how machine learning techniques have been utilised in 5G systems is presented. The literature is categorised according to the three ML classifications: namely unsupervised learning (USL), supervised learning (SL), and reinforcement learning (RL).

2.5.1 Supervised Learning

2.5.1.1 Techniques

- KNN

- Regression Models
- SVM
- Bayesian Learning

2.5.1.1 Applications

SL models can be employed to predict or estimate radio parameters linked to particular users, for instance in massive multiple-input multiple-output (MIMO) systems. A hierarchical SVM (H-SVM), which is characterised by a finite number of SVM classifiers in a multi-level structure, was presented in [63]. The proposed mechanism was implemented in a MIMO system to approximate the noise statistics of the Gaussian channel.

SVM and KNN are both suitable candidates for addressing the issue of determining optimum handover methods, especially in heterogeneous networks characterised by frequent handovers. These models can also be applied to learn usage trends, as illustrated by the work in [64], and further advanced by utilising location-specific prediction. Moreover, the authors in [64] used real data for experimentations, however the user profiles are not available for use by the public. Their results demonstrated that KNN algorithms are effective for accurate estimation of energy demand by up to ninety percent.

For cognitive spectrum prediction capabilities requirements in future wireless networks, the Bayesian learning techniques can be relied on. For instance, the work in [65] used these models in a massive MIMO scenario to accurately learn and predict the channel attributes. In particular, the learning and prediction were performed using a Gaussian mixture (GM) and the expectation-maximisation (EM) models, respectively.

Another application of Bayesian learning for cognitive spectral estimation was demonstrated by the work in [66], where the spectrum sensing issue is addressed through EM. The EM algorithm repeatedly tries to learn a maximum a posteriori approximation in the presence of incomplete or hidden data points. Unlike [66], the work in [67] used the EM technique as part of a hidden Markov Model (HMM) process to determine the exact channel attributes, as opposed to an

estimation. Bayesian learning models were also applied in cognitive radio networks by the work in [68] for the purpose of identifying network attributes and trends in traffic data.

Table 2.6 summarises the key features of supervised learning techniques in 5G networks and how the algorithms have been applied in literature.

Table 2.6. Supervised machine learning techniques for 5G networks

Learning technique	**Key characteristic**	**Application in 5G systems**
Regression models	• Predict relationships between variables • Linear and logistics regression	Energy learning
K-nearest neighbour	non-parametric and lazy learning	Energy learning
Support vector machines	• Maximum margin hyperplane • Classification or regression	MIMO channel learning
Bayesian learning	• $\boldsymbol{a}$ posteriori estimate • EM, GM and HMM	• Massive MIMO learning • Cognitive spectrum learning

2.5.2 Unsupervised Learning

2.5.2.1 Techniques

- K-Means Clustering
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)

2.5.2.1 Applications

One of the key challenges for heterogeneous 5G networks is clustering, prominently in environments with WiFi, micro and macro cells, and device-to-device communications. The work in [69] applied k-means clustering as a means to cooperatively optimise the division of mesh access points (MAPs) and allocation of virtual channels.

The ICA and PCA have been demonstrated to be effective models for signal processing. In wireless sensor and mesh networks, these techniques show great promise for applications related to issues with fault, anomaly and intrusion detection. Moreover, they may be used in cognitive radio networks to categorise the users' behaviours, as demonstrated by the work in [70], where the efficacy of ICA and PCA for data security and transmission efficiency improvement was illustrated. Another appropriate application of the ICA was identified in [71], where an iterative Binary ICA algorithm was devised to learn primary users' behaviours under hidden signal sources.

Table 2.7 presents the key features of unsupervised learning techniques in 5G networks and how the algorithms have been applied in literature.

Table 2.7. Unsupervised machine learning techniques for 5G networks

Learning technique	**Key characteristic**	**Application in 5G systems**
PCA	Orthogonal transformation	Smart grid
K-means clustering	• ***K*** partition clustering • Iterative updating algorithm	Heterogeneous Networks with D2D, WiFi and diverse cell sizes
ICA	Reveal latent independent factors	Spectrum learning in cognitive radio

2.5.3 Reinforcement Learning

2.5.3.1 Techniques

- Multi-Armed Bandits (MAB)
- Q-Learning
- Partially Observable Markov Decision Process (POMDP)
- Markov Decision Process (MDP)

2.5.3.1 Applications

MDP/POMDP techniques are powerful methods for facilitating decision making processes in future wireless networks. One traditional utilisation of these techniques is energy harvesting (EH), such as [72]. However, POMDP models are obstinate in terms of computing a precise solution, as opposed to an approximation. Q-learning is another classical use of reinforcement learning in heterogeneous networks, usually combined with the abovementioned MDP/POMDP techniques. For instance, the work in [73] considered automation and self-optimisation of femtocells as a means to address resource management and interference coordination issues. An additional example is in [74], where Q-learning was illustrated to produce remarkable performance enhancement.

The MAB and multi-player multi-armed bandit (MP-MAB) models are advantageous in wireless networks for addressing complex resource management challenges, particularly under unknown channel parameters. For example, the work in [75] models the problem of channel selection in D2D networks as an MP-MAB game, with D2D users representing players in the game.

Table 2.8 summarises the key features of reinforcement learning techniques in 5G networks and how the algorithms have been applied in literature.

Table 2.8. Reinforcement machine learning techniques for 5G networks

Learning technique	**Key characteristic**	**Application in 5G systems**
Q-learning	- Unknown system transition model	Femtocells

	- Q-function maximisation	
MDP/POMDP	- Bellman equation Maximization - Value iteration algorithm	Energy harvesting
Multi-armed bandit	- Exploration vs. Exploitation - Multi-armed bandit game	D2D networks

The findings from this section are summarised as:

- The objective of supervised learning techniques is to learn a function that maps an input to an output based on a labelled data set. In 5G networks, these techniques have been applied extensively to solve spectrum sensing and channel estimation problems.
- Unsupervised learning techniques, on the other hand, focus on identifying hidden patterns in unlabelled data. In 5G networks, PCA and ICA are the commonly adopted unsupervised learning methods for discovering hidden structures among attributes. Clustering methods can also be used to identify anomalies or faults.
- Finally, reinforcement learning techniques, which learn from interacting with the environment, are useful for decision making. The popular method in this category is Q-learning.

2.6 Resource Management Techniques in Fog Computing

A fog computing-based network is characterised by different kinds of data-generating IoT devices that require processing, and sometimes real-time processing. With each request made by these devices for data processing, resources are utilised [76]. Thus, resource management is a requirement for fog computing [77]. In this section, relevant literature that used ML in fog computing resource management is reviewed.

2.6.1 Computing Optimisation

The authors in [78] proposed a linear regression algorithm for faster and more balanced data computing at the edge of the network. Likewise, the work in [79] presented algorithms based on ML for autonomous management of computing resources. ML methods were also implemented by the work in [80] for music recognition and cognition, while guaranteeing efficient computing resource allocation. The work in [81] set out to address the issue of big data management through DL and succeeded in developing an effective system for efficient big data processing. Correspondingly, the work in [46] derived a data distribution algorithm for Floating Car Data (FCD). The proposed algorithm was able to circumvent the issue of data loss in the face of connectivity outages. Furthermore, the authors used the output of the data distribution algorithm as the input to conditional restricted Boltzmann machines (CRBMs) in order to favour distributed data modelling. The work in [82] used PCA to compress mid infrared spectroscopy (MIRS) data as part of compressed learning. When used effectively, MIRS-inspired compressed learning can be significantly advantageous for big data processing and fog computing, including reduced application delay, efficient bandwidth utilisation in rural networks and optimised communication and computation energy efficiency. The work in [83] presented cognition-inspired communications, which are initiated from AI computing, as a means to provide network analytics. In addition, the paper considered energy efficiency and resource allocation in virtualised networks. Likewise, the work in [19] aimed to leverage the capabilities of ML to minimise delay and energy consumption. The work in [51] presented an architecture for IoT-based health monitoring systems that enabled hierarchical partitioning, while performing ML-inspired data analytics.

Table 2.9 summarises the studies that adopted ML techniques as a means to optimise computing operations. To conclude the application of machine learning models in fog computing, classification techniques in SL have been used extensively for healthcare and other applications with low latency requirements. Clustering approaches in USL can be applied to wide range of IoT and mMTC

services, including traffic modelling and smart farming. In fog computing, SL and USL techniques are effective fog optimising computing capabilities at the edge of the network. This is demonstrated in Table 2.9, where it is highlighted that there are numerous techniques in machine learning that can be used to address a variety of issues related to computing.

Table 2.9. Resource management techniques for computing optimisation in fog computing

Learning technique	Problem	Application
Supervised (linear regression)	Edge device communication	Seismic imaging
Supervised (classification- SVM)	Continuous and real-time patient monitoring	Healthcare patient monitoring
Unsupervised (clustering-hidden Markov model)	Automatically generates musical score from a huge amount of music data in a IoT network	Music cognition
DL	Large amount of IoT sensor data adopted in industrial productions	Smart industry
Unsupervised (density estimation- CRBMs)	Centralised data processing	Traffic modelling
Unsupervised (clustering-PCA)	Centralised data processing	Smart dairy farming
Cognition-based communications	Higher Quality of Experience (QoE) and higher energy efficiency for users' applications demands	User-centric cognitive communications and cognitive Internet of Vehicles
Supervised (classification- SVM, decision tree, and Gaussian naïve Bayes)	Energy efficiency and latency requirements for time-critical IoT	Time-critical IoT applications
Supervised (classification- SVM)	Accuracy and adaptability of data analytics on the edge of a network	Health monitoring systems

2.6.2 Decision Making

An architecture termed SmartFog was presented by the authors in [59] as a means to model the complex functions of the human brain using fog computing and

machine learning, resulting in a flexible architecture with efficient resource management. Furthermore, the proposed architecture was able to respond to changes and make accurate decisions, while ensuring low latency. In [84], a directional mesh network (DMN) mechanism was presented for a smart telehealth use case, which facilitated decision making regarding data transmission to the remote cloud servers. Moreover, the Smart Cargo idea presented by the authors in [85] used Multi-armed bandit in order to provide real-time responses to decisions made regarding unexpected situations.

Table 2.10 summarises the research works in literature that investigated the use of machine learning techniques to advance decision making in fog computing. It is evident from the table that clustering approaches from USL are the popular choice for ML-aided decision making in fog. Furthermore, it is clear that handling computationally extensive tasks is a recurring challenge in fog computing, especially when considering the constrained computing capability of the fog nodes.

Table 2.10. Resource management techniques for decision making in fog computing

Learning technique	Problem	Application
Unsupervised- Spectral clustering	Unpredictable load patterns of distributed IoT applications	IoT network for applications
Unsupervised- PCA clustering	Limited radio spectrum resources	DMN
Multi-armed bandit	Real-time response to detected unexpected situations	Smart cargo

2.6.3 Resource Provisioning

A deep learning mechanism for edge resource provisioning was presented in [86], which is both parallel and distributed. Correspondingly, the work in [87] also considered ML-aided resource provisioning through the design of an algorithm that can accurately estimate the available resources. The work in [88] concentrated on using machine learning to aid dynamic mobility management.

Meanwhile, network latency estimations for animation rendering are investigated in [89] through a variety of ensemble methods, including random forest, gradient boosting trees, and SVM. The authors in [90] recommended four types of ML approaches that can be used along with the Message Queuing Telemetry Transport (MQTT) protocol as a solution for the communication challenge between resource constrained devices. Finally, the work in [91] illustrated the efficacy of machine learning models for efficient energy consumption of centralised data processing tasks.

Table 2.11summarises the research works in literature that investigated the use of machine learning techniques to enhance resource provisioning in fog computing. It is evident from this table that classification methods from SL are widely applied for ML-inspired resource provisioning in fog computing. Furthermore, machine learning techniques can be applied to a wide range of different challenges related to resource provisioning.

Table 2.11. Resource management techniques for resource provisioning in fog computing

Learning technique	**Problem**	**Application**
DL	Latency of analysing large amounts of data	Smart city
Supervised (classification- random forest, gradient boosting trees, SVM)	Predicting the completion time of each rendering job	Animation rendering
Supervised (classification- naïve Bayesian classifier)	Ultra-low latency mobile networking for AVS	Mobile network
Supervised (regression)	Network quality, accuracy, and operational overhead	Image processing service
Supervised, neural network, decision tree-linear regression	Energy of end devices, communication among resource-constrained devices	Industry 4.0 factories
	Energy consumption of centralised data processing	Environment measurement

2.7 Resource Management Techniques in 5G

One of the key issues for mMTC communications in 5G networks is radio resource management or scheduling. This section surveys the common techniques generally applied to resource management issues in 5G systems.

An interference-aware radio resource (IARR) allocation for uplink data transmission was proposed in [92] by presenting a sum-rate maximisation problem. The main objective of the proposed algorithm was to minimise transmission delays as a means to realise high reliability and advance the link quality in next generation wireless networks. In this work, an interference-aware heuristic solution was recommended to decrease the computational complexity of the sum-rate maximisation problem. Noteworthy improvements were identified in the link reliability and reduced latency with IARR algorithm when compared with the conventional round-robin scheduling (RRS).

The work in [93] introduced a resource management framework inspired by Non-Orthogonal Multiple Access (NOMA) and Successive Interference Cancellation (SIC), as a means to tackle synchronised spectrum sharing and data traffic management issues in heterogeneous 5G networks. This increased data rates by reducing energy consumption and increasing spectral efficiency.

Model-free reinforcement learning is very useful in wireless networks, where the complete model of the environment is unknown and the accuracy of information is an uncertainty. Specifically, this technique can be applied to stochastic optimisation problems. In sequential decision-making scenarios, RL can be used to find the optimum policy by interacting with the environment. Much consideration has also been given to Q-learning mechanisms lately. For instance, the work in [94] derived distributed algorithms inspired by Q-learning, in which small cells act at the agent that interacts with the environment to adapt sleeping patterns. The objective of the RL agent was to minimise the energy consumption while guaranteeing system performance. The work in [95] represented the network as a Multi- Agent Reinforcement Learning (MARL) system, where a Q-learning algorithm was executed in every small cell. However, this work was

extended in [96] as a result of no efficient management between the base stations and the Q-learning algorithm. As an extension to RL, the authors introduced a centralised neural network algorithm known as Heuristically-Accelerated Multi-agent Reinforcement Learning (HAMRL) [97].

Neural networks can be effective for radio resource allocation in 5G New Radio [98]. The work in [99] addresses the issue of resource management in a virtual RAN environment. In this work, an analytical model for managing virtual radio resources was proposed, which is responsible for the estimation of available radio resources and their allocation to different virtual mobile network operators. As a means to demonstrate how the challenges of learning Radio Resources Management (RRM) algorithms in a radio environment could be addressed, a learning framework was presented in [100] that consisted of Neural-Fitted Q-Iteration (NFQ), ensemble learning and transfer learning. A novel resource allocation algorithm termed threshold controlled access (TCA) was introduced in [101], which considered the power consumption of resource-constrained MTC devices in machine-to-machine communications. Two schedulers for IoT communications based on QoS requirements were proposed in [102] as a means to provide a trade-off between the traffic in machine-to-machine and human-centric communications by ensuring the network performance and enabling effective utilisation of network resources. Through extensive simulation, the results proved that the proposed approach is successful in obtaining maximum bandwidth utilisation, which is a key issue in the management of 5G radio resources.

The summary for resource management techniques in 5G is presented below:

- Machine learning techniques have demonstrated potential for creating autonomous and self-optimised wireless networks. Future wireless networks require the flexibility to adjust to fast evolving network environments, and this can only be realised through machine learning approaches. Therefore, machine learning in fog computing is an effective method for maximum resource utilisation as well as enhanced decision making of resources in the network.

2.8 Resource Allocation Techniques for 5G F-RAN

F-RANs have been presented as a favourable architecture for the provision of high energy efficiency and spectrum efficiency in future wireless networks. While the potential of F-RANs has been highlighted in numerous relevant works, the cache resource optimization is still a challenge due to the unpredictable nature of user file requests dynamics. The work in [23] proposed an algorithm based on deep reinforcement learning (DRL) as a means to improve this. In addition to cache resource optimisation, realising ultra-low latency in future wireless networks remains challenging due to constrained fronthaul capacity. The work in [21] attempted to achieve ultra-low latency by presenting a distributed content sharing and computing mechanism combined with the greedy algorithm. The proposed approach, which was successful at optimising the transmission rate, was proven to be a sub-optimal solution. In [103], an approach was devised based on DRL to minimise power consumption of the network in the long-term. The authors demonstrated that integrating transfer learning with DLR yields promising performance gains and requires much fewer interaction with the environment. The data offloading problem for delay-sensitive applications was formulated as a latency optimisation problem for different intensity of traffic and processing volume of tasks in [104]. The proposed Quadratically Constraint Quadratic Programming (QCQP) algorithm demonstrated the effectiveness of the joint task offloading and computational resource allocation scheme. A latency-driven cooperative algorithm for the F-RAN was presented in [105]. The dynamic programming inspired approach attempts to address the computing task assignment and communication resource allocation problem while guaranteeing minimum service latency. The authors in [106] attempted to resolve the latency optimisation problem for F-RANs through a DRL based cooperative proactive power allocation and cache placement mechanism inspired by DRL. The main premise of the proposed approach is to learn the user's demand in order to make an intelligent decision regarding caching appropriate content and dynamically adjusting power resources. The design of computation offloading in F-RANs to minimise the total cost with respect to the offloading latency and the energy consumption was investigated in [107]. In particular, a joint optimization problem

was formulated to optimise the offloading decision, the computation and the radio resources allocation. An iterative algorithm was designed, which showed promising performance gains, including computational complexity. Similarly, an iterative algorithm was adopted in [108] to optimise the offloading decision, the CPU-cycle frequency and the transmit power control. The proposed solution, which was based on the conventional convex optimisation methods, minimised the sum of energy consumption while satisfying the delay tolerance of each task and the constraints of maximum transmission delay tolerance, fronthaul and backhaul capacity limits. As part of the F-RAN resource management effort, the authors in [25] set out to design a resource allocation strategy based on differential game and bipartite graph multiple matching, and proposed a distributed uplink computation offloading strategy with Lyapunov theory and deviation update decision algorithm (DUDA). The proposed mechanism performed well in terms of system consumption and resource demand satisfaction rate. In [109], the resource allocation problem was formulated as a Markov Decision Process (MDP), for which an optimal decision policy was presented through reinforcement learning. The proposed resource allocation method learned from the IoT environment how to strike the right balance between two conflicting objectives of maximising the total served utility and minimising the idle time of the fog node. The transmission latency between fog nodes, node-to-UE, and fronthaul latency strongly depends on interference power from the undesired network element as well as end-users. At the same time, the computational latency increases with the queuing delay. In [110], a load balancing scheme was proposed to address the trade-off between transmission and computing latencies in F-RANs. The suggested method outperforms the greedy approach in terms of low latency and minimal task offloading to the cloud.

The applications of resource management techniques in F-RANs are listed in Table 2.12.

Table 2.12. Summary of resource management technique in F-RANs

Method	Problem	Objective
Deep reinforcement learning	cache resource optimisation	Maximise successful transmission probability of user requests
Deep reinforcement learning	resource management and mode selection	minimising long-term system power consumption
Greedy algorithm	distributed computing and content delivery	Minimise transmission delay
Convex optimisation	Computation offloading	minimising the sum of energy consumption
Quadratically Constraint Quadratic Programming	task offloading and computational resource allocation	Minimise latency
Dynamic programming	communication resource allocation and computing task assignment	minimum service latency
DRL	cache placement and power allocation	Latency optimisation
Iterative algorithm	computation and radio resources allocation	Minimising the total cost
Deviation update decision algorithm	uplink computation offloading	Network consumption and resource demand satisfaction rate optimisation
Markov Decision Process	load balancing	Maximising the total served utility Minimising the idle time

From the table, it is evident that latency optimisation is a significant consideration in the design of 5G F-RAN systems.

2.9 Conclusion

In this chapter, the relevant work related to resource management techniques in F-RANs was presented.

The issue of the allocation of Virtual Network Functions (VNFs) to Virtual Machines (VMs) while ensuring minimal network delays remains one of the key challenges for NFV, especially when considering the fast evolving network environment. While extensive research has gone into investigating the common issue of VNF placement, the problem of optimising virtual network performance in a dynamically changing resource availability has not been comprehensively considered yet.

Considerable advances have been made in the area of designing algorithms for the management of resources in the 5G F-RAN architecture, with machine leaning paving the way for dynamic and autonomous mechanisms. Despite the promise, most of these efforts make the assumption that the resources are fixed and/or the network functions are executed on black boxes. The problem with this approach is that it does not account for the dynamic formation of the 5G F-RAN. Furthermore, most approaches to the resource allocation problem only adopt computation offloading, which may result in adverse consequences to the performance due to additional offloading delays in scenarios where the fog nodes are very resource constrained. There is a limited number of studies in the domain of the self-management of networks with softwarised and virtualised resources. Based on these shortcomings, two techniques for resources management proposed in the subsequent chapters of this dissertation are proposed for F-RANs in 5G networks. The work in Chapter 3 defines the mathematical model for 5G F-RAN architectures and formulates the resource allocation problem. Chapter 4 and Chapter 5 of the dissertation propose techniques for resource management in 5G F-RANs. Firstly, an allocation method based on reactive auto-scaling is presented, then machine learning capabilities are leveraged to design a dynamic and autonomous resource management algorithm based on proactive auto-scaling.

Chapter 3 – Mathematical Model for Fog Radio Access Network Architecture in 5G

3.1 Introduction

The C-RAN model is the conventional architecture proposed for 5G networks. However, this model- which is intrinsically characterised by a heavy burden on fronthaul- is inadequate to cope with the increasing number of IoT applications requiring low latency for better user satisfaction. The F-RAN architecture, on the other hand, eases the heavy burden on the fronthaul and minimises transmission delays. This chapter examines the F-RAN architecture and provides justification for its selection. In particular, this chapter defines the overall system architecture, which includes the network model based on F-RANs in 5G cellular systems, and the virtualisation model. Then, the issue of resource allocation for 5G F-RAN systems is formulated as an optimisation problem. Finally, the performance of the optimisation solution is evaluated against the conventional C-RAN model through extensive simulations.

3.2 System Model

The system model considers a three-tiered hierarchical architecture, which is composed of UEs, 5G base stations, fog nodes and remote cloud servers. UEs connect to the fog nodes through wireless communication links using 5G base stations, while fog nodes access servers in the remote cloud data centre through fibre-optic communication as illustrated in Figure 3.1. This work focuses on underserved communities, which are often characterised by intermittent or no Internet connectivity. Therefore, most data processing is completed in the fog network, while the cloud is used for historical storage and batch analytics.

The model is based on dual radio connectivity, incorporating the architecture of 4G LTE and 5G New Radio (NR). The LTE eNB is deployed as the master eNB, while the NR eNB/gNB is the secondary eNB balancing the load and enhancing user throughput.

The network functions in the system are softwarised and run on isolated virtual machines (VMs) through the Network Function Virtualisation (NFV) technique. These VMs, also referred to as fog nodes, connect to each other using Software Defined Networking (SDN), while also monitoring and managing network traffic among them. In this work, VM, virtual node and fog node are used interchangeably.

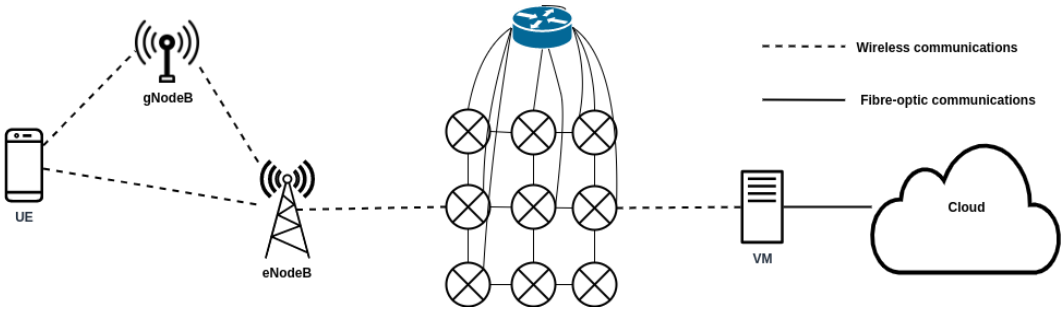


Figure 3.1. F-RAN system model

The specification of VN resource requirements is represented by a weighted undirected graph $G = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ represent the sets of fog nodes and links respectively. Each virtual link $l_{ij} \in L$ or fog node $i \in N$ is characterised by requirements such as maximum delay, CPU, memory, bandwidth etc. In the network, the set of UEs is denoted by $\mathcal{K} = \{1, 2, \dots, |\mathcal{K}|\}$. A set of UEs of a specific node n is denoted as K_n, while k_n denotes a single UE of the node. Various kinds of UEs send their task data to a certain fog node n, and the service arrival rate to the node is γ_{k_n} packets per second.

3.2.1 Delay Model

The experienced latency of the k^{th} user is defined as

$$T_k = T_k^w + T_k^{tran} + T_k^{comp} \tag{3.1}$$

Where T_k^w is the wireless latency incurred by sending the data of the computation task from the k^{th} user to a base station (BS), T_k^{tran} is the network latency caused by transferring the k^{th} user's data to a fog node, and T_k^{comp} is the computation latency for performing the task on the assigned fog node.

The wireless latency is determined by the data size of a user's computation task and achievable transmission rate. Denote M_k as the data packet size and R_k as the achievable transmission rates in bits/s, then the wireless latency experienced by the k^{th} user is given by

$$T_k^w = \frac{M_k}{R_k} \tag{3.2}$$

The maximum transmission rate is characterized by the signal-to-noise ratio (SNR) through Shannon channel capacity. Shannon's fundamental limit on the capacity of a communications channel gives an upper bound for the achievable transmission rates, as a function of available bandwidth (B) in Hz and SNR in dB:

$$R_k = B \log_2(1 + SNR) \tag{3.3}$$

Denote $x_{k_n} \in \{0, 1\}$ as the task assignment indicator whether the k^{th} user is served by the n^{th} fog node. Denote $\mathcal{X} = \{x_{k_n} \mid k \in \mathcal{K}, n \in \mathcal{N}\}$ as the set of users' task assignments. x_{k_n} is set as a binary variable to indicate that a user can only be served by one fog server at a time. Let τ_{k_n} denote the network latency

between the k^{th} user's associated base station and the n^{th} fog node. Then, the network latency of the k^{th} user can be modelled as

$$T_k^{tran} = \sum_{n \in \mathcal{N}} x_{k_n} \tau_{k_n} \tag{3.4}$$

The computation latency is closely related to the computational complexity of a user's task and available computation resources on servers [111]. Let a_k be the minimum processing density requirement - GFLOPS (Giga Floating-point Operations Per Second)- for performing the k^{th} user's computation task and $d_k = M_k a_k$ be the required CPU cycles . f_{k_n} is defined as the allocated computation resources by the n^{th}fog node to the k^{th} user. Denote $\mathcal{F} = \{f_{k_n} \mid k \in \mathcal{K}, n \in \mathcal{N}\}$ as the set of computation resource allocations. Therefore, the computation latency experienced by the k^{th} user can be modelled as

$$T_k^{comp} = \sum_{n \in \mathcal{N}} x_{k_n} \frac{d_k}{f_{k_n}} \tag{3.5}$$

Based on the above analytical model, the overall service latency experienced by the k^{th} user is

$$T_k = \frac{M_k}{R_k} + \sum_{n \in \mathcal{N}} x_{k_n} \left(\tau_{k_n} + \frac{d_k}{f_{k_n}} \right) \tag{3.6}$$

3.2.2 Throughput and Utilisation Model

Consider the fog network as an open network, in which arrivals from, and departures to, the outside world are permitted [112]. In this case, the outside world may refer to the UEs in the device tier or the servers in the remote cloud

data centre. It is assumed that packets may arrive at a VM from an outside source (such as a UE) according to a Poisson process that is specific for that VM.

Exponential service rates are assumed for all fog nodes in the system. In this case, the aggregate arrival rate is equal to the sum of all the individual arrival rates. If the individual arrival rate of each node n is defined as γ_n, then the aggregate rate is given as:

$$\lambda_n = \sum_{k=0}^{\mathcal{K}} \gamma_{k_n} \tag{3.7}$$

Each node's utilisation can be defined as:

$$U_n = \frac{C_n}{\mu_n} \tag{3.8}$$

Where μ_n denotes the average service rate of a fog node and C_n is the node's throughput, which is defined by:

$$C_n = \sum_{j=0}^{N} C_j P_{nj}$$

$$C_n = \sum_{j=1}^{N} C_j P_{nj} + \gamma_n \tag{3.9}$$

Where P_{nj} is the probability that the n^{th} fog node will send the task to j.

In a system with N queues and associated fog nodes, given that packets leave the fog node n with the probabilities defined as $P_{n0} = 1 - P_{nj}$. This definition states

that the possibility of a task leaving the system is equal to the complement of the probability that a task will remain in the system. Since the throughput arriving from the IoT network is known, C_0 can be set to λ and solve for the remaining C terms.

All the notations used in this chapter are described in Table 3.1.

Table 3.1: F-RAN architecture notation definitions

Symbol	Description
$\mathcal{K}$/K	Set/number of UEs
$\mathcal{N}$/N	Set/number of fog nodes
$\mathcal{X}$	Set of task assignments
$\mathcal{F}$	Set of computation resource allocations
M_{k}	Data packet size
λ	Packet arrival rate
μ	Service rate
C_n	VM throughput
B	Channel bandwidth
T_k	Overall latency
T_k^w	UE to fog node transmission delay
T_k^{comp}	Computation latency
x_{k_n}	Task assignment indicator
τ_{k_n}	BS to fog node network latency
T_k^{tran}	BS to fog node transmission delay
a_k	Processing density
d_{k}	Processing capacity requirement
f_{k_n}	Allocated computation resources
R_{k}	Achievable transmission rate
U_n	Fog node utilisation
μ_n	Average service rate
P_{nj}	Task probability
f_n^{max}	n^{th} node total computation resources
T_k^{max}	Maximum tolerable latency

3.3 Problem Formulation and Optimisation

Given the abovementioned system model, the problem of resource allocation for a fog-enabled 5G communication system is formulated. Since the goal of the system is to minimise the total end-to-end latency experienced by users through

computation resource allocation F while enforcing the maximum tolerable latency requirement constraint, the optimisation problem is defined as:

$$\min_{\{\mathcal{F}\}} \sum_{k \in \mathcal{K}} T_k \tag{3.10}$$

Subject to:

$$\sum_{k \in \mathcal{K}} x_{k_n} f_{k_n} \leq f_n^{max}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{3.11}$$

$$T_k \leq T_k^{max}, \forall k \in \mathcal{K} \tag{3.12}$$

$$f_{k_n} \geq 0, \forall n \in \mathcal{N}, k \in \mathcal{K} \tag{3.13}$$

Where f_n^{max} is the total computation resources in the n^{th} node and T_k^{max} denotes the maximum tolerable latency of the k^{th} user. The constraint in equation (3.11) ensures that the computation resources on individual nodes are not allocated in excess, equation (3.12) guarantees that the service latency experienced by individual users do not exceed their maximum tolerable latency and equation (3.13) is the non-negative constraint on computation resource allocation. In this work, only CPU is considered as a computation resource.

To solve the problem, the binary variables x_{k_n} are first relaxed to continuous variables $\widetilde{x_{k_n}}$. Denote $\widetilde{\mathcal{X}} = \{\widetilde{x_{k_n}} \mid k \in \mathcal{K}, n \in \mathcal{N}\}$. The relaxed problem becomes

$$\min_{\{\mathcal{F}\}} \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{N}} \widetilde{x_{k_n}} \left(\tau_{k_n} + \frac{d_k}{f_{k_n}} \right)$$

(3.14)

When the task assignments $\widetilde{\mathcal{X}}$ are given, the problem can be reduced to

$$\min_{\{\mathcal{F}\}} \sum_{k \in \mathcal{K}} \sum_{n \in N} \widetilde{x_{k_n}} \frac{d_k}{f_{k_n}}$$

(3.15)

The problem is strictly convex with respect to the computation resource allocation $\mathcal{F}$.

Proof: For any feasible $f_{m_n}, f_{i_j}, \forall m, i \in \mathcal{K}, \forall n, j \in \mathcal{N}$,

$$\frac{\partial^2 T}{\partial f_{i_j} \partial f_{m_n}} = \begin{cases} \dfrac{2\widetilde{x_{i_j}} c_i}{\left(f_{i_j}\right)^3}, & i = j \text{ and } m = n \\ 0, & i \neq j \text{ or } m \neq n \end{cases}$$

(3.16)

The Hessian matrix $T = \left[\frac{\partial^2 T}{\partial f_{i_j} \partial f_{m_n}}\right]_{KN \times KN}$ is symmetric and positive definite. The constraints (3.11), (3.12) and (3.13) are convex with respect to $\mathcal{F}$. Hence, the problem is strictly convex with respect to $\mathcal{F}$ [113].

Finally, the continuous task assignment $\widetilde{x_{k_n}}$ can be converted to the integer task assignment x_{k_n} according to

$$x_{k_n} = \begin{cases} 1, & n = arg_{i \in \mathcal{N}} \max \widetilde{x_{k_i}} \\ 0, & otherwise \end{cases}$$

(3.17)

The procedure of the computation resource allocation has been summarized in Figure 3.2.

Input: T_k^{max}, f_n^{max}, $\mathcal{X}_0$
Output: $\mathcal{F}$, $\mathcal{X}$
$\tilde{\mathcal{X}} \leftarrow \mathcal{X}_0$
while *True* **do**
  $\mathcal{F} \leftarrow$ solve problem with fixed $\tilde{\mathcal{X}}$
Determine $\mathcal{X}$ according to Eq.(3.17)
return $\mathcal{F}$, $\mathcal{X}$

Figure 3.2. Computation resource allocation pseudocode

3.4 Evaluation and Results

This section presents the performance evaluation of the computation resource allocation, which is implemented in a 5G F-RAN architectural model, against the conventional C-RAN architecture. As part of modelling a 5G network, a smart farm use case is considered. In smart farming, various sensing technologies are deployed across the field for the provision of data to be processed and implemented as need be in order to enable farmers to monitor and optimize crop yield while adapting to changing environmental factors [1]. The idea behind smart farming is to leverage real-time connectivity to enable machine-to-machine communication between farm equipment and other machines on the field. Thus, making 5G and fog computing technologies suitable enablers for the use case.

The simulation environment was created using 5G K network simulator, 5G K-SimNet [114]. 5G K-SimNet is an open source ns3-based network simulator for evaluating end-to-end performance of 5G systems. Its key elements for 5G include support for 5G New Radio based on mmWave, 5G core, multi-connectivity, SDN, and NFV modules.

A mobile edge computing network was modelled with 10 fog nodes and 20 BSs. The users and BSs were randomly distributed in a 1x1km area. The total system bandwidth B was 20 MHz. The computing capacity of the fog node was set to

5000 GFLOPS**.** The network latency τ was uniformly distributed between 10 and 50 milliseconds.

Three types of user applications were considered in the simulation, as shown in Table 3.2. Type I applications have stringent latency requirements with medium data sizes and compute intensity. Type II applications have high computation requirements with small data sizes and medium latency-sensitivity. Type III applications have large data sizes but with low compute-intensity and high latency tolerance. In the default simulation setting, the percentages of type I, type II, and type III applications are 40%, 30%, and 30%, respectively.

Table 3.2. User application parameters

Parameter	Type I	Type II	Type III
Data size (Mbits)	1	0.1	10
Computing density (GFLOPS)	100	1000	10
Tolerable latency (s)	0.2	1	2

To quantify the user's satisfaction about edge computing services, a dissatisfaction ratio ρ is defined as

$$\rho = \frac{|\mathcal{K}_u|}{|\mathcal{K}|} \tag{3.18}$$

Where $\mathcal{K}_u = \{k \mid T_k \geq T_k^{max}, \forall k \in \mathcal{K}\}$ is the set of unsatisfied users and $|\mathcal{K}|$ is the number of unsatisfied users.

The average packet latency of users has been measured as given by the equation:

$$\bar{T}_k = \frac{\sum_{i=1}^{p_k} \sum_{j}^{q_{ik}} \left(T_{departure}^{kij} - T_{arrival}^{kij}\right)}{\sum_{i=1}^{p_k} q_{ik}} \tag{3.19}$$

Where p_k denotes the number of packet transmissions during the simulation period, and q_{ik} is the number of packets in the i^{th} transmission.

In the simulations, the computation resource management optimisation is implemented in an F-RAN architecture and compared to the traditional C-RAN architecture.

3.4.1 Impact of the Number of Users

Figure 3.3 shows the impact of the number of users on the average service latency. In the F-RAN, the demand for computation resources increases when there are more users in the system. Since the computation resources are limited, the increment of the number of users leads to the increase in the average service latency. However, the F-RAN model achieves the lowest latency due to the optimisation of computation resources. When compared to the C-RAN model, the F-RAN model reduces the average service latency by approximately 51% when $K = 1000$.

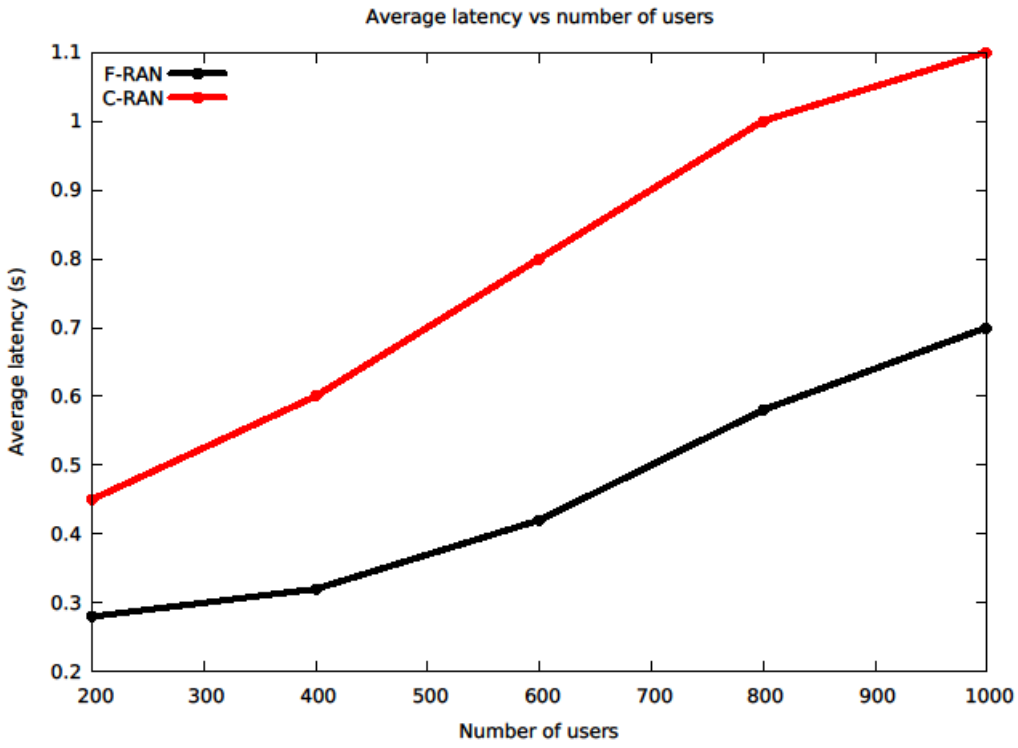


Figure 3.3. Impact of the number of users on average latency

Figure 3.4 shows the impact of the number of users on the dissatisfaction ratio. When compared to the C-RAN model, the F-RAN model reduces the dissatisfaction ratio by 40% when $K = 1000$. Moreover, when $K = 600$, the dissatisfaction ratio of the F-RAN reaches zero while that of the C-RAN is 35% higher.

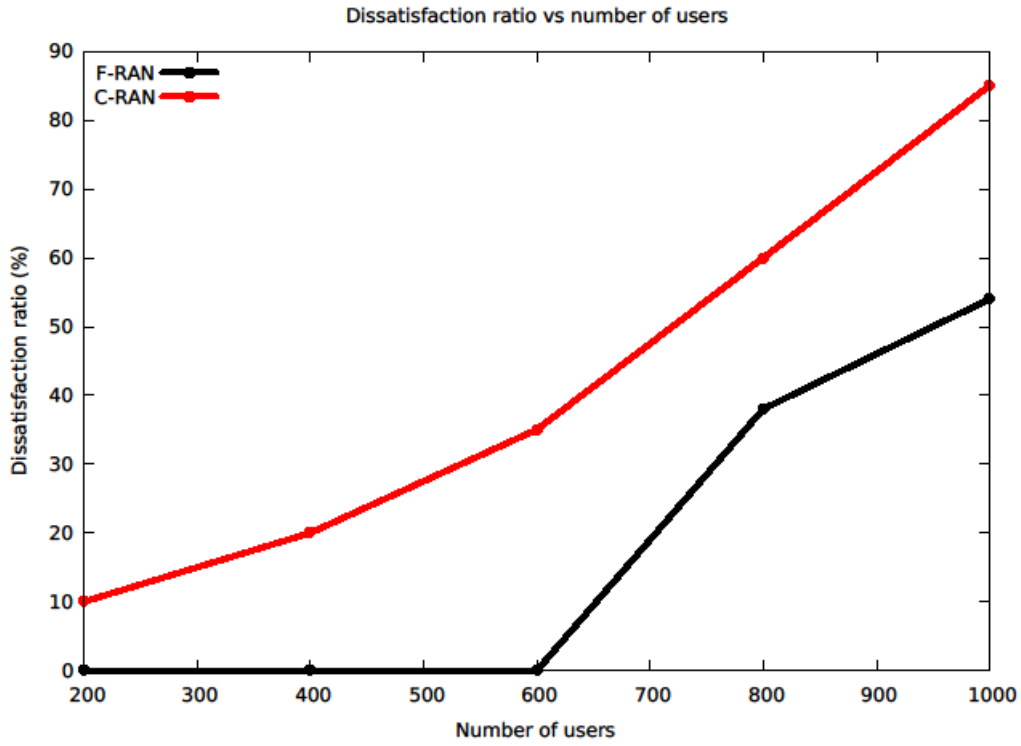


Figure 3.4. Impact of the number of users on dissatisfaction ratio

The impact of the number of users on average user throughput was also studied. Figure 3.5 illustrates the outcome of the average user throughput for a varied number of users. As shown in the graph, the average user throughput decreases with an increase in traffic demand, which is measured by the number of users. As noted, both models measured the highest average user throughput with the minimum number of users, i.e. when $K = 200$.

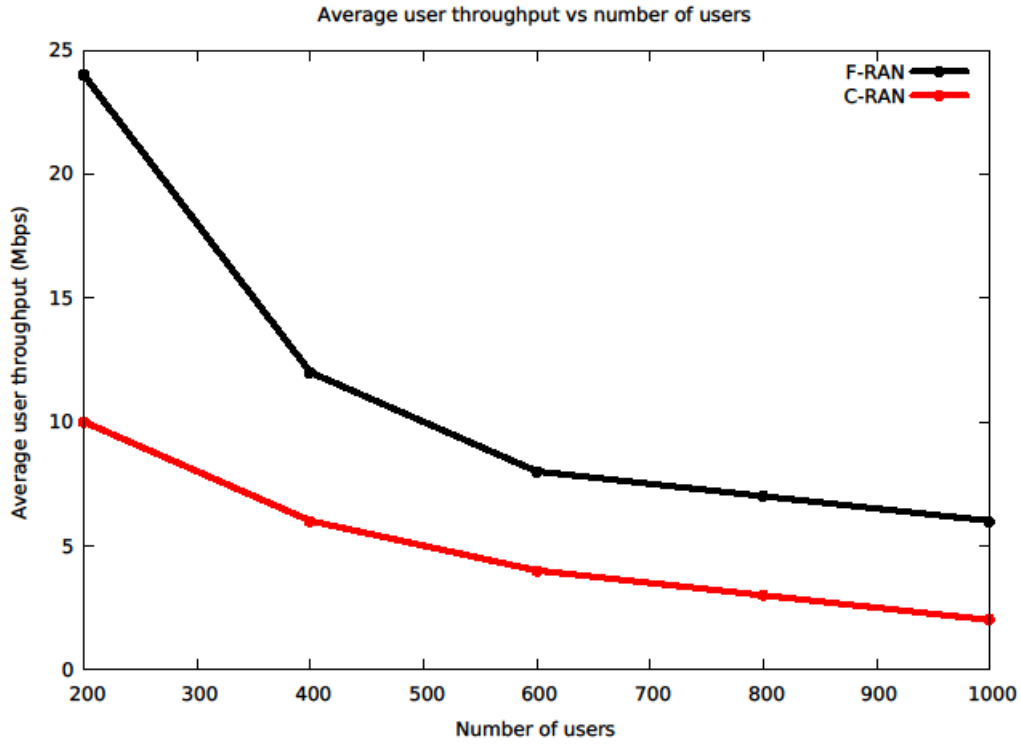


Figure 3.5. Impact of the number of users on average user throughput

Figure 3.6 shows the comparison of average resource utilisation with varying number of users. The resource utilisation was measured in terms of the average number of computing resources used entirely by the assigned tasks. In the C-RAN model, computation resources are distributed equally among all users irrespective of the application requirements. The F-RAN model, in contrast, optimises computation resource allocations based on the maximum tolerable latency and minimum processing density requirements of applications to as to ensure efficient utilisation. F-RAN outperforms C-RAN in terms of average resource utilisation. As illustrated in the figure, F-RAN achieves an average of 89% when $K = 1000$, while the C-RAN is at 65%. Moreover, when $K = 200$, the C-RAN achieves an average resource utilisation of 31%. On the other hand, the F-RAN always manages to achieve an average over 50%.

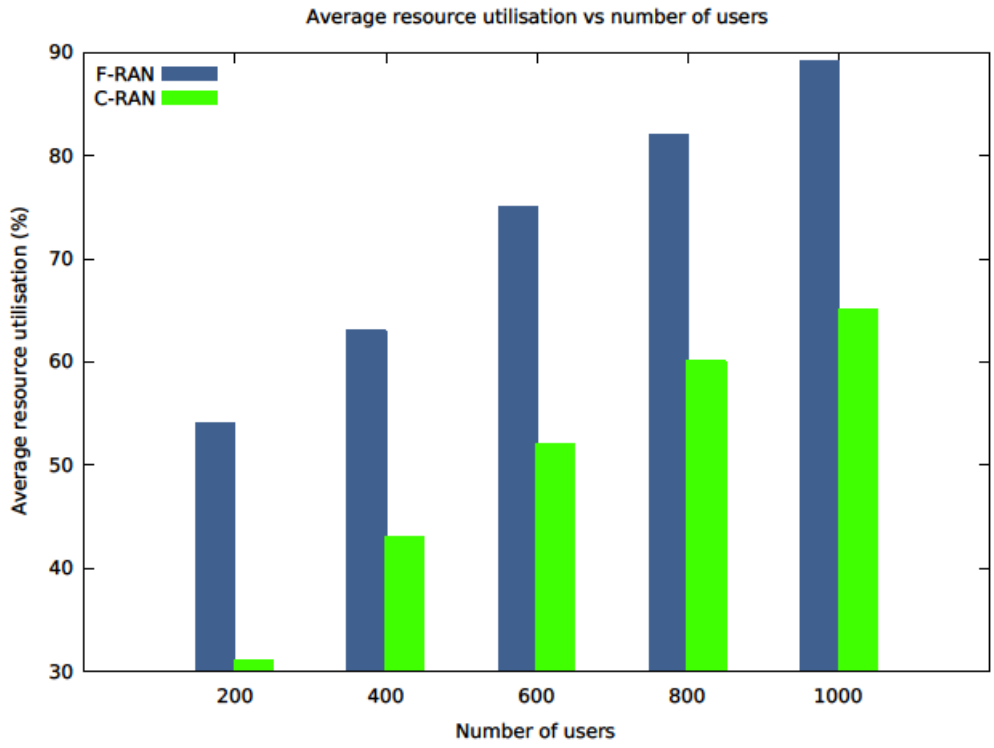


Figure 3.6. Impact of the number of users on average resource utilization

3.4.2 Impact of User Applications

The user applications listed in Table 3.2 have different properties that impact the service latency experienced by users. The impact of different applications on the average service latency, dissatisfaction ratio, user throughput, and resource utilisation was investigated. In the simulations, three user application settings were considered. In the first setting, the user applications were composed of 80% type I, 10% type II, and 10% type II applications. In the second setting, the user applications were composed of 10% type I, 80% type II, and 10% type III applications. In the third setting, the user applications were composed of 10% type I, 10% type II, and 80% type III applications.

Figure 3.7 shows that the F-RAN model outperforms the C-RAN model on average service latency under all three user application settings. Under setting I, the average service latency with the optimisation approach is 31% lower that its C-RAN counterpart. The observed performance gain can be attributed to the task assignments. The major user applications under setting I have strict latency

requirements. Without the task assignments, the C-RAN approach executes all computing tasks in cloud servers, thus increasing the average service latency. In comparison, the F-RAN distributes the workload among edge services. The F-RAN model reduces the average service latency by approximately 10% for applications requiring low latencies and 10% for applications with a high latency tolerance. This latency reduction is mainly attributed to the resource optimisation.

The performance of both models is comparable under setting II, whereby a majority of the applications are compute intensive (type II). In the F-RAN, a large number of computationally intensive workloads may lead to an increase in the number of tasks being offloaded to the cloud, thus increasing the average service latency.

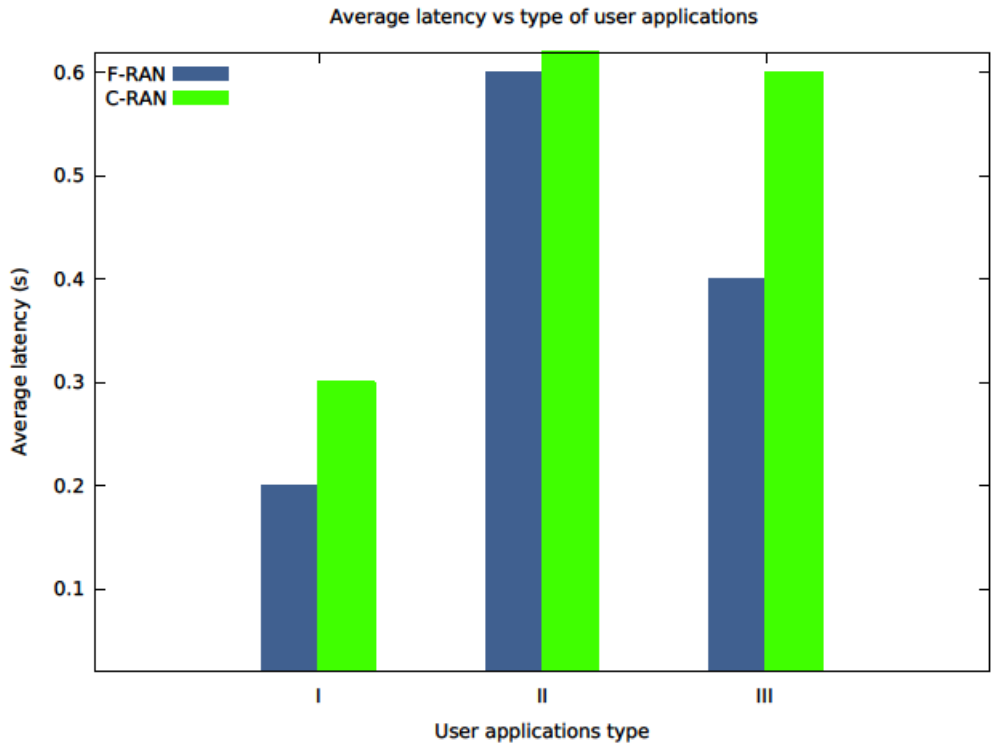


Figure 3.7. Impact of user applications on average latency

Figure 3.8 shows the impact of user applications on the dissatisfaction ratio. The F-RAN achieves zero dissatisfaction ratio under all the settings. In contrast, the C-RAN achieves a ratio as high as 35% under setting II.

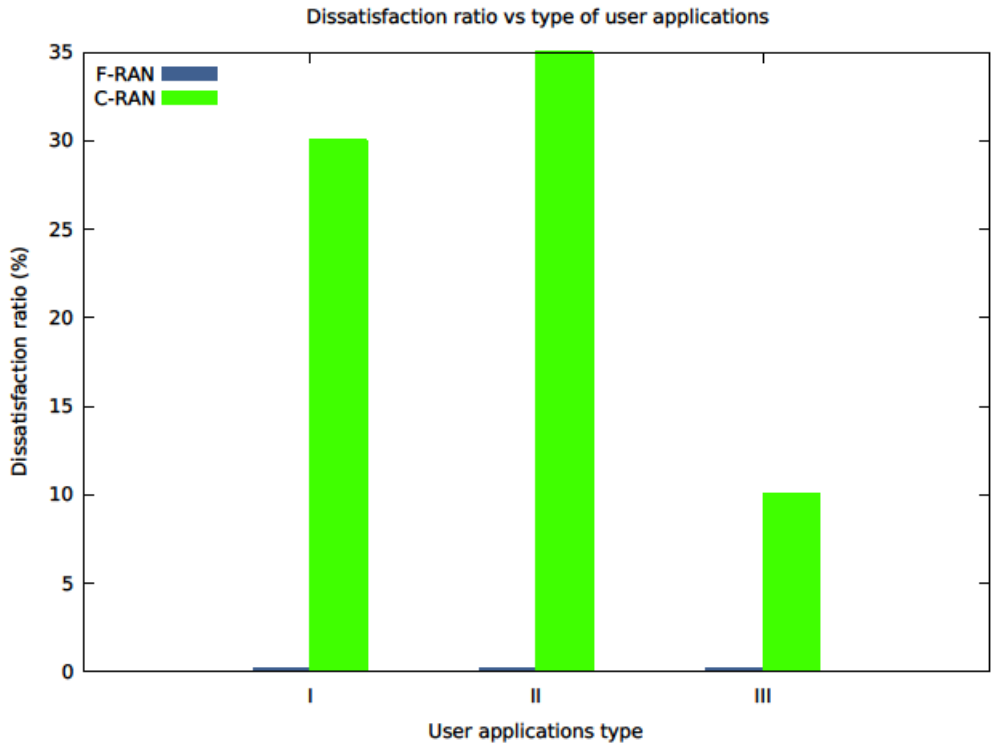


Figure 3.8. Impact of user applications on dissatisfaction ratio

3.4.3 Impact of Computation Capacity

Figure 3.9 and Figure 3.10 show the impact of the computation capacity on the average service latency and dissatisfaction ratio, respectively. In the F-RAN model, the computation capacity of the edge server was varied, while the C-RAN refers to the capacity of the cloud server. It is shown that the average service latency and dissatisfaction ratio are reduced with the increment of the computation capacity. This is because higher computation capacity helps to reduce the computation latency. The F-RAN reduces 52% of the average service latency and achieves zero dissatisfaction ratio when the computation capacity is 5000 GFLOPS. When the computation capacity is 4000 GFLOPS or more, only the F-RAN is able to maintain a zero unsatisfactory ratio. In other words, the C-RAN fails to satisfy all users' latency requirements. Therefore, the computation resource management is necessary to ensure the provision of low latency in mobile edge computing.

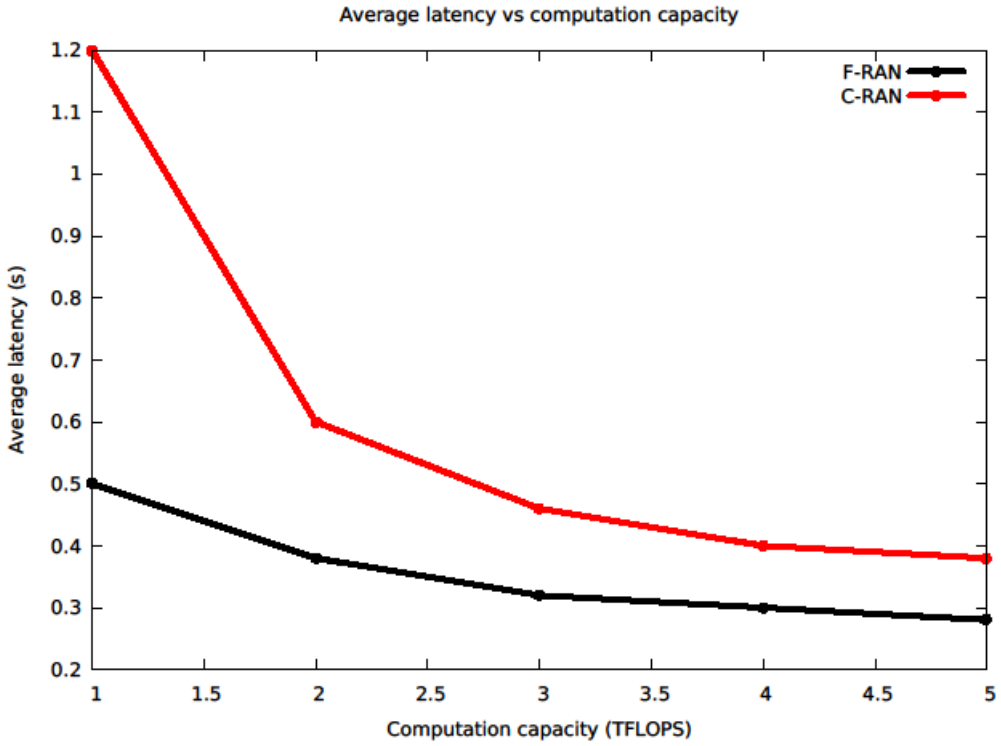


Figure 3.9. Impact of computation capacity on average latency

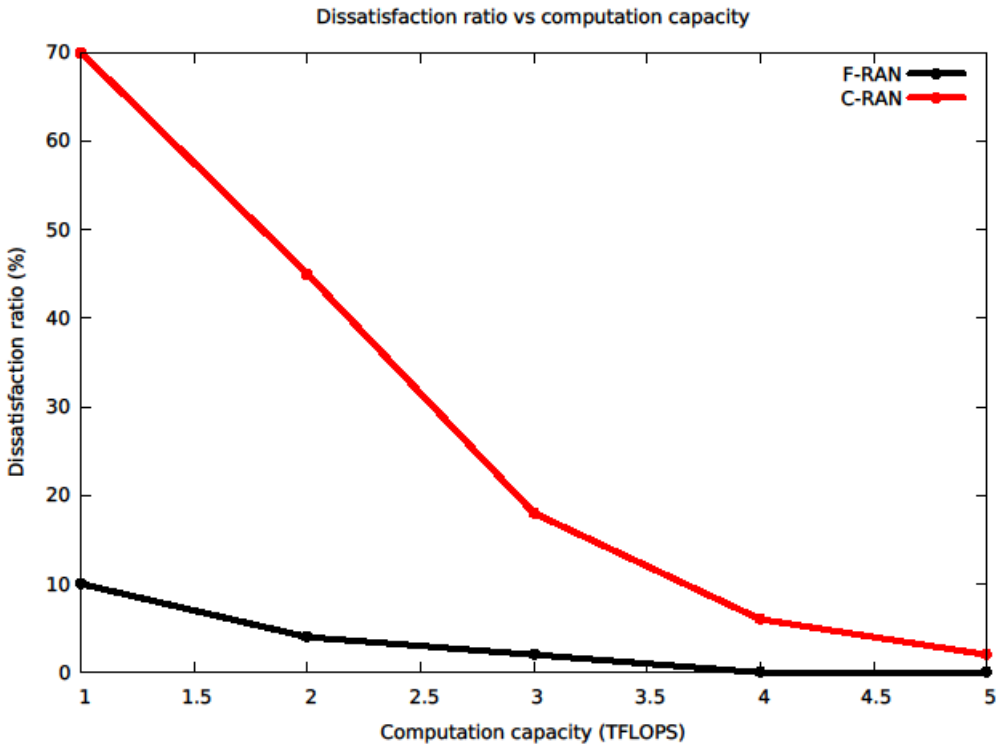


Figure 3.10. Impact of computation capacity on dissatisfaction ratio

3.5 Conclusion

In this chapter, the computation resource management problem for latency-constrained applications in mobile edge networks has been studied. An optimization problem was formulated, which aims to minimize the summation of users' average service latency under practical constraints. A solution to the optimization problem was described, which enables low-latency mobile edge computing. Extensive simulations were conducted to evaluate the performance of the proposed solution. The simulation results have demonstrated that the solution can effectively reduce the average service latency while meeting all users' latency requirements. Furthermore, the results measured in this chapter can be interpreted as a demonstration of the scalability of fog-based architectures. In the case of C-RAN deployment models, the uncontrolled growth of networks results in network congestion and performance degradation. The next chapter presents a virtualization-inspired technique for resource management in the F-RAN architecture as a means to minimize latency and consequently address the computation resource allocation problem formulated in this chapter.

Chapter 4 – Reactive Auto-scaling Resource Allocation Technique

4.1 Introduction

Auto-scaling mechanisms, which enable applications running on virtualised environments to maintain efficient resource utilisation while ensuring low operational cost, have become a typical paradigm in cloud computing environments. The benefits of auto-scaling techniques in centralised cloud-based models have been demonstrated in some research works, including improved response time [115], [116], increased resource utilisation [117] and reduced bandwidth [118]. However, designing auto-scaling techniques for applications deployed based on edge computing frameworks and distributed environments is challenging.

This chapter focuses on reactive auto-scaling as a proposed method for resource allocation in F-RAN architectures. The reactive auto-scaling mechanism described in this dissertation supports the dynamic provision of resources in response to changes in current system workload by deciding (i) whether to offload a workload to fog nodes or the cloud servers, and (ii) how to allocate or deallocate resources to a workload. In this chapter, the components of the proposed resource allocation framework and the proposed reactive auto-scaling algorithm are presented. Section 4.2 describes the proposed resource allocation framework. In section 4.3, the proposed dynamic auto-scaling algorithm is presented. A description of other resource management techniques considered for comparison is provided in section 4.4, followed by a definition of the evaluation setup in section 4.5. Finally, the results are presented in section 4.6 and the chapter is concluded with a brief discussion on the main findings.

4.2 Proposed Resource Management Architecture

As an extension of the system model described in section 3.2, the resource management architecture can be developed. As illustrated in Figure 4.1, the proposed framework uses the following components on the fog node:

- **Request Manager (RM):** This component handles incoming requests and forwards them to the decision maker. The primary responsibility of the RM is to receive the requests from users and forward those requests to the scheduler which decides whether to accept or reject the request. The decision of whether the request will be accepted or rejected is communicated back to the user.
- **Decision maker (DM):** The DM is responsible for decision making related to resource allocation and auto-scaling. It is comprised of the following sub-components:
    - **Resource allocator:** This is the component responsible for keeping track of the available CPU cores.
    - **Auto-scaler:** Based on the metrics obtained from the resource allocator, the auto-scaler dynamically allocates or de-allocates resources by implementing the auto-scaling algorithm.
- **Scheduler:** This component is responsible for allocating requests to be executed on a resource that is selected from the resource pool. The scheduler uses an Earliest Deadline First (EDF) strategy for scheduling, however other algorithms can also be considered as alternatives. Using EDF, the requests with the least time or deadline are given higher priority in the queue. If the newly arrived request can be allocated and scheduled on an existing resource such that its deadline can be met, the request is accepted. If the requested cannot be scheduled, then the auto-scaler is called on to determine if a new resource should be acquired.

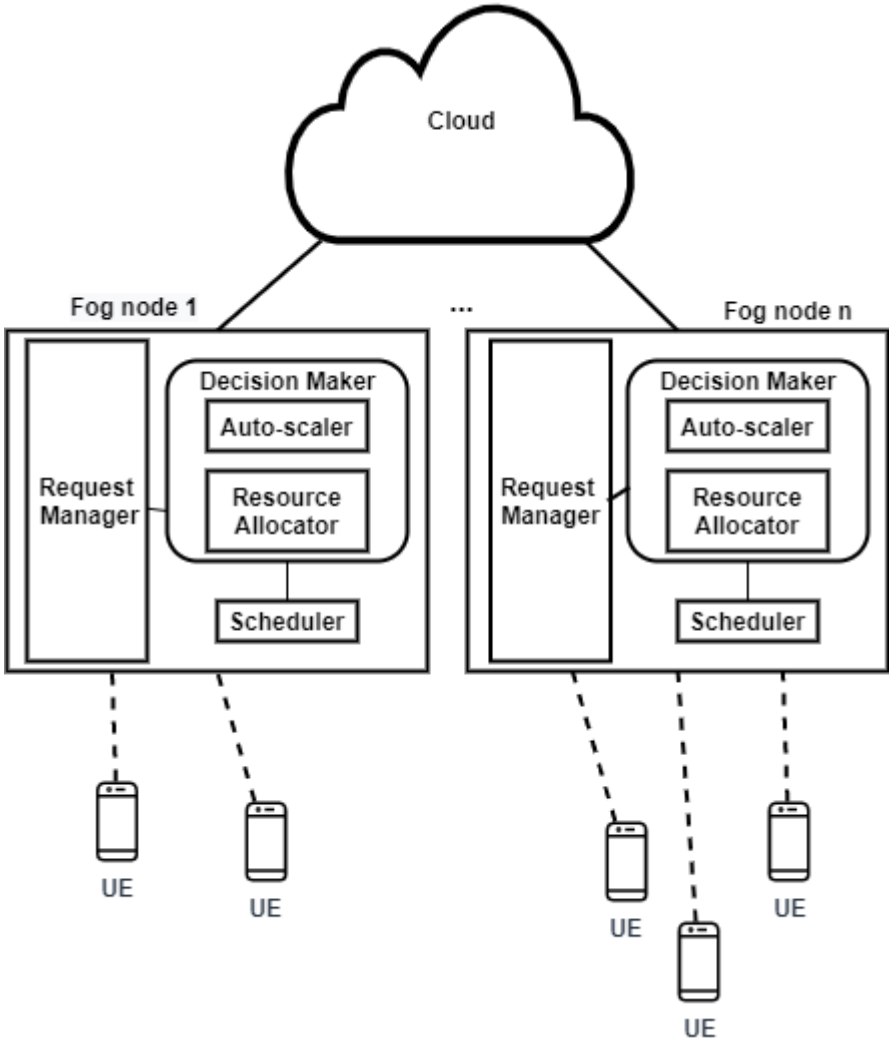


Figure 4.1. Architecture of fog nodes in the proposed reactive framework

4.3 Proposed Reactive Auto-scaling Algorithm

This section presents the proposed dynamic auto-scaling algorithm implemented by the resource management framework. The mathematical notations used in the proposed resource management mechanism are listed in Table 4.1.

The proposed auto-scaling mechanism is initiated by the request manager checking whether the node can support edge services. If a positive response is received, a request is sent to the user for the application details, including the size of sensor data, priority level, the minimum processing density and the maximum tolerable latency. Alternatively, the request manager can terminate its fog server.

Contrary to the servers in the cloud computing architecture, fog computing is expected to use edge servers for shorter time intervals as dictated by user demand and constrained edge node resources. Therefore, the request manager will be required to assume the role of terminating edge servers, in addition to performing auto-scaling operations. There are three conditions under which edge services may be terminated: (i) there is a lack of availability of resources, (ii) the edge server providing the service has been idle for too long and (iii) the service fails to enhance the application's QoS.

When a request is received, its priority and available resources are queried against the list of edge servers and corresponding priority levels kept by the request manager, with higher priorities being processed first. If a new request has a lower priority than the edge servers presently processing, the request is rejected. Similarly, new requests are rejected if there are insufficient resources. Note that an application's priority is constant throughout processing. Otherwise, if it is absolutely necessary, altering the priority will require the application to be relaunched on the server with the updated priority.

It is essential to perform scaling operations to resources on edge servers because firstly, fog nodes are characterised by constrained hardware resources, and secondly, the edge server processing on the fog node needs more or less resources to the meet the QoS objectives of users. Figure 4.2 outlines the proposed auto-scaling method, which is executed periodically.

The proposed auto-scaling method is initiated when the resource allocator performs a check to determine whether there are sufficient available resources. If the specified minimum resource requirement surpasses the available resources, then all the edge servers on the priority queue are offloaded to the cloud. Alternatively, the auto-scaler determines (i) whether the edge server is connected to any users or end devices, and (ii) whether the edge service can process the task within the predefined network latency objective of the application. The two checks are required for the auto-scaler to make appropriate decisions regarding the termination of edge servers. In this way, edge servers that are idle for extended periods of time or that do not improve QoS sufficiently may be offloaded to other

fog nodes or the cloud, where they may be more useful. The next step in the process is the check to identify whether or not fog nodes should be allocated more resources. If the computed network latency of the application exceed the desired latency requirement, additional resources will be allocated to the fog node. Alternatively, resources may be deallocated to a fog node if the computed latency is well within the minimum requirement.

```
   Input: F, f_i, R_x, R_a, U_i
 1 for f_i ← 1 to F do
 2     Measure T_i^n
 3     Measure T_i^c
 4     T_i^a ← T_i^n + T_i^c
 5     if R_x >= R_a then
 6         if T_i^n < T_i^max then
 7             if T_i^a > T_i^max then
 8                 scale(up, R_a, R_x)
 9             else
10                 scale(down, R_a, R_x)
11         else
12             Migrate and offload to cloud
13     else
14         Migrate and offload to cloud
```

Figure 4.2. Reactive auto-scaling algorithm

When the decision to perform a scaling operation has been made, the auto-scaler either allocates more resources if the decision is to *scaleUp* or deallocates a unit of resources if the decision is to *scaleDown*. In a *scaleUp* operation, a check is first performed by the fog monitor to establish whether there are sufficient free resources R_x for the fog node to facilitate scaling up. If the test passes, an additional unit of resources R_a is allocated to the fog node. Alternatively, the bottommost fog node on the priority queue in the network *F* will be terminated and the resources freed up for other high priority nodes. This process is executed iteratively for all low priority fog nodes until there are sufficient available resources to support scaling up or the list of low priority nodes is exhausted. For a *scaleDown* operation, a single unit of resources is deallocated from the server. The conclusion of the scaling mechanism is marked by the edge server being

updated with the modified resource allocations. The procedure of the scaling mechanism has been summarised in Figure 4.3.

```
   Input: scalingDecision, R_a, R_x
 1 if scalingDecision == up then
 2     Measure R_u
 3     if R_x >= R_a then
 4         R_u ← R_u + R_a
 5     else
 6         R_r ← 0
 7         while R_x < R_a do
 8             Measure R_n
 9             R_x ← R_x + R_n
10             R_r ← R_r + R_n
11             R_u ← R_u + R_r
12 if scalingDecision == down then
13     Measure R_u
14     R_u ← R_u + R_a
```

Figure 4.3. Scaling procedure

Auto-scaling mechanisms are generally unpredictable in scenarios where the resources of fog nodes are depleted. The proposed mechanism, however, alleviates uncertainties because the amount of resources that need to be deallocated from a fog node is recognized in advance. Another source of instability for auto-scaling approaches is the scaling down of fog nodes with lower priority, which can be addressed by circumventing progressive scaling down of low priority nodes. Hence, the proposed auto-scaling mechanism terminates low priority nodes one by one until enough additional resources are available for nodes with the highest priority.

Table 4.1. Notations used in the reactive auto-scaling algorithm

Parameter	Description
R_x	Free CPU on the fog node available for F
R_a	One unit of CPU
F	A set of n edge servers hosted on n containers in a fog node, ordered by priority. $f_i \in F, i = 1, \dots, n$
R_u	CPU used by f_u on the fog node; R_n is the CPU used by f_n on the fog node
U_i	A set of users to connect to f_i
T_k^{max}	Maximum tolerable latency
T_i^n	Measured average round-trip network latency of f_i
T_i^c	Measured average computing latency of f_i
T_i^a	Computed round-trip application latency of f_i ; $T_i^a = T_i^n + T_i^c$
R_r	CPU released by terminating servers on the fog node
scalingDecision	Flag for 'scaleup' or 'scaledown' containers on a fog node

4.4 Description of Other Auto-Scaling Resource Management Frameworks

The proposed resource management architecture described in section 4.2 enables each fog node to be responsible for its own decision making related to resource allocation. The proposed framework based on distributed control, which incorporates a reactive auto-scaling algorithm to facilitate efficient resource allocation, will be referred to as System I and is compared with other resource management architectures for performance evaluation.

The authors of [119], [120] described a dynamic resource allocation framework for an NFV-enabled mobile fog cloud. The proposed framework consists of a fast heuristic-based incremental allocation mechanism that dynamically performs resource allocation and a re-optimisation algorithm that periodically adjusts allocation over time. An offline algorithm estimates the desired response time with minimum resources, and the auto-scaling and load-balancing algorithm makes provision for workload variations. When the capacity violation detection algorithm identifies a failure of the auto-scaling mechanism, a network latency

constraint greedy algorithm initialises an NFV-enabled edge node to cope with the failure. This system will be referred to as System II.

The work in [121] proposed a provisioning and auto-scaling algorithm for edge node resources. The proposed auto-scaling mechanism was designed to manage the workload for maximising the performance of containers that are hosted on the edge by periodically monitoring resource utilisation. This system will be known as System III. Finally, System IV will describe the generic threshold-based auto-scaling mechanism in which the question of *when* to allocate resources as well as *how many* resources to be allocated is decided by a fixed value. For instance, whenever the fog node capacity is less than or above the threshold, auto-scaling may be performed.

The summary of the various systems is provided in Table 4.2.

Table 4.2. Summary of resource management systems

Name	Description
System I	The proposed system implementing a reactive auto-scaling algorithm.
System II	An alternative system implementing an auto-scaling and load-balancing algorithm.
System III	An alternative system implementing an auto-scaling algorithm.
System IV	The proposed system implementing a threshold-based scaling algorithm.

4.5 Evaluation Setup

4.5.1 Simulation

A smart farm use case is considered, as described in the previous chapter. In the simulation, UEs generate data after regular time interval and transmit it to fog nodes for processing. Since the output after processing is usually small, the simulation only considers the uplink communication for the environment.

A LTE eNB is deployed as the master node and a NR gNB as the secondary node, with UEs using the dual radio interfaces of both LTE and NR for connectivity.

142 UEs are connected to the RAN, and uplink packets are generated continuously throughout the simulation time, which is set to 300 s. The gNB is also connected to the SDN network consisting of an OpenFlow controller and OpenFlow switches. OpenFlow, which is a network standard, defines the interface that allows the controller to instruct the switch on how to handle incoming data packets.

The parameters used in the simulation are listed in Table 4.3 below.

Table 4.3: Simulation parameter settings

Parameter	**Value**
Configuration parameters	
LTE bandwidth	20 MHz
LTE link capacity	75 Mbps
LTE carrier frequency	1800 MHz
mmWave bandwidth	2.16 GHz
mmWave carrier frequency	60 GHz
X2 data rate	10 Gb/s
X2 link delay	50 ms
gNB transmission power	46 dBm
eNB transmission power	23 dBm
Application parameters	
Transport layer protocol	TCP
Number of nodes	4
Simulation time	300 s
Number of simulation runs	100
Confidence interval	0.95
SDN parameters	
Inter-switch data rate	10 Mbps
Switch-GW data rate	100 Mbps
Switch-gNB data rate	100 Mbps

4.5.2 Performance Metrics

The following metrics were computed for the purpose of performance analysis of the proposed reactive auto-scaling mechanism:

- User satisfaction ratio (%)
- End-to-end latency (ms)

- Cost efficiency (%)
- Throughput (Mbps)

4.6 Performance Evaluation

In this section, the performance of the proposed reactive auto-scaling model (System I) is evaluated against System II, System III and System IV. The results are presented in terms of latency, throughput, cost efficiency and user satisfaction.

4.6.1 Demonstration of Auto-Scaling

The relationship between the number of active requests that approximates demand and capacity represented by the number of free compute resources in the system is presented for the proposed reactive auto-scaler in Figure 4.4. The data plotted was measured in every 30 second interval for the duration of the simulation period (300 s). In every time slot, the number of free resources is either equal to or slightly higher than the number of active requests. This demonstrates the effectiveness of the proposed auto-scaling algorithm because every increase or decrease in the number of active requests is accompanied by a corresponding increase or decrease in the number of free resources.

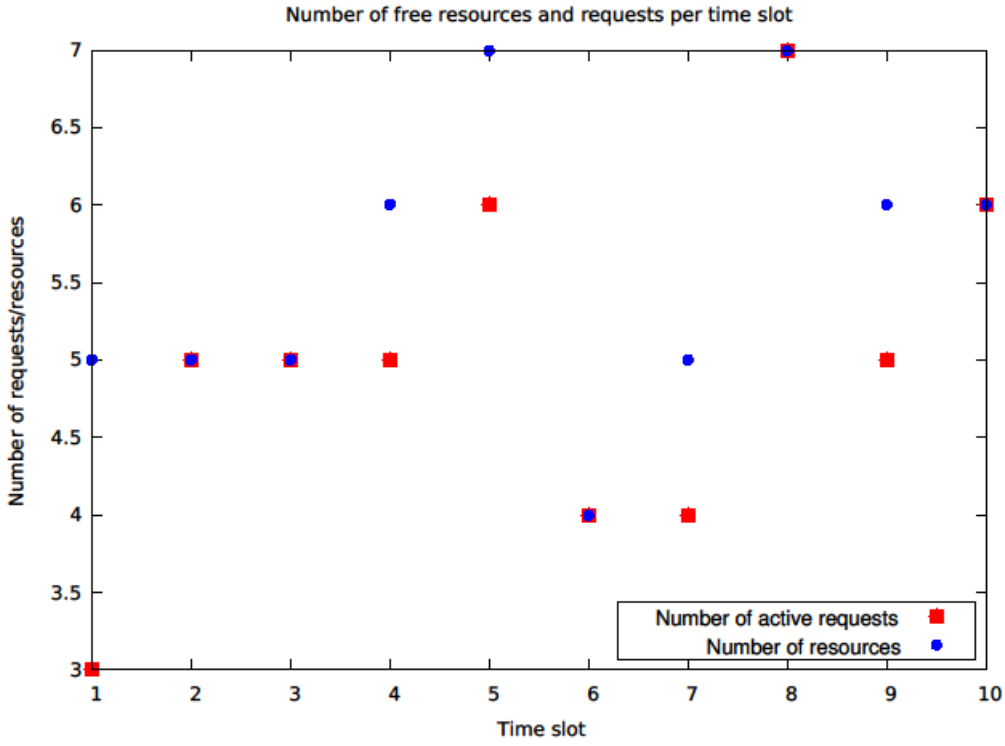


Figure 4.4. Relationship between the number of active requests and free resources

4.6.2 Impact of Latency Requirements

In order to take into account the resource utilisation of the fog nodes, cost efficiency is measured, which quantifies the percentage of users who receive their services within the services' latency requirements. The maximum tolerable latency requirements are categorised as ultra-low (< 1 ms), low (10 ms), medium (100 ms), high (150 ms), and mixed (randomly selected between 0 and 150 ms). Figure 4.5 illustrates the impact of maximum tolerable latency requirement on cost efficiency. The graph shows that extremely strict network latency requirements are less cost effective than more tolerant latencies. Systems I, II and III achieves a cost efficiency above 50% for all latency requirements, while System IV achieves a maximum of 45% cost efficiency for a workload with lenient latency requirements. Therefore, it can be concluded that dynamic auto-scaling is more efficient than the fixed threshold counterpart. System II appears to be more cost efficient and outperforms all three systems for each considered latency requirement, however System I is comparable. For flexible latency

requirements, both System I and System II achieve an optimal operational cost where the efficiency is equal to 100 percent.

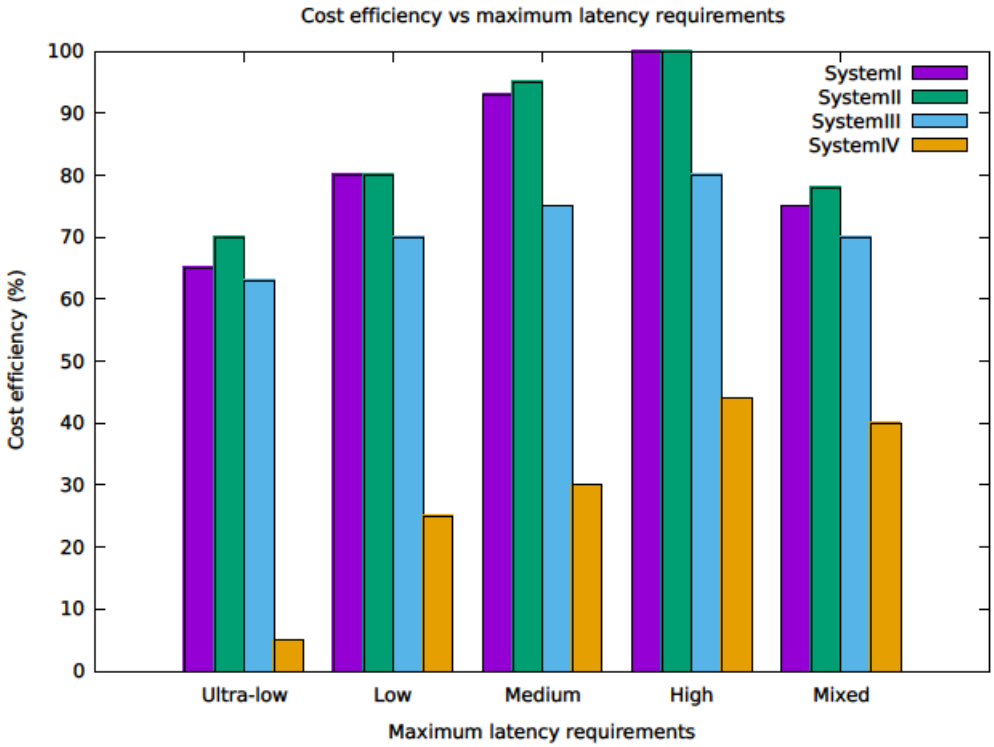


Figure 4.5. Impact of latency requirements on cost efficiency

4.6.3 Data Transmitted

Figure 4.6 shows the average amount of data transferred by varying the number of connected users in the proposed architecture, compared with the cloud-only model. It is observed that using the proposed resource management architecture, a significant reduction in the amount of data transferred between users and the cloud can be achieved. On average, the amount of data transferred between the fog node and the cloud server is reduced by up to 90%, which is encouraging particularly for mMTC applications which are characterised by a large volume of data generated by connected sensors. The proposed auto-scaling architecture facilitates data processing closer to the users at the fog nodes such that very little traffic is transmitted beyond the local network. This is promising for applications in underserved communities.

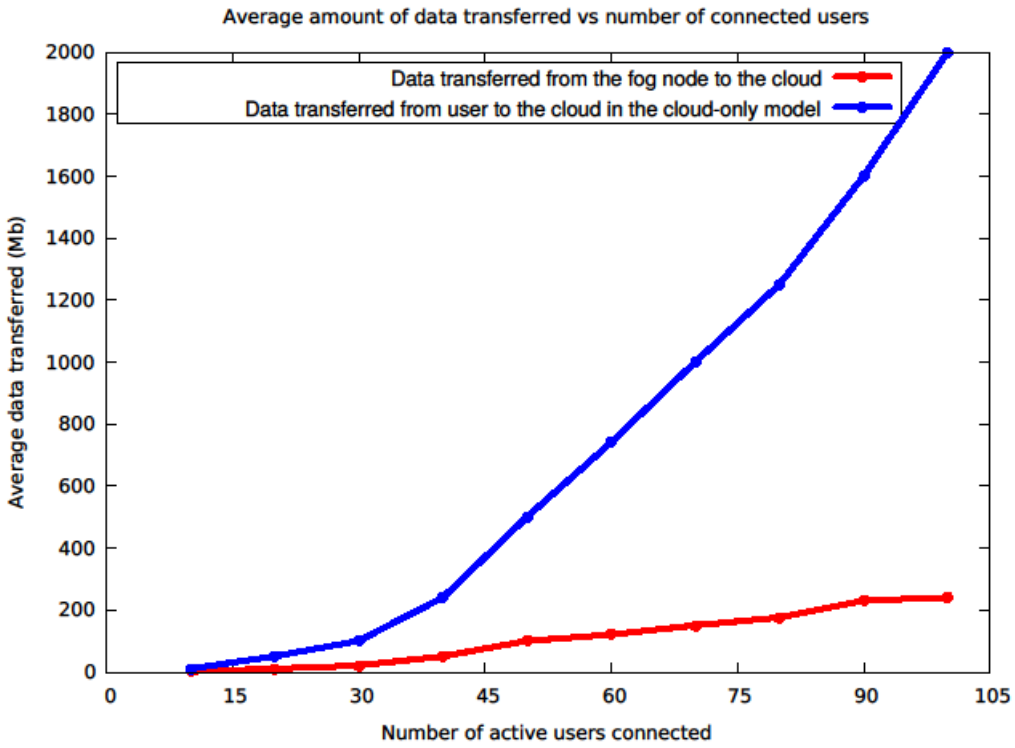


Figure 4.6. Average data transmitted vs number of connected users

4.6.4 Latency

A network latency comparison between the proposed reactive auto-scaling mechanism and Systems II, III and IV is shown in Figure 4.7. The end-to-end delay in fulfilling a request increases with an increase in network traffic load. However, in System IV, the latency increase much more significantly with traffic demand. As the need for computational resource grows, the fog nodes become increasingly constrained and more requests are offloaded to the cloud for processing.

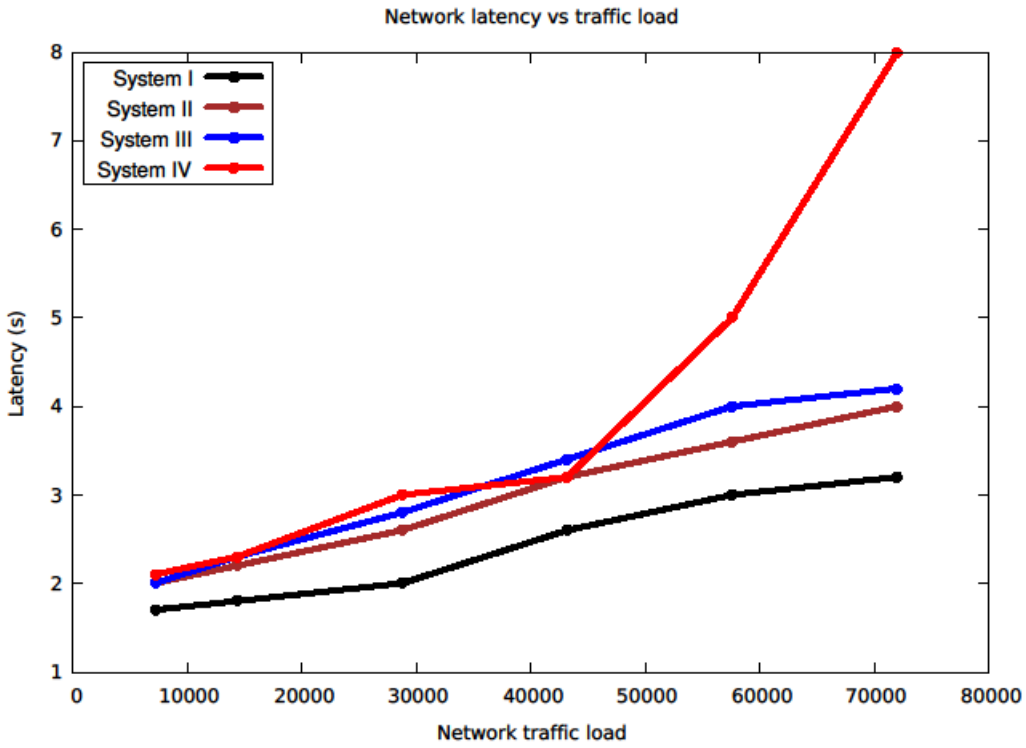


Figure 4.7. Impact of network traffic load on latency

4.6.5 Throughput

Figure 4.8 shows the system throughput comparison against network traffic load. It should be noted that because the LTE link capacity has been configured to 75 Mbps, the upper bound for the achievable throughput as defined in equation (**3.3**) is 100 Mbps. However, by aggregating both LTE and 5G NR bandwidths through dual connectivity, significant throughput improvements can be realised. System throughput increases substantially with a rise in network traffic load until it reaches the plateaux then starts to either fluctuate or increase gradually. As the network traffic load grows, Systems I-III always manage to achieve throughput above the upper bound for the achievable throughput, while System IV obtains a maximum throughput of 106 Mbps and starts to deteriorate.

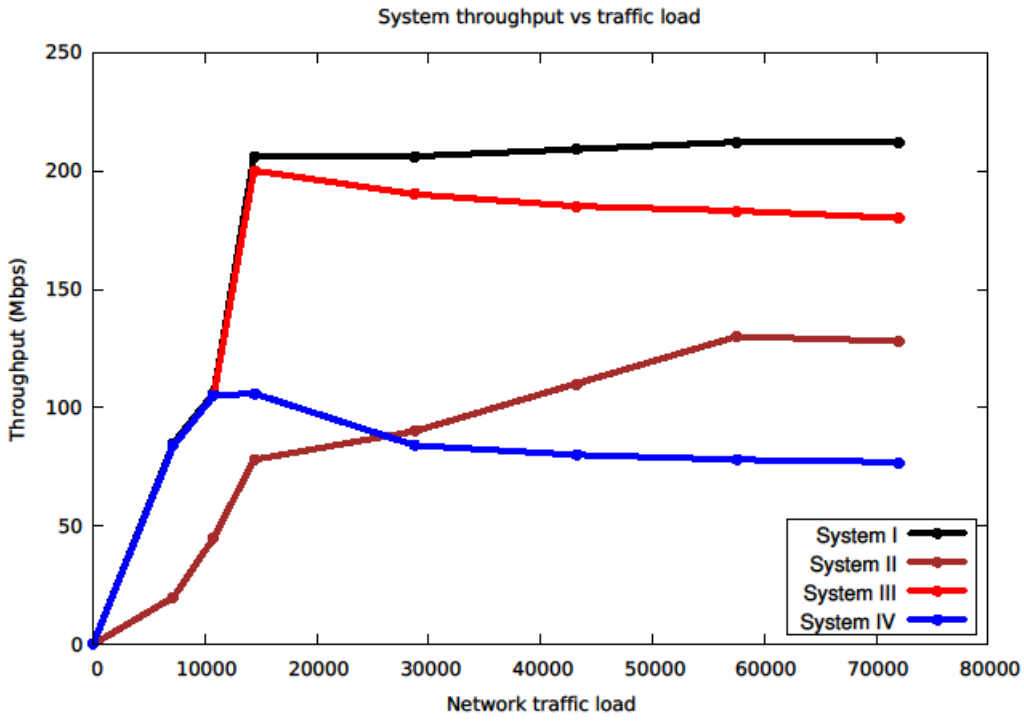


Figure 4.8. Impact of network traffic load on throughput

4.6.6 User Satisfaction

The ratio between the achieved throughput and the maximum achievable throughput, referred to as user satisfaction probability, can also be calculated. Figure 4.9 depicts the achieved user satisfaction probability as a function of the number of users. All the approaches compared exhibit a high user satisfaction probability, which begins to deteriorate with an increase in the number of users. However, the rates at which the models decline show significant differences. As illustrated, the fixed threshold approach of System IV shows the worst performance, while the proposed reactive auto-scaling mechanism (System I) outperforms all the methods. System II is comparable to the proposed System I.

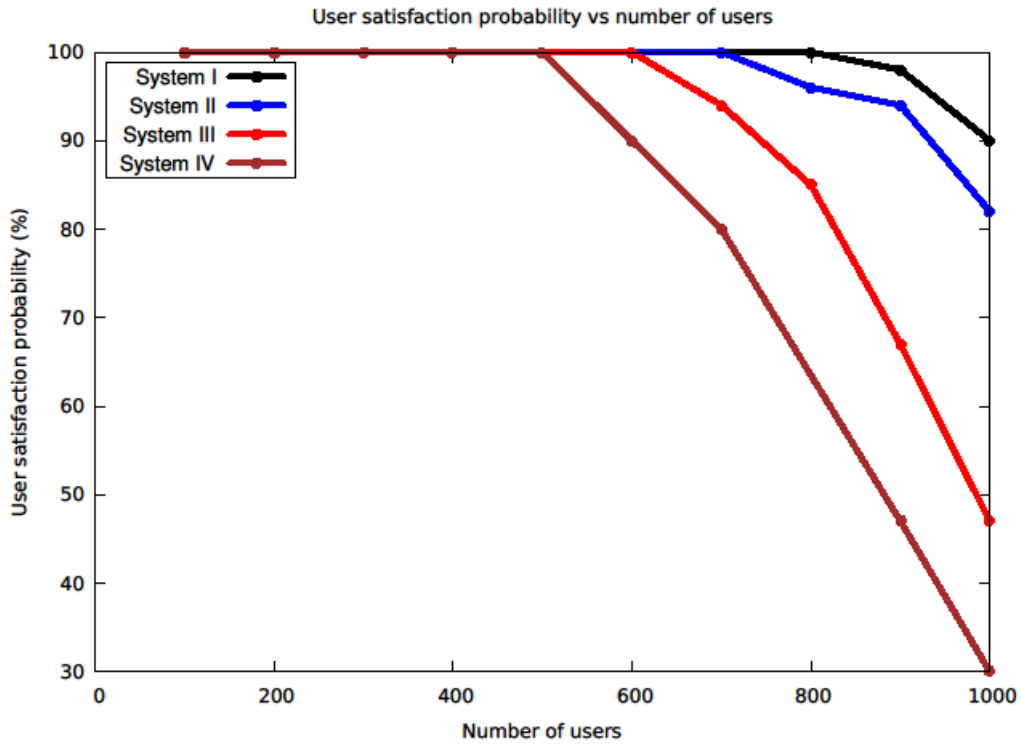


Figure 4.9. User satisfaction ratio comparison

4.7 Discussion

The goal of the proposed reactive auto-scaling system is to minimise the total end-to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint. It has been demonstrated that the proposed framework always manages to allocate sufficient resources in time to guarantee continuous satisfaction of applications' low latency requirements under dynamic workloads. The amount of data transferred is an important metric to gauge the communication frequency between the fog nodes and the remote cloud server. Reducing the amount of data transferred between fog nodes and the cloud also decreases the transmission delay, which has an impact on end-to-end latency. Furthermore, reduced frequency of communication between fog nodes and cloud servers reduces the propagation delay, since the distance between users and fog nodes is much shorter than the distance to the remote cloud and therefore there is a fewer number of hops for packets to travel. Given that round-trip latency proportionately declines

when the number of hops and packet size drops [122], one can deduce that the proposed algorithm leads to lower latency. This is supported by the measurements of latency against varying traffic loads, which demonstrated that the proposed algorithm exhibits the minimum latency. Additionally, it was observed that the proposed reactive auto-scaling algorithm allows all users to receive their services within the services' latency requirements when the requirements are lenient. However, improvements should be considered for applications with ultra-low latency requirements.

4.8 Conclusion

This chapter presented a dynamic resource management technique based on reactive auto-scaling, along with the architecture in which the algorithm is implemented. The proposed reactive auto-scaling algorithm performs better than the fixed threshold approach and other auto-scaling methods. However, reactive resource management mechanisms are a sub-optimal solution when considering the needs of future IoT applications, particularly ultra-low latency and cost efficiency. Therefore, performance improvements must be considered. The next chapter presents a proactive technique based on machine learning as a solution to the resource management issue in 5G F-RAN.

Chapter 5 – Proactive Auto-scaling Resource Allocation Technique based on Reinforcement Learning

5.1 Introduction

Auto-scaling techniques are an efficient solution to the resource management problem in 5G F-RAN, as demonstrated in Chapter 4 through the design and evaluation of the reactive auto-scaling algorithm. However, reactive decision policies are sub-optimal because they are unable to deal with variable traffic patterns in a timely manner, and they cannot predict future traffic demand and future resource utilisation in order to prepare for the adjustment of resources in advance. The main drawback of this approach is that it is incapable of adapting to the anticipated demands of 5G networks, such as ultra-low latency and high network scalability, to achieve the objective. An improved solution for the F-RAN resource allocation problem is to use reinforcement learning techniques which can continuously learn the environment and adapt the decision rule accordingly.

This chapter focuses on proactive auto-scaling using the reinforcement learning technique. Section 5.2 describes the proposed system parameters for the reinforcement learning model. The proposed proactive auto-scaling algorithm is presented in section 5.3, followed by a description of other resource management techniques considered for comparison and the evaluation setup in section 5.4 and 5.5, respectively. After the discussion of results in section 5.6, the chapter is concluded with a summary of key findings.

5.2 Proposed Reinforcement Learning Model

The need to deal with the ongoing proliferation of the scale, complexity, and connectivity of IoT services has necessitated the development of computing systems that are capable of self-management [123]. Conventional legacy approaches to system management involve algorithms that are programmed to work according to a predetermined case-based reasoning method and gradually fail when there are unpredictable changes in the workload. Hence, the proposed approach to the resource allocation problem leans towards an autonomous method in order to dynamically manage the heterogeneity of 5G applications.

In general, machine learning is aimed at constructing models and algorithms that can learn to make decisions directly from data without following pre-defined rules [40]. In the case of reinforcement learning, as illustrated in Figure 5.1 [124], learning takes place as a result of interaction between an agent and the environment (system). The agent gains rewards from the environment for every action it takes, and once the optimum policy of actions is learned, the agent will be able to maximise the measure of reward in the long run (expected cumulative reward), adapt to the environment, and achieve the goal [125].

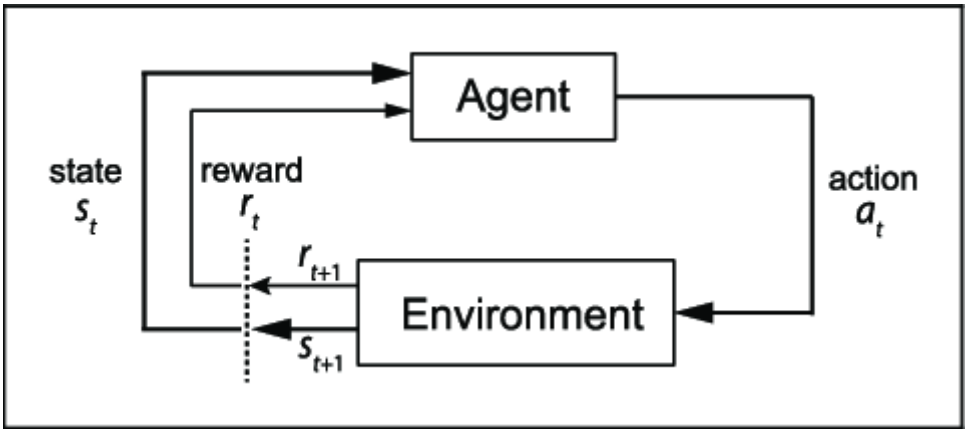


Figure 5.1. Reinforcement learning model

In general, because of the uncertainty and randomness properties, reinforcement learning problems can be modelled as a Markov Decision Process (MDP), which describe that the current state of the environment (system) is dependent on the last state and the action taken on it. The MDP model is defined as STAR, where *S* defines the set of system states, *T* is the set of probabilistic transition from current state s_t to the next state s'_t, on the action a_t, *A* is the set of actions and *R* is the set of feedback reward for the action a_t on state s.

Reinforcement learning can be portrayed as a collection of trial and error method by which a learner agent learns to make good decisions through a sequence of interactions with a system or an environment. Each interaction involves (i) observing the system's current state s_t at time t, (ii) take some legal action a_t in-state s_t to derive the new state s_{t+1} and finally (iii) receiving a reward r_t, a numerical value that the agent would like to maximize by taking better decision from its experience in long term.

The key MDP parameters for the F-RAN system are defined as follows:

- State: The current system state s_t is determined by the data processing requirements of users and the state of resource availability in the fog network. The system state at time slot t is defined as $s_t = [v, e, t, R_i, R_a, R_x] \in S$, where v denotes the sum of user requests, e represents the sum of request arrival rate, R_a is the percentage of resource allocation, $R_x$is the percentage of allocated resources currently unused, R_i is the sum of minimum allocation requirement, and t defines the sum of maximum delay requirement
- Action: The action at time instant t is defined as $a_t = \{upscape, downscale, no\ operation\}$
- Reward: When an agent takes an action, the networks are monitored, recording the link delays, packet drops and virtual network resource utilisation so as to determine a reward. The reward is determined by the link delay D_{ij}, packet drop ratio P_i and resource allocation ratio R_a and resource utilization R_u. r_t is then defined by:

$$r_t = \begin{cases} -100, & R_a \leq R_{min} \\ \alpha R_u - (\beta D_{ij} + \theta P_i), & otherwise \end{cases}$$

(5.1)

Where α , β , θ and are constants that adjust the influence of R_u, D_{ij} and P_i on the overall reward, and R_{min} is the threshold for minimum resource allocation. The objective of the reward function is to encourage high virtual resource utilisation while punishing nodes for dropping packets and links for having a high delay. A punitive reward of -100 to R_a below R_{min} has also been assigned to ensure that this is the minimum allocation to a virtual resource and therefore avoids adverse effects to QoS in cases of fast changes from very low to high virtual network (VN) loading.

- Next state: A node's change of state to a new state will be dictated by the predicted number of expected requests at the next time slot $t + 1$. In order to predict the expected request arrival rate (δ_n), linear regression statistical modelling is employed. The general form of the linear regression model is given by equation (5.2):

$$Y_{t+1} = aX_t + b$$

(5.2)

Where X_t denotes the sample service request, Y_{t+1} is he number of expected service requests, and t is the time value when the request was taken. The values of a and b can be calculated by solving the linear regression equation as given by equation (5.3) and equation (5.4) below:

$$a = \frac{\sum {X_t}^2 \sum Y_t - \sum X_t \sum X_t Y_t}{n \sum {X_t}^2 - (\sum X_t)^2}$$

(5.3)

$$b = \frac{n\sum X_t Y_t - \sum X_t \sum Y_t}{n\sum {X_t}^2 - (\sum X_t)^2} \quad (5.4)$$

Where n is the total number of service requests received.

5.3 Proposed Proactive Auto-Scaling Algorithm

To describe the procedure of the proposed method, the RL agent compares the number of requests that arrived at a time t and the amount of free resources in fog resource pool, then it classifies the state of the current system as either over-utilized or under-utilized. If the state is found to be over-utilized then an upscaling operation is initiated. On the other hand, if the state found to be underutilized, it is assumed that there exist excess resources that are not utilized; hence those resources need to be released back to the resource pool through a downscaling operation. In case neither of the two states, the system resumes its normal operation.

Given the prediction about the expected number of service requests at time $t + 1$, the RL agent decides whether the next state of the system will go over-utilized or under-utilized. Then the agent recommends for the appropriate action to be taken either to upscale or downscale or perform no scaling operation depending on the requirement of resources and the predicted future availability.

The described state-action mapping for the scaling decision is shown in Table 5.1.

Table 5.1: State-action mapping

	State	**Action**
Time t: $v > f$ Time t+1: $\delta > e$	Over-utilisation	Upscale
Time t: $v = f$ Time t+1: $\delta = e$	Normal operation	No scaling operation
Time t: $v < f$ Time t+1: $\delta < e$	Under-utilisation	Downscale

If the entire environment model is known, a MDP problem can be easily resolved by some dynamic programming methods, such as value iteration and policy iteration. However, in most cases, the state transition probability function and reward function are not known in advance. Q-learning [126] is a model-free reinforcement learning algorithm which can be used to find optimal policies by learning from previous decision-making experiences. The term model free refers to an algorithmic technique that does not need a prior trained model to take dynamic decisions. It does not rely on complete *a priori* knowledge of the environment. Following the basic idea of reinforcement learning, agents constantly perform actions in different states and then observe state transitions and relevant rewards.

Among several RL techniques, Q-learning requires low computational resources for its implementation and does not require the knowledge of the model of the environment, thus being a suitable learning technique for the resource-constrained fog nodes [127]. Furthermore, Q-learning has been used extensively to address resource allocation problems [128], thus being a suitable learning technique for the problem.

The Q-learning algorithm is expressed by the Q-function $Q(s, a)$ where at time t an action a_t is taken on the current state s_t which will lead to the next state s_{t+1}, and $\gamma \in [0, 1]$ is the discount factor which describes how much future reward affects current decision. It is used to finitely evaluate the overall expected reward for an infinite sequence of decisions. The Q-function is then updated by the Bellman equation (5.5):

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max a_{t+1} Q(s_{t+1}, a_{t+1}) \tag{5.5}$$

The pseudocode illustrating procedures of the proposed proactive auto-scaling RL-based resource allocation algorithm is presented in Algorithm 5.1.

Algorithm 5.1. Proactive auto-scaling algorithm

1. **Initialise** number of fog nodes
2. **for** each time step t **do**
3. **for** each fog node **do**
4. Read the total number of requests
5. **for** each request **do**
6. Collect processing requirements
7. Read resource availability
8. Collect the request arrival rate
9. Calculate the expected number of service requests at time $t + 1$ using equation (5.2)
10. **end for**
11. Initialise Q-values table of pairs (s, a) by zero
12. Observe the current state s_t
13. Choose an action a_t based on the state-action table
14. Perform the action a_t, receive the feedback reward r_{t+1} to reach the next state s_{t+1}
15. Update the Q-value table using equation (5.5)
16. $s_t = s_{t+1}$
17. **end for**
18. $t = t + 1$
19. **end for**

5.4 Description of Other Systems

The proposed reinforcement learning framework in this chapter implementing a proactive auto-scaling algorithm based on Q-learning will be referred to as System I. Meanwhile, the reactive auto-scaling mechanism described in Chapter 4 will be known as System II. The work in [129] described several RL-based methods for resource allocation in FRAN architectures. The algorithm based on SARSA will be referred to as System III, while the Monte Carlo mechanism is System IV.

5.5 Evaluation setup

5.5.1 Simulation Parameters

As an extension of the simulation environment described in section 4.5, Table 5.2 lists the parameters related to the reinforcement learning system.

Table 5.2. Reinforcement learning system parameters

Parameter	Value
Discount factor	0.7
Learning rate	0.01
Number of iterations	1000
Minimum allocation threshold	0.25

5.5.2 Performance Metrics

The following metrics were computed for the purpose of performance analysis of the proposed proactive auto-scaling algorithm:

- CPU utilisation (%): This is calculated as a percentage of the available processor time as :

$$R_u = \frac{R_a - R_x}{R_a} \tag{5.6}$$

- Virtual link utilisation (%): This measures the average traffic over a virtual link expressed as a percentage of the total link capacity. Neglecting the processing time and acknowledgement transmission time, link utilisation is given by [130]:

$$U_{k_n}^{link} = \frac{T_{k_n}^{tran}}{T_{k_n}^{tran} + 2T_{k_n}^{prop}}$$

$$= \frac{1}{1 + 2a} \tag{5.7}$$

  Where $T_{k_n}^{tran}$ and $T_{k_n}^{prop}$ denote the transmission time and propagation delay respectively, and $a = \frac{T_{k_n}^{prop}}{T_{k_n}^{tran}}$.

- Latency (ms)

- Cost efficiency (%)
- Algorithm efficiency (%)

5.6 Performance Evaluation

In this section, the performance of the proposed autonomous resource allocation reinforcement learning model is evaluated.

5.6.1 Comparison with the Proposed Reactive Auto-Scaling Technique

As part of performance evaluations, resource utilisation of the proposed proactive auto-scaling algorithm (System I) is measured against the reactive auto-scaling approach where VMs provisioning is performed based on a fixed scaling threshold (System II). The CPU utilisation of fog nodes is measured with every time slot, as illustrated in Figure 5.2. The proposed reinforcement learning-based algorithm far surpasses the reactive approach in terms of CPU usage. This is because in the former, resources are dynamically allocated based on the actual traffic demand, with unused resources being released back into the resource pool to be reused by other VMs. The inferior performance of the dynamic Q-learning approach can be attributed to the initial learning period of the agent. In the early stages of the simulation when the agent is still learning, fog nodes are allocated less resources than their demand. However, the algorithm progressively learns the Q-function, updating it only for the visited states if and only when visited.

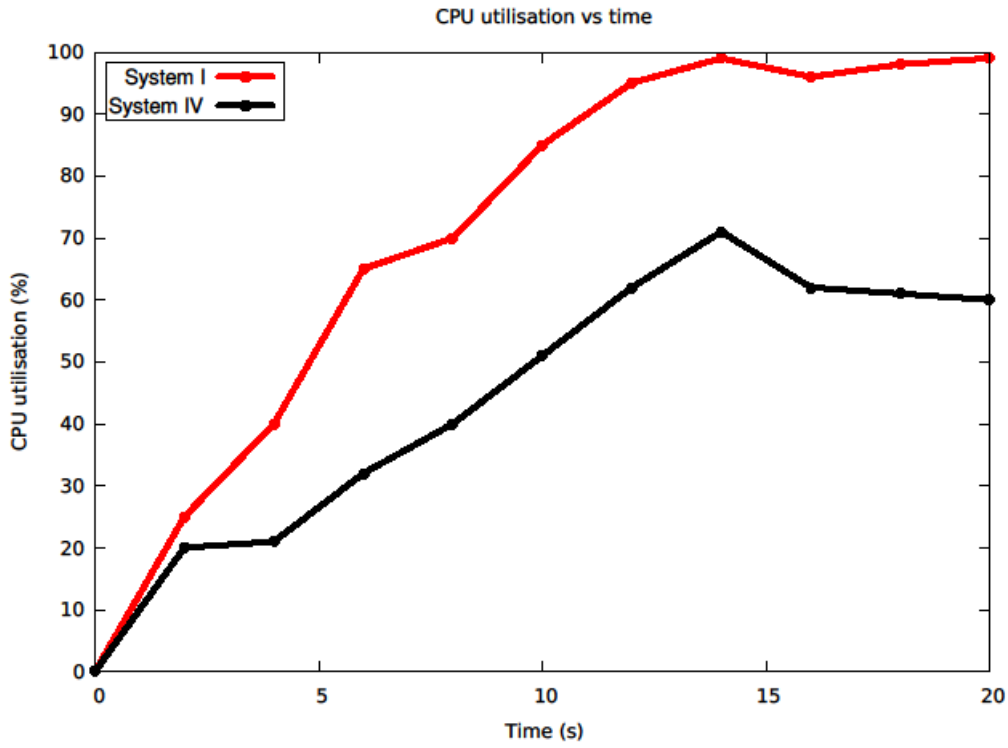


Figure 5.2. CPU utilization comparison

As shown in Figure 5.3, the proposed proactive auto-scaling mechanism performed better than the reactive auto-scaling approach in terms of link utilisation. Therefore, the proactive approach achieves better link efficiency because the virtual links in the proactive model utilise more bandwidth than the links in their reactive counterparts.

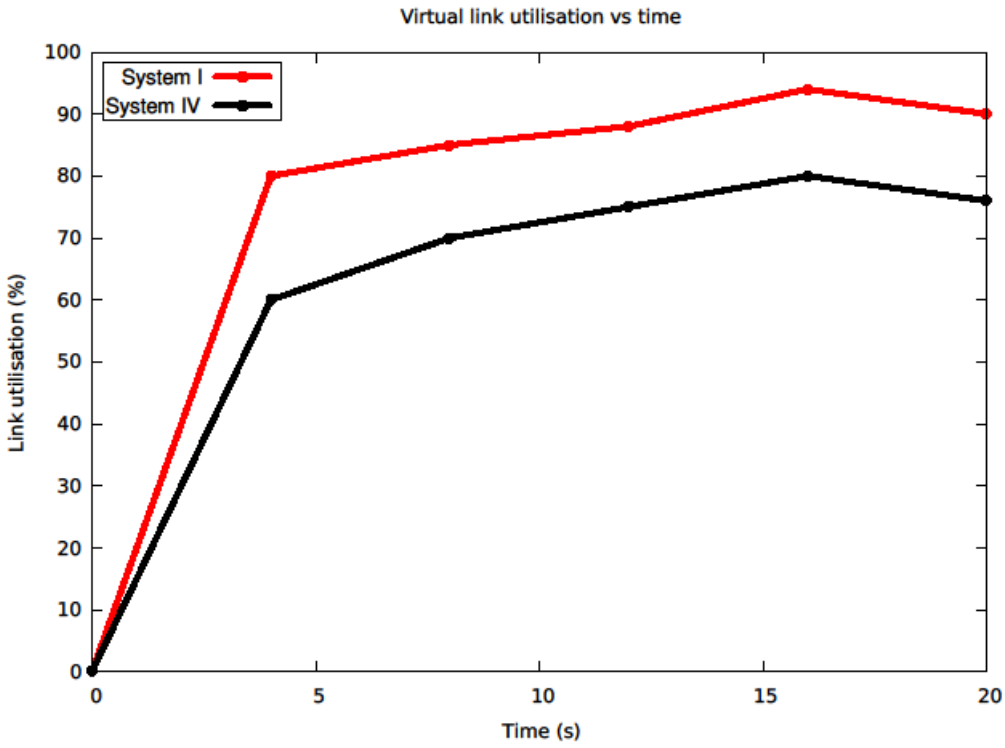


Figure 5.3. Virtual link utilization comparison

The sum of latency experienced by users is measured in Figure 5.4. The maximum total latency experienced by users in System I is 77 ms, while System II shows a sixfold increase. The inferior performance of System II can be attributed to the reactive nature of the algorithm, which makes decisions related to resource provisioning as a response to changes in current system workload. On the other hand, the proactive algorithm learns from the current workload in order to predict future availability and prepare resources accordingly. System I achieves a lower latency due to the optimisation of computation resources.

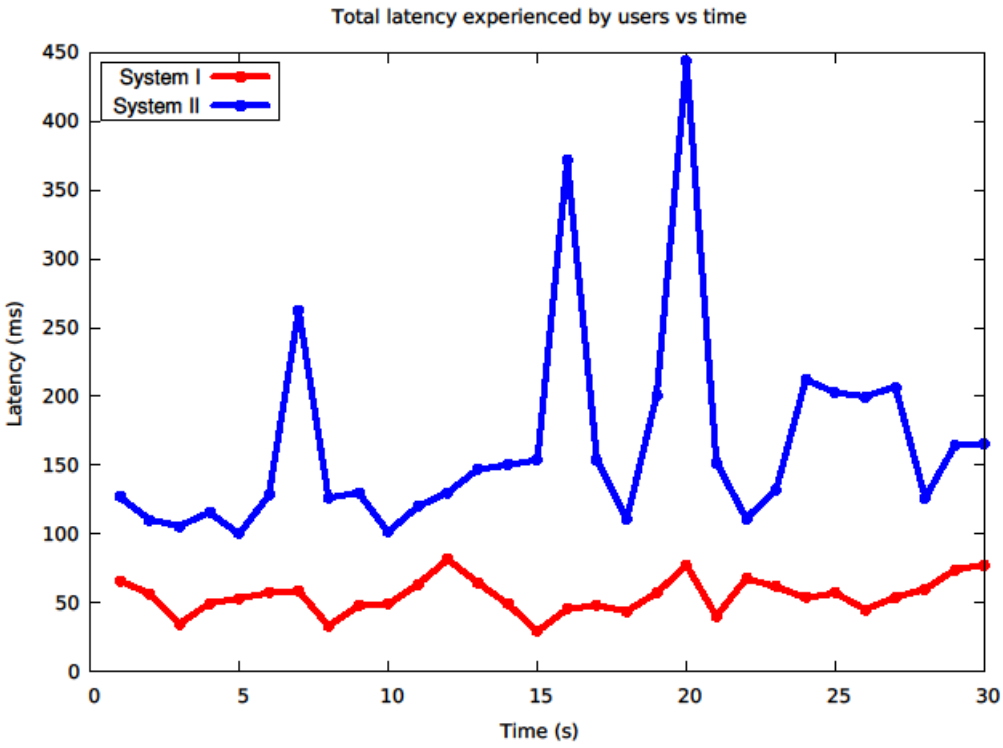


Figure 5.4. Sum of latency experienced by users in proactive vs reactive system

5.6.2 Comparison with Other RL Techniques

The graph in Figure 5.5 shows the sum of latency experienced by users in the RL-based systems. The general pattern is marked by an exponential decrease in latency in the initial stages of training until an equilibrium is reached. As illustrated, System I converges to the minimum total latency of 3 ms after 300 iterations, while System III reaches 4,4 ms after 450 iterations and System IV requires 550 iterations to obtain a minimum latency of 5,4 ms. The maximum latencies observed for Systems I, III and IV are 20 ms, 40 ms and 60 ms, respectively.

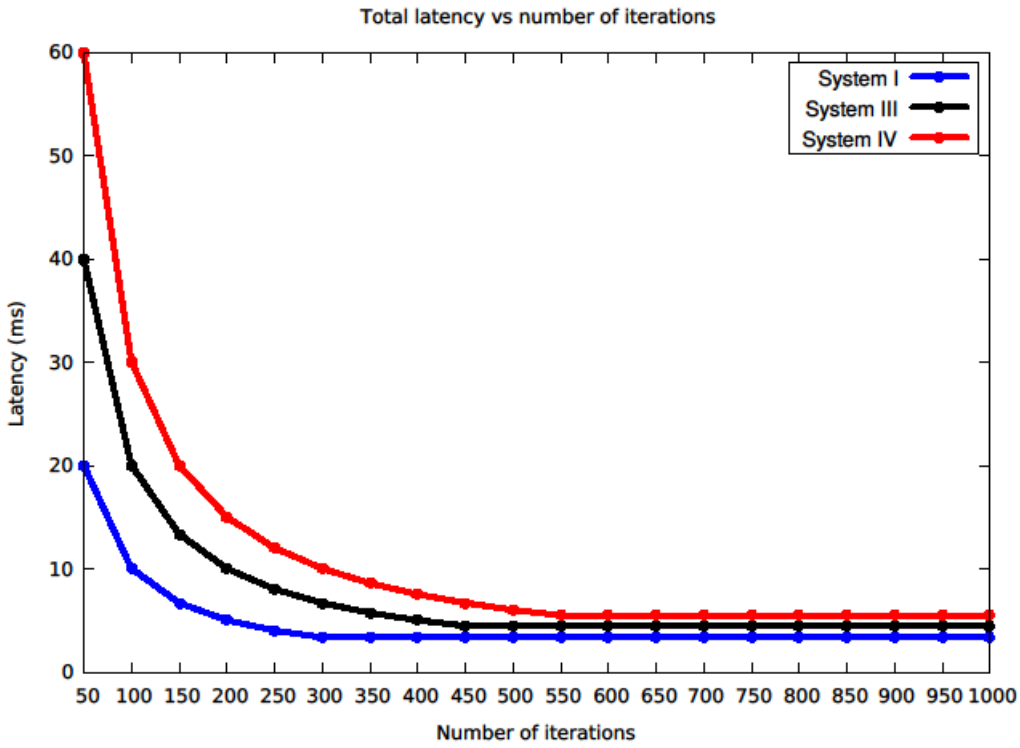


Figure 5.5. Sum of latency comparison of RL systems

The percentage of users who receive their services within the application's latency requirements, referred to as cost efficiency, was also measured for a system in which the latency requirements are random integers between zero and 150 ms. The cost efficiency observed, illustrated in Figure 5.6, exhibits poor performance in the initial stages and converges to optimal values. The beginning of the training period marks the agent's initial learning period, thus the curve of cost efficiency is at a state of constant fluctuation. System I achieves the highest maximum efficiency of 92% after 300 iterations, while System III and System IV achieve a maximum efficiency of 88% and 81% after 450 and 550 iterations, respectively.

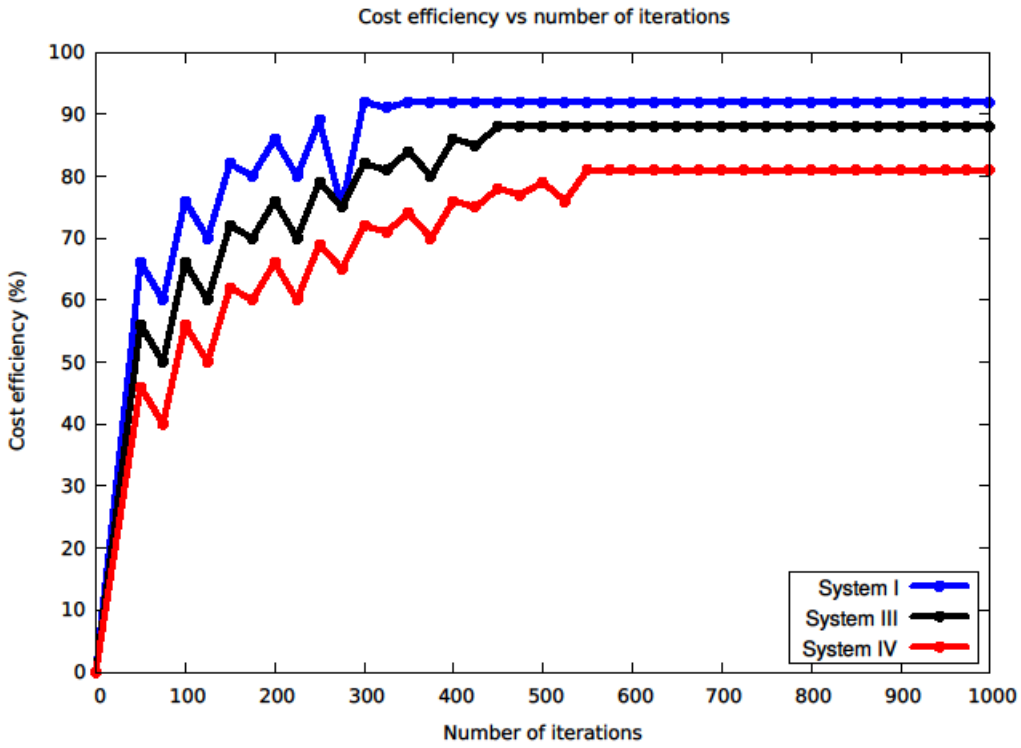


Figure 5.6. Cost efficiency comparison of RL systems

The measured CPU utilisation, which quantifies the percentage of the available processor time, is shown in Figure 5.7. In all three systems, resources are dynamically allocated based on the actual traffic demand, with unused resources being released back into the resource pool to be reused by other VMs. CPU utilisation increases with the rise in the number of iterations, with System I achieving the highest CPU utilisation. System I obtains the highest maximum CPU utilisation earlier than the other systems.

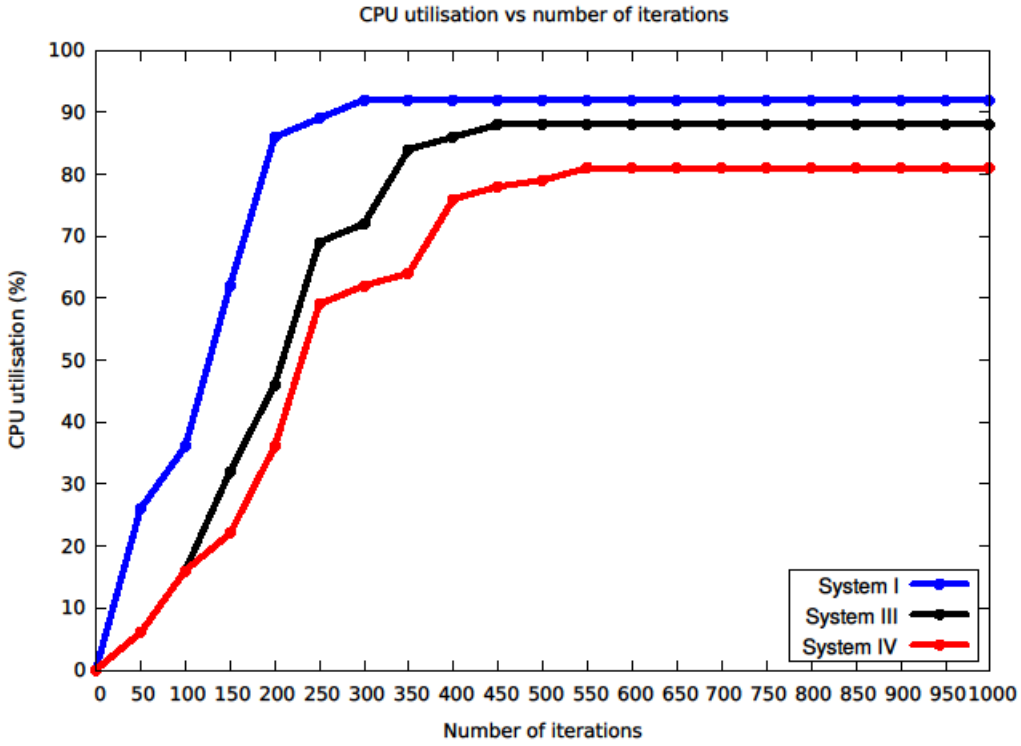


Figure 5.7. CPU utilisation comparison of RL systems

5.7 Discussion

The proposed proactive auto-scaling algorithm achieves minimum latency through computation resource allocation while ensuring maximum CPU utilisation and maximum link utilisation, compared with the reactive auto-scaling counterpart. This serves as a demonstration of the potential of machine learning capabilities in 5G F-RAN architectures for resource management. In comparison with other reinforcement learning systems, namely SARSA and Monte Carlo, the proposed Q-learning algorithm achieved a higher percentage of users who receive their services within the application's latency requirements. Furthermore, the proposed reactive auto-scaling algorithm converges faster than the other systems.

5.8 Conclusion

This chapter proposed a proactive auto-scaling algorithm, based on reinforcement learning, as a means to ensure minimum total end-to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint. The proposed proactive auto-scaling algorithm performs better than the proposed reactive auto-scaling algorithm and other popular reinforcement learning algorithms.

Chapter 6 – Concluding Remarks

6.1 Introduction

Fog computing is the proposed architecture for dealing with the growing number of IoT devices and the increasing amount of traffic anticipated in 5G networks. Furthermore, the fog radio access network is cost-effective and resource friendly, thus making it feasible for deployment in underserved areas. Chapter 1 introduces the issue of resource allocation in F-RANs and recognises that in order to design a 5G F-RAN system that is capable of autonomous and dynamic management, artificial intelligence notions and machine learning techniques must be integrated.

A background and literature survey of machine learning in 5G networks and fog computing was presented in Chapter 2 . The design of 5G F-RAN resource management architectures was explained, including the challenges prominent in existing approaches and the corresponding practical solutions for potential implementation. Several relevant approaches were extracted from literature to illustrate the capabilities of machine learning techniques and highlight their potential in future mobile networks.

Chapter 3 examined the 5G F-RAN architecture and provided justification for its selection. In this chapter, the overall system architecture was defined, and the issue of resource allocation for 5G F-RAN systems was formulated as an optimisation problem to minimise the total end-to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint. Furthermore, the considered system model was examined as part of a simulation against the conventional C-RAN model.

Chapter 4 focused on reactive auto-scaling as a proposed method for resource allocation in 5G F-RAN architectures. A resource allocation framework was constructed to support the dynamic provision of resources in response to changes in current system workload, and a dynamic algorithm was devised based on

reactive auto-scaling. The performance evaluation of the algorithm was carried out through simulations.

In Chapter 5 , reinforcement learning is presented as an improved solution for the F-RAN resource allocation problem. In particular, the resource allocation problem was modelled as a Markov Decision Process and an algorithm was designed based on Q-learning and proactive auto-scaling. Simulation experiments were conducted to evaluate the performance of the algorithm.

The remainder of this chapter summarises the dissertation as follows. Sections 6.2 and 6.3 discuss the initial objectives and how they were achieved. After highlighting the key scientific findings in Section 6.4, Section 6.5 outlines the benefits of this research. The final chapter recommendations and proposed future work are discussed in Section 6.6.

6.2 Statement of Initial Objectives

The main goal of this dissertation was to leverage the capabilities of machine learning and fog computing in order to address the computing resource allocation problem in 5G F-RAN architectures for mMTC services in underserved communities. To that end, the key objectives of this research were the following:

Objective 1: To investigate how machine learning-based techniques have been utilised in fog computing and 5G networks to address various challenges, and their potential applications in 5G F-RAN architectures.

Objective 2: To design a resource allocation architecture for mMTC applications in 5G F-RAN systems.

Objective 3: To develop a machine learning algorithm to address the problem of dynamic and autonomous allocation of computing resources in 5G F-RAN architectures.

6.3 Achieved Objectives

- **Investigation of machine learning applications:** As part of the investigation into the application of machine learning techniques in fog computing and 5G, Chapter 2 conducted a literature survey and presented the state-of-the-art.
- **Resource allocation architecture for mMTC applications in 5G F-RAN systems:** Section 4.2 presents a resource management architecture that consists of a request manager, decision maker, resource allocator, auto-scaler and scheduler. The functions of the components are also discussed. Section 4.3 discusses the implementation of the proposed resource management framework through the presentation of a dynamic auto-scaling algorithm. In Section 5.2, a model is presented based on reinforcement learning. The system parameters of the model are defined, which include the state, action, reward and next state.
- **Machine learning algorithm for dynamic and autonomous resource allocation:** A dynamic auto-scaling algorithm is devised in Section 5.3 based on Q-learning for dynamic and autonomous resource allocation.

6.4 Key Research Findings

Through the simulations conducted as part of performance evaluations of the proposed resource allocation methods, these were the key findings of this research:

- Reactive methods for resource allocation in 5G F-RANs are a sub-optimal solution to the problem of minimising the total end-to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint because they are unable to deal with variable traffic patterns in a timely manner. While the proposed reactive approach allows all users to receive their services within the services' latency requirements when the requirements are lenient, improvements should be considered for applications with

ultra-low latency requirements. For mMTC applications, where latency is not a stringent requirement, reactive methods may be sufficient.

- As expected, proactive methods are more efficient than their reactive counterpart for resource allocation. In particular, through the application of reinforcement learning, this work demonstrated the potential of machine learning capabilities in fog-enabled 5G networks.
- In comparison to SARSA and Monte Carlo, Q-learning is more efficient in terms of CPU utilisation and cost efficiency, which is a measure of the percentage of users who receive their services within the services' latency requirements.
- The introduction of intelligent functions at the edge is capable of creating a dynamic and autonomous system that can efficiently meet the demands of IoT applications and services in next-generation wireless networks.

6.5 Benefits of the Study

This research focused on the integration of fog computing, machine learning and 5G technologies as a means to aid 5G deployment in underserved communities. To that end, the benefits of this work are divided into three parts. Firstly, by exploiting the capabilities of the fog computing architecture to configure a 5G network with reduced cost, this research contributes to advancing the limited body of knowledge about making efforts to deploy 5G in underserved regions of developing countries as a means to bridge the digital divide. Secondly, in comparison to URLLC and eMBB, mMTC applications in 5G are an areas that is lesser explored. There is a lack of studies in the domain of utilising enhanced next-generation network features such as 5G New Radio to support deployment scenarios for mMTC services and applications. Therefore, by modelling an mMTC application to measure the performance of the proposed methods, this research makes an effort to validate the envisioned requirements of IoT applications in 5G networks. Finally, there is a limited number of studies that discuss the integration of fog computing, machine learning and 5G. By using machine learning techniques to address the resource allocation problem in 5G F-

RAN architectures, this research has contributed to the area of machine learning applications in fog computing and 5G.

6.6 Recommendations and Future Work

The following are possible directions for the extension of this research in the future:

- Due to a lack of real datasets containing resource measurements in 5G networks, a simulation environment was constructed as a representation of the network. However, simulation modelling is prone to limitations, with the main one being that the network is only as good as the rules and assumptions upon which it is based. As a result, future extensions of this research may be motivated to evaluate the proposed techniques in a real-world network such as a testbed. In addition, there is a shortage of studies on the design of 5G testbeds. To this end, further research could be inspired to contribute.
- This research focused on mMTC applications by modelling a smart farming use case. It would be interesting to see how the proposed solution performs for eMBB and URLLC applications, like in a network slicing architecture.
- The resource management techniques presented in this work could be extended to consider network resources and other computing resources such as memory and disk storage.
- The auto-scaling approaches discussed in this thesis use only horizontal auto-scaling techniques in which the number of resources is increased or decreased for handling changes in the workload. A combination of horizontal scaling with vertical scaling of compute resources needs further investigation.
- A hybrid algorithm based on reactive and proactive auto-scaling presents an interesting direction for future work.

References

[1] A. Gupta and R. K. Jha, ‘A survey of 5G network: Architecture and emerging technologies’, *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[2] E. K. Markakis, K. Karras, A. Sideris, G. Alexiou, and E. Pallis, ‘Computing, Caching, and Communication at the Edge: The Cornerstone for Building a Versatile 5G Ecosystem’, *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 152–157, 2017.

[3] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ‘Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks’, *ArXiv Prepr. ArXiv171002913*, 2017.

[4] S. Kitanov, E. Monteiro, and T. Janevski, ‘5G and the Fog—Survey of related technologies and research directions’, 2016, pp. 1–6.

[5] ‘IEEE 5G and Beyond Technology Roadmap White Paper’, IEEE, Oct. 2017. [Online]. Available: https://futurenetworks.ieee.org/images/files/pdf/ieee-5g-roadmap-white-paper.pdf.

[6] A. Checko *et al.*, ‘Cloud RAN for Mobile Networks—A Technology Overview’, *IEEE Commun. Surv. Tutor.*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015, doi: 10.1109/COMST.2014.2355255.

[7] P. Chanclou, A. Pizzinat, Y. denis, and sebastien randazzo, ‘C-RAN architecture and fronthaul challenges’, Jan. 2015.

[8] Z. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, and H. Franke, ‘Virtual base station pool: towards a wireless network cloud for radio access networks’, in *Proceedings of the 8th ACM International Conference on Computing Frontiers - CF ’11*, Ischia, Italy, 2011, p. 1, doi: 10.1145/2016604.2016646.

[9] Y.-J. Ku *et al.*, ‘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.

[10] 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.

[11] D. Pouillot, ‘The Dynamics of Broadband Markets in Europe: Realizing the 2020 Digital Agenda’, *Commun. Strateg.*, no. 99, p. 183, 2015.

[12] S. Lavanya, N. M. S. Kumar, S. Thilagam, and S. Sinduja, ‘Fog computing based radio access network in 5G wireless communications’, in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017, pp. 559–563, doi: 10.1109/WiSPNET.2017.8299819.

[13] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, ‘Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing’, *Sensors*, vol. 19, no. 9, May 2019, doi: 10.3390/s19092122.

[14] C. Hernández-Chulde and C. Cervelló-Pastor, ‘Intelligent Optimization and Machine Learning for 5G Network Control and Management’, in *Highlights*

of Practical Applications of Survivable Agents and Multi-Agent Systems. The PAAMS Collection, 2019, pp. 339–342.
[15] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, 'Machine learning in agriculture: A review', *Sensors*, vol. 18, no. 8, p. 2674, 2018.
[16] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, 'Traffic and Computation Co-Offloading With Reinforcement Learning in Fog Computing for Industrial Applications', *IEEE Trans. Ind. Inform.*, vol. 15, no. 2, pp. 976–986, Feb. 2019, doi: 10.1109/TII.2018.2883991.
[17] L. Huang, X. Feng, C. Zhang, L. Qian, and Y. Wu, 'Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing', *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 10–17, Feb. 2019, doi: 10.1016/j.dcan.2018.10.003.
[18] Y. Wei, F. R. Yu, M. Song, and Z. Han, 'Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled IoT Using Natural Actor–Critic Deep Reinforcement Learning', *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019, doi: 10.1109/JIOT.2018.2878435.
[19] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, 'Enabling intelligence in fog computing to achieve energy and latency reduction', *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 3–9, Feb. 2019, doi: 10.1016/j.dcan.2018.10.008.
[20] Y. Zeng, A. Al-Quzweeni, T. E. H. El-Gorashi, and J. M. H. Elmirghani, 'Energy Efficient Virtualization Framework for 5G F-RAN', in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, Angers, France, Jul. 2019, pp. 1–4, doi: 10.1109/ICTON.2019.8840170.
[21] G. M. S. 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.
[22] S. Lavanya, N. M. S. Kumar, S. Thilagam, and S. Sinduja, 'Fog computing based radio access network in 5G wireless communications', in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017, pp. 559–563, doi: 10.1109/WiSPNET.2017.8299819.
[23] Y. Zhou, M. Peng, S. Yan, and Y. Sun, 'Deep Reinforcement Learning Based Coded Caching Scheme in Fog Radio Access Networks', in *2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Aug. 2018, pp. 309–313, doi: 10.1109/ICCChinaW.2018.8674478.
[24] Md. S. Hossain, M. R. Ramli, J. M. Lee, and D.-S. Kim, 'Fog Radio Access Networks in Internet of Battlefield Things (IoBT) and Load Balancing Technology', in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2019, pp. 750–754, doi: 10.1109/ICTC46691.2019.8939722.
[25] L. Ruan, Z. Liu, X. Qiu, Z. Wang, S. Guo, and F. Qi, 'Resource allocation and distributed uplink offloading mechanism in fog environment', *J. Commun. Netw.*, vol. 20, no. 3, pp. 247–256, Jun. 2018, doi: 10.1109/JCN.2018.000037.

[26] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, and A. Ghosh, 'LTE-M Evolution Towards 5G Massive MTC', in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOMW.2017.8269112.
[27] K. E. Skouby, I. William, and A. Gyamfi, *Handbook on ICT in Developing Countries: 5G Perspective*. River Publishers, 2017.
[28] 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.
[29] OpenFog Consortium Architecture Working Group, 'OpenFog reference architecture for fog computing', *OPFRA001*, vol. 20817, p. 162, 2017.
[30] P. Bellavista, L. Foschini, and D. Scotece, 'Converging Mobile Edge Computing, Fog Computing, and IoT Quality Requirements', in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug. 2017, pp. 313–320, doi: 10.1109/FiCloud.2017.55.
[31] 'A Smart Collaborative Policy for Mobile Fog Computing in Rural Vitalization'. https://www.hindawi.com/journals/wcmc/2018/2643653/ (accessed May 11, 2020).
[32] P. Hu, S. Dhelim, H. Ning, and T. Qiu, 'Survey on fog computing: architecture, key technologies, applications and open issues', *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, 2017.
[33] S. Agarwal, S. Yadav, and A. K. Yadav, 'An efficient architecture and algorithm for resource provisioning in fog computing', *Int. J. Inf. Eng. Electron. Bus.*, vol. 8, no. 1, p. 48, 2016.
[34] Rachid El Hattachi and Javan Erfanian, *5G White Paper*. NDMN Alliance, 2015.
[35] S. Telecom, 'SK Telecom's View on 5G Vision, Architecture, Technology, and Spectrum', *5G White Pap.*, 2014.
[36] M. Peng, Y. Li, Z. Zhao, and C. Wang, 'System Architecture and Key Technologies for 5G Heterogeneous Cloud Radio Access Networks', *IEEE Netw.*, vol. 29, no. 2, pp. 6–14, Mar. 2015, doi: 10.1109/MNET.2015.7064897.
[37] 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.
[38] L. Ferreira *et al.*, 'An architecture to offer cloud-based radio access network as a service', Jun. 2014, pp. 1–5, doi: 10.1109/EuCNC.2014.6882627.
[39] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, 'Architecture Harmonization Between Cloud Radio Access Networks and Fog Networks', *IEEE Access*, vol. 3, pp. 3019–3034, 2015, doi: 10.1109/ACCESS.2015.2509638.
[40] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, 'Machine learning for networking: Workflow, advances and opportunities', *IEEE Netw.*, vol. 32, no. 2, pp. 92–99, 2018.
[41] R. Boutaba *et al.*, 'A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities', *J. Internet Serv. Appl.*, vol. 9, May 2018, doi: 10.1186/s13174-018-0087-2.

[42] C. Shen, P. Wang, and A. van den Hengel, ‘Optimally Training a Cascade Classifier’, *ArXiv10083742 Cs*, Aug. 2010, Accessed: May 11, 2020. [Online]. Available: http://arxiv.org/abs/1008.3742.
[43] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou, ‘Machine Learning Differential Privacy With Multifunctional Aggregation in a Fog Computing Architecture’, *IEEE Access*, vol. 6, pp. 17119–17129, 2018, doi: 10.1109/ACCESS.2018.2817523.
[44] D. Wu *et al.*, ‘A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing’, *J. Manuf. Syst.*, vol. 43, pp. 25–34, Apr. 2017, doi: 10.1016/j.jmsy.2017.02.011.
[45] R. Samanta, C. Kumari, N. Deb, S. Bose, A. Cortesi, and N. Chaki, ‘Node localization for indoor tracking using artificial neural network’, in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, Apr. 2018, pp. 229–233, doi: 10.1109/FMEC.2018.8364071.
[46] J. L. Pérez, A. Gutierrez-Torre, J. Ll. Berral, and D. Carrera, ‘A resilient and distributed near real-time traffic forecasting application for Fog computing environments’, *Future Gener. Comput. Syst.*, vol. 87, pp. 198–212, Oct. 2018, doi: 10.1016/j.future.2018.05.013.
[47] A. Abeshu and N. Chilamkurti, ‘Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing’, *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 169–175, Feb. 2018, doi: 10.1109/MCOM.2018.1700332.
[48] U. Drolia, K. Guo, and P. Narasimhan, ‘Precog: prefetching for image recognition applications at the edge’, in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, San Jose California, Oct. 2017, pp. 1–13, doi: 10.1145/3132211.3134456.
[49] G. Grassi, M. Sammarco, P. Bahl, K. Jamieson, and G. Pau, ‘Poster: ParkMaster: Leveraging Edge Computing in Visual Analytics’, in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, Paris, France, Sep. 2015, pp. 257–259, doi: 10.1145/2789168.2795174.
[50] M. Hogan and F. Esposito, ‘Stochastic delay forecasts for edge traffic engineering via Bayesian Networks’, in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, Oct. 2017, pp. 1–4, doi: 10.1109/NCA.2017.8171341.
[51] I. Azimi *et al.*, ‘HiCH: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT’, *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 5s, p. 174:1-174:20, Sep. 2017, doi: 10.1145/3126501.
[52] D. Zissis, ‘Intelligent security on the edge of the cloud’, in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Jun. 2017, pp. 1066–1070, doi: 10.1109/ICE.2017.8279999.
[53] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, ‘QoS prediction for service recommendations in mobile edge computing’, *J. Parallel Distrib. Comput.*, vol. 127, pp. 134–144, May 2019, doi: 10.1016/j.jpdc.2017.09.014.
[54] A. Shrestha and A. Mahmood, ‘Review of Deep Learning Algorithms and Architectures’, *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
[55] U. von Luxburg, ‘A tutorial on spectral clustering’, *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007, doi: 10.1007/s11222-007-9033-z.

[56] D. Borthakur, H. Dubey, N. Constant, L. Mahler, and K. Mankodiya, ‘Smart fog: Fog computing framework for unsupervised clustering analytics in wearable Internet of Things’, in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2017, pp. 472–476, doi: 10.1109/GlobalSIP.2017.8308687.
[57] E. Balevi and R. D. Gitlin, ‘Unsupervised machine learning in 5G networks for low latency communications’, in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, Dec. 2017, pp. 1–2, doi: 10.1109/PCCC.2017.8280492.
[58] E. Balevi and R. D. Gitlin, ‘Optimizing the Number of Fog Nodes for Cloud-Fog-Thing Networks’, *ArXiv180100831 Cs*, Jan. 2018, Accessed: Jun. 11, 2019. [Online]. Available: http://arxiv.org/abs/1801.00831.
[59] D. Kimovski, H. Ijaz, N. Saurabh, and R. Prodan, ‘Adaptive Nature-Inspired Fog Architecture’, 2018, pp. 1–8.
[60] J. Schneible and A. Lu, ‘Anomaly detection on the edge’, in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, Oct. 2017, pp. 678–682, doi: 10.1109/MILCOM.2017.8170817.
[61] K. Das and R. N. Behera, ‘A Survey on Machine Learning: Concept,Algorithms and Applications’, *undefined*, 2017. https://www.semanticscholar.org/paper/A-Survey-on-Machine-Learning%3A-Concept%2CAlgorithms-Das-Behera/4d7855b8e5ef36acd4ac41deef596e67ac899e76 (accessed May 11, 2020).
[62] J. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, ‘Managing Fog Networks using Reinforcement Learning Based Load Balancing Algorithm’, *ArXiv190110023 Cs*, Jan. 2019, Accessed: May 14, 2020. [Online]. Available: http://arxiv.org/abs/1901.10023.
[63] V. 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, Mar. 2012, doi: 10.1109/TPDS.2011.156.
[64] 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, Aug. 2014, doi: 10.1109/TMC.2013.94.
[65] C. Wen, S. Jin, K. Wong, J. Chen, and P. Ting, ‘Channel Estimation for Massive MIMO Using Gaussian-Mixture Bayesian Learning’, *IEEE Trans. Wirel. Commun.*, vol. 14, no. 3, pp. 1356–1368, Mar. 2015, doi: 10.1109/TWC.2014.2365813.
[66] K. W. Choi and E. Hossain, ‘Estimation of Primary User Parameters in Cognitive Radio Systems via Hidden Markov Model’, *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 782–795, Feb. 2013, doi: 10.1109/TSP.2012.2229998.
[67] A. Assra, J. Yang, and B. Champagne, ‘An EM Approach for Cooperative Spectrum Sensing in Multiantenna CR Networks’, *IEEE Trans. Veh. Technol.*, 2016, doi: 10.1109/TVT.2015.2408369.

[68] C.-K. Yu, K.-C. Chen, and S.-M. Cheng, 'Cognitive Radio Network Tomography', *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1980–1997, May 2010, doi: 10.1109/TVT.2010.2044906.
[69] M. Xia, Y. Owada, M. Inoue, and H. Harai, 'Optical and wireless hybrid access networks: Design and optimization', *IEEEOSA J. Opt. Commun. Netw.*, vol. 4, no. 10, pp. 749–759, Oct. 2012, doi: 10.1364/JOCN.4.000749.
[70] 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, Dec. 2011, doi: 10.1109/TSG.2011.2160101.
[71] 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, Apr. 2013, doi: 10.1109/TWC.2013.022213.112260.
[72] 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, Oct. 2013, doi: 10.1109/JSTSP.2013.2258656.
[73] G. Alnwaimi, S. Vahid, and K. Moessner, 'Dynamic Heterogeneous Learning Games for Opportunistic Access in LTE-Based Macro/Femtocell Deployments', *IEEE Trans. Wirel. Commun.*, vol. 14, no. 4, pp. 2294–2308, Apr. 2015, doi: 10.1109/TWC.2014.2384510.
[74] O. Onireti *et al.*, 'A Cell Outage Management Framework for Dense Heterogeneous Networks', *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2097–2113, Apr. 2016, doi: 10.1109/TVT.2015.2431371.
[75] 'Channel Selection for Network-Assisted D2D Communication via No-Regret Bandit Learning With Calibrated Forecasting - IEEE Journals & Magazine'. https://ieeexplore.ieee.org/document/6939716 (accessed May 14, 2020).
[76] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, 'Enabling technologies for fog computing in healthcare IoT systems', *Future Gener. Comput. Syst.*, vol. 90, pp. 62–78, Jan. 2019, doi: 10.1016/j.future.2018.07.049.
[77] Y. Sun and F. Lin, 'Non-cooperative differential game for incentive to contribute resource-based crowd funding in fog computing', *Boletin Tec. Bull.*, vol. 55, pp. 69–77, Jan. 2017.
[78] G. Kamath, P. Agnihotri, M. Valero, K. Sarker, and W.-Z. Song, 'Pushing Analytics to the Edge', in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6, doi: 10.1109/GLOCOM.2016.7842181.
[79] I. Azimi, A. Anzanpour, A. M. Rahmani, P. Liljeberg, and T. Salakoski, 'Medical warning system based on Internet of Things using fog computing', in *2016 International Workshop on Big Data and Information Security (IWBIS)*, Oct. 2016, pp. 19–24, doi: 10.1109/IWBIS.2016.7872884.
[80] L. Lu, L. Xu, B. Xu, G. Li, and H. Cai, 'Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm', *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 4, pp. 1142–1151, Dec. 2018, doi: 10.1109/TCSS.2018.2871694.

[81] L. Li, K. Ota, and M. Dong, ‘Deep Learning for Smart Industry: Efficient Manufacture Inspection System With Fog Computing’, *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4665–4673, Oct. 2018, doi: 10.1109/TII.2018.2842821.
[82] D. Vimalajeewa, C. Kulatunga, and D. P. Berry, ‘Learning in the compressed data domain: Application to milk quality prediction’, *Inf. Sci.*, vol. 459, pp. 149–167, Aug. 2018, doi: 10.1016/j.ins.2018.05.002.
[83] M. Chen and V. C. M. Leung, ‘From cloud-based communications to cognition-based communications: A computing perspective’, *Comput. Commun.*, vol. 128, pp. 74–79, Sep. 2018, doi: 10.1016/j.comcom.2018.07.010.
[84] ‘Artificial intelligence based directional mesh network design for spectrum efficiency - IEEE Conference Publication’. https://ieeexplore.ieee.org/document/8396558 (accessed May 14, 2020).
[85] ‘Smart Cargo for Multimodal Freight Transport: When “Cloud” becomes “Fog” - ScienceDirect’. https://www.sciencedirect.com/science/article/pii/S2405896316308187 (accessed May 14, 2020).
[86] ‘Implementing Deep Learning and Inferencing on Fog and Edge Computing Systems - IEEE Conference Publication’. https://ieeexplore.ieee.org/document/8480168 (accessed May 14, 2020).
[87] H.-J. Hong, J.-C. Chuang, and C.-H. Hsu, ‘Animation Rendering on Multimedia Fog Computing Platforms’, in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Dec. 2016, pp. 336–343, doi: 10.1109/CloudCom.2016.0060.
[88] K.-C. Chen, T. Zhang, R. D. Gitlin, and G. Fettweis, ‘Ultra-Low Latency Mobile Networking’, *IEEE Netw.*, vol. 33, no. 2, pp. 181–187, Mar. 2019, doi: 10.1109/MNET.2018.1800011.
[89] J. Patman, M. Alfarhood, S. Islam, M. Lemus, P. Calyam, and K. Palaniappan, ‘Predictive analytics for fog computing using machine learning and GENI’, in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, Apr. 2018, pp. 790–795, doi: 10.1109/INFCOMW.2018.8407027.
[90] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, ‘Fog computing based efficient IoT scheme for the Industry 4.0’, in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, May 2017, pp. 1–6, doi: 10.1109/ECMSM.2017.7945879.
[91] S. Saraswat, H. P. Gupta, and T. Dutta, ‘Fog based energy efficient ubiquitous systems’, *2018 10th Int. Conf. Commun. Syst. Netw. COMSNETS*, 2018, doi: 10.1109/COMSNETS.2018.8328238.
[92] H. Malik, M. M. Alam, Y. Le Moullec, and Q. Ni, ‘Interference-Aware Radio Resource Allocation for 5G Ultra-Reliable Low-Latency Communication’, in *2018 IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOMW.2018.8644301.
[93] A. Mamane, M. E. Ghazi, G.-R. Barb, and M. Oteșteanu, ‘5G Heterogeneous Networks: An Overview on Radio Resource Management Scheduling

Schemes', in *2019 7th Mediterranean Congress of Telecommunications (CMT)*, Oct. 2019, pp. 1–5, doi: 10.1109/CMT.2019.8931369.
[94] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, 'Switch-On/Off Policies for Energy Harvesting Small Cells through Distributed Q-Learning', *2017 IEEE Wirel. Commun. Netw. Conf. Workshop WCNCW*, 2017, doi: 10.1109/WCNCW.2017.7919075.
[95] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, 'Distributed Q-learning for energy harvesting Heterogeneous Networks', in *2015 IEEE International Conference on Communication Workshop (ICCW)*, Jun. 2015, pp. 2006–2011, doi: 10.1109/ICCW.2015.7247475.
[96] M. Miozzo and P. Dini, 'Layered Learning Radio Resource Management for Energy Harvesting Small Base Stations', in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, Jun. 2018, pp. 1–6, doi: 10.1109/VTCSpring.2018.8417657.
[97] R. A. C. Bianchi, M. F. Martins, C. H. C. Ribeiro, and A. H. R. Costa, 'Heuristically-Accelerated Multiagent Reinforcement Learning', *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 252–265, Feb. 2014, doi: 10.1109/TCYB.2013.2253094.
[98] '(PDF) Radio Resource Allocation in 5G New Radio: A Neural Networks Approach'. https://www.researchgate.net/publication/329908540_Radio_Resource_Allocation_in_5G_New_Radio_A_Neural_Networks_Approach (accessed May 13, 2020).
[99] 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, p. 128, Dec. 2017, doi: 10.1186/s13638-017-0908-1.
[100] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, 'Learning Radio Resource Management in 5G Networks: Framework, Opportunities and Challenges', *IEEE Commun. Mag.*, vol. 56, Sep. 2018, doi: 10.1109/MCOM.2018.1701031.
[101] A. Ali, G. A. Shah, and J. Arshad, 'Energy Efficient Resource Allocation for M2M Devices in 5G', *Sensors*, vol. 19, no. 8, Apr. 2019, doi: 10.3390/s19081830.
[102] '5G radio resource management approach for multi-traffic IoT communications | Elsevier Enhanced Reader'. https://reader.elsevier.com/reader/sd/pii/S1389128618303876?token=F1A7B604609B73A19CE340722B68AB2C8864AC79E92257D5E0D8FA3298EC5BA48FC4EF25BC3BD1470C2268B63048718F (accessed May 13, 2020).
[103] Y. Sun, M. Peng, and S. Mao, 'Deep Reinforcement Learning-Based Mode Selection and Resource Management for Green Fog Radio Access Networks', *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019, doi: 10.1109/JIOT.2018.2871020.
[104] M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang, and C. X. Mavromoustakis, 'Joint Task Offloading and Resource Allocation for Delay-Sensitive Fog Networks', in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–7, doi: 10.1109/ICC.2019.8761239.

[105] T.-C. Chiu, A.-C. Pang, W.-H. Chung, and J. Zhang, ‘Latency-Driven Fog Cooperation Approach in Fog Radio Access Networks’, *IEEE Trans. Serv. Comput.*, vol. 12, no. 5, pp. 698–711, Sep. 2019, doi: 10.1109/TSC.2018.2858253.
[106] G. M. S. Rahman, M. Peng, S. Yan, and T. Dang, ‘Learning Based Joint Cache and Power Allocation in Fog Radio Access Networks’, *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4401–4411, Apr. 2020, doi: 10.1109/TVT.2020.2975849.
[107] Z. Zhao *et al.*, ‘On the Design of Computation Offloading in Fog Radio Access Networks’, *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7136–7149, Jul. 2019, doi: 10.1109/TVT.2019.2919915.
[108] 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. Supplement2, pp. 131–139, 2016, doi: 10.1109/CC.2016.7833467.
[109] A. Nassar and Y. Yilmaz, ‘Resource Allocation in Fog RAN for Heterogeneous IoT Environments Based on Reinforcement Learning’, in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761626.
[110] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, ‘Transmission and Latency-Aware Load Balancing for Fog Radio Access Networks’, in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647580.
[111] S. Sardellitti, G. Scutari, and S. Barbarossa, ‘Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing’, *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015, doi: 10.1109/TSIPN.2015.2448520.
[112] P. J. Fortier and H. Michel, *Computer Systems Performance Evaluation and Prediction*. USA: Butterworth-Heinemann, 2002.
[113] ‘Convex Optimization – Boyd and Vandenberghe’. https://web.stanford.edu/~boyd/cvxbook/ (accessed Jan. 14, 2021).
[114] S. Choi *et al.*, ‘5G K-SimNet: End-to-End Performance Evaluation of 5G Cellular Systems’, in *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651686.
[115] ‘(PDF) Auto-scaling Strategy for Amazon Web Services in Cloud Computing’. https://www.researchgate.net/publication/301935688_Auto-scaling_Strategy_for_Amazon_Web_Services_in_Cloud_Computing (accessed Jun. 07, 2020).
[116] A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, ‘Adaptive, Model-driven Autoscaling for Cloud Applications’, p. 9.
[117] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, ‘An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach’, *Future Gener. Comput. Syst.*, vol. 78, pp. 191–210, Jan. 2018, doi: 10.1016/j.future.2017.02.022.
[118] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, ‘Enhancing survivability in virtualized data centers: A service-aware approach’, *IEEE J. Sel. Areas*

Commun., vol. 31, no. 12, pp. 2610–2619, Dec. 2013, doi: 10.1109/JSAC.2013.131203.

[119] '(PDF) Seamless Support of Low Latency Mobile Applications with NFV-Enabled Mobile Edge-Cloud'. https://www.researchgate.net/publication/308511759_Seamless_Support_of_Low_Latency_Mobile_Applications_with_NFV-Enabled_Mobile_Edge-Cloud (accessed Jun. 11, 2020).

[120] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, 'Cost-Efficient NFV-Enabled Mobile Edge-Cloud for Low Latency Mobile Applications', *IEEE Trans. Netw. Serv. Manag.*, 2018, doi: 10.1109/TNSM.2018.2790081.

[121] 'ENORM: A Framework For Edge NOde Resource Management - IEEE Journals & Magazine'. https://ieeexplore.ieee.org/document/8039523 (accessed Jun. 13, 2020).

[122] A. U. Qureshi, *LIGHT WEIGHT MOBILE CLOUD COMPUTING ENVIRONMENT FOR MOBILE APPLICATIONS*. Concepts Books Publication.

[123] E. Casalicchio, D. Menascé, and A. Aldhalaan, 'Autonomic resource provisioning in cloud systems with availability goals', presented at the Proceedings of the 2013 ACM cloud and autonomic computing conference, Aug. 2013, doi: 10.1145/2494621.2494623.

[124] I. R. Galatzer-Levy, K. Ruggles, and Z. Chen, 'Data Science in the Research Domain Criteria Era: Relevance of Machine Learning to the Study of Stress Pathology, Recovery, and Resilience', *Chronic Stress Thousand Oaks Calif*, vol. 2, Dec. 2018, doi: 10.1177/2470547017747553.

[125] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, *Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues*. 2018.

[126] T. E. Bogale and X. Wang, 'MACHINE INTELLIGENCE TECHNIQUES FOR NEXT-GENERATION CONTEXT-AWARE WIRELESS NETWORKS', no. 1, p. 11, 2018.

[127] M. Moh and R. Raju, 'Machine Learning Techniques for Security of Internet of Things (IoT) and Fog Computing Systems', in *2018 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2018, pp. 709–715, doi: 10.1109/HPCS.2018.00116.

[128] S. K. Sharma and X. Wang, *Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions*. 2018.

[129] A. T. Nassar and Y. Yilmaz, 'Reinforcement Learning-based Resource Allocation in Fog RAN for IoT with Heterogeneous Latency Requirements', *ArXiv180604582 Cs*, May 2018, Accessed: Jun. 05, 2019. [Online]. Available: http://arxiv.org/abs/1806.04582.

[130] 'Server Utilization - an overview | ScienceDirect Topics'. https://www.sciencedirect.com/topics/computer-science/server-utilization (accessed Jan. 17, 2021).