QoS-aware Adaptive Data Dissemination in Mobile Edge Computing Ecosystem

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science & Engineering

By

Gagandeep Kaur

Registration Number: 41900514

Supervised By

Dr. Balraj Singh (13075)

Department of Computer Science & Engineering (Associate Professor)

Lovely Professional University, Punjab, India

Co-Supervised by

Dr. Ranbir Singh Batth (64540)

Lecturer and Unit Coordinator

Sydney International School of Technology and Commerce, Australia

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

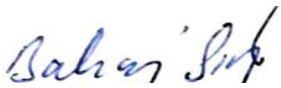
DECLARATION

I, hereby declared that the presented work in the thesis entitled **"QoS-aware Adaptive Data Dissemination in Mobile Edge Computing Ecosystem"** in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision Dr. Balraj Singh, working as Associate Professor in the Department of Computer Science and Engineering of Lovely Professional University, Punjab, India and Dr. Ranbir Singh Batth working as Lecturer and Unit Coordinator at Sydney International School of Technology and Commerce, Australia. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of another investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)
Gagandeep Kaur
41900514
Department of Computer Science and Engineering
Lovely Professional University, Punjab, India

CERTIFICATE

This is to certify that the work reported in the PhD. Thesis entitled **QoS-aware Adaptive Data Dissemination in Mobile Edge Computing Ecosystem** submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the Department of Computer Science and Engineering, is a research work carried out by Gagandeep Kaur, 41900514, is bonafede record of his original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



Signature of Supervisor
Name of Supervisor: Dr. Balraj Singh

Designation: Associate Professor
Department/school: School of Computer Science & Engineering
University: Lovely Professional University, Punjab, India

Signature of Co-Supervisor
Name of Co-Supervisor: Dr. Ranbir Singh Batth
Designation: Lecturer and Unit Coordinator
Department/school: Sydney International School of Technology and Commerce
Australia

Abstract

In the evolving paradigm of Mobile Edge Computing (MEC), ensuring Quality of Service (QoS) in dynamic, real-time environments present significant challenges due to fluctuating network topologies, heterogeneous resource constraints, and increasing demand from data-intensive applications. This thesis work addresses these challenges by implementing a series of adaptive, intelligent, and QoS-aware models rooted in bioinspired and machine learning approaches, aimed at enhancing traffic control, data dissemination, and resource scheduling in MEC deployments. A unique Dynamic Traffic Flow Control (DTFC) framework, combined with a QoS-aware Adaptive Data Dissemination Engine (QADE), was presented to address the issues of network congestion and delay. Based on temporal and geographical parameters, this model adaptively manages communication flows by utilizing a hybrid Elephant Herding Particle Swarm Optimizer (EHPSO) in conjunction with reinforcement learning approaches. During extensive simulations, the system showed notable gains in latency, throughput, energy efficiency, and packet delivery ratio. Additionally, using Flower Pollination Optimization (FPO) and the predictive ability of a VARMAx (Vector Autoregressive Moving Average with exogenous variables) model, a bioinspired resource scheduling model was created. By taking into account a wide range of task and resource characteristics, our hybrid architecture effectively mapped tasks to virtual machines. Additionally, it enabled the dynamic recalibration of virtual machine capacity by predicting future workloads, thereby improving scheduling effectiveness, energy conservation, and deadline adherence. Extensive tests on real-world datasets confirmed that the suggested models performed well in comparison to existing techniques. Altogether, this thesis work advances the state of the art in QoS-aware data dissemination and resource management, provides innovative, scalable, and intelligent solutions for MEC, and establishes a solid basis for upcoming real-time edge computing systems.

Acknowledgement

I am thankful to the Lovely Professional University, Phagwara, Punjab, for giving me an opportunity to pursue my Ph.D. It has been a great experience for me to pursue my research in this esteemed Organization. I am grateful to the Computer Science Department for providing the facilities during my research. I take this opportunity to thank all the people who have helped and supported me during my research. I am thankful to all the faculty and non-teaching staff within the department as well as outside the department for all the help that I have received during my Ph D at Lovely Professional University, Phagwara, Punjab.

Whenever I reach a milestone in life, I realize how little my effort would have meant without the support of so many people. I would like to start by thanking my supervisor, Dr. Balraj Singh, Associate Professor, Lovely Professional University, Phagwara, Punjab, for his valuableguidance and inputs towards my research work. I am indebted to my supervisor, Dr. Balraj Singh, under whose supervision I started my research work. I am thankful to him for the patience he has shown and the encouragement he has provided. The intelligence and approach to research of Dr. Balraj Singh have been a source of inspiration for me. I would like to thank him for his dedicated support, timely advice, inspiration, encouragement, and continuous support throughout my Ph.D. I am also thankful to Dr. Ranbir Singh Batth and Dr. Rachit Garg, who support me during this tenure.

On the successful completion of the research work and this thesis, I would like to pay all praise to the Almighty God. In this thesis, the work presented would not have been possible without my close association with many people. With utmost humbleness, I would like to take this opportunity to extend my deep sense of gratitude to all those who have directly or indirectly made this research thesis possible.

Gagandeep Kaur

Table of Contents

List of Figures

List of Tables

Glossary

MEC.....Mobile Edge Computing

QoS..... Quality of Service

DHR.....Deadline Hit Ratio

DTFC..... Dynamic Traffic Flow Control

EHPSO.....Elephant Herding Particle Swarm Optimizer

FPO..... Flower Pollination Optimization

PDR.....Packet Delivery Ratio

VM..... Virtual Machine

QADE..... Adaptive Data Dissemination Engine

RL..... Reinforcement Learning

MIPS..... Million Instructions Per Second

RAM..... Random Access Memory

IoT.....Internet of Things

SE.....Scheduling Efficiency

DoS RA.....Distributed Resource Allocation

PSO..... Particle Swarm Optimization

CNN.....Convolution Neural Network

ACO..... Ant Colony Optimization

LIST OF PUBLICATIONS

- ....Kaur, G., Singh, B., Batth, R.S. et al. BATFE: design of a hybrid bioinspired model for adaptive traffic flow control in edge devices. Microsyst Technol (2024). https://doi.org/10.1007/s00542-024-05826-5. **SCI IF 1.6 (Published)**

- ....Kaur, G., Singh, B. and Faheem, M. (2025), Bioinspired Adaptive Resource Scheduling for QoS in Mobile Edge Deployments. IET Commun., 19: e70017. https://doi.org/10.1049/cmu2.70017. **SCI IF 1.6 (Published)**

- ....Gagandeep Kaur, Balraj Singh, Ranbir Singh Batth, "Design of an efficient QoS-Aware Adaptive Data Dissemination Engine with DTFC for Mobile Edge Computing Deployments", International Journal of Computer Networks and Applications (IJCNA), 10(5), PP: 728-744, 2023, DOI: 10.22247/ijcna/2023/223420. **Scopus (Published)**

- ....G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 254-259, doi: 10.1109/ICIEM51511.2021.9445331. **Scopus (Published)**

- ....G. Kaur and B. Singh, "A Comprehensive Review on Edge Computing: Architecture, Applications and Resource Management Techniques", CODD 100 – 8th International Conference on Computing Sciences. **Scopus (Accepted and Presented)**

- ....G. Kaur, B. Singh and R. S. Batth, Mobile Edge Traffic Insights: A Comprehensive Study of Network Traffic Flow Control for Resource Allocation, Third International Conference on Next Generation Computing Systems ICNGCS - 2025. **Scopus (Accepted)**

CHAPTER 1

INTRODUCTION

The need for more intelligent, decentralized, and responsive computing infrastructures has been highlighted in recent years by the exponential expansion of data traffic, the quick spread of mobile devices, and the emergence of latency-sensitive applications. By moving computational resources and services closer to the data sources and end users, Mobile Edge Computing (MEC) has emerged as a paradigm-shifting approach to meet these demands. In addition to improving real-time processing and lowering latency and network congestion, this decentralization also improves the overall Quality of Service (QoS) that customers experience. MEC is now a key component for allowing applications like augmented reality, driverless cars, healthcare monitoring, and smart cities thanks to its incorporation into next-generation communication networks. Nevertheless, there are drawbacks to the advantages that MEC provides [1]. Achieving effective data distribution, appropriate resource scheduling, and adaptive traffic flow control is significantly hampered by the dynamic and resource-constrained nature of edge environments. Because of their limited flexibility and incapacity to react to real-time changes in network conditions, traditional static and centralized models frequently fail to meet these constraints. Furthermore, managing varying workloads, diverse devices, and geographically dispersed edge nodes makes it more difficult to maintain QoS metrics like low latency, high throughput, and reliability. Researchers have resorted to intelligent and adaptive models that can learn from and change in response to the dynamic network environment in order to get around these problems. Among these, bioinspired algorithms, which are based on the ideas of biological systems and natural evolution, have demonstrated exceptional promise. These algorithms are especially well-suited for resolving optimization issues in dynamic environments such as self-adaptation, robustness, and scalability [2]. For a variety of resource management applications, methods like hybrid fuzzy-logic systems, particle swarm optimization (PSO), ant

colony optimization (ACO), and genetic algorithms (GA) have been thoroughly investigated. By suggesting a QoS-aware adaptive framework for data distribution in MEC ecosystems and employing hybrid bioinspired methodologies to overcome important performance bottlenecks, the current thesis work adds to this changing landscape. The main goal of this thesis work is to improve the responsiveness and dependability of resource allocation and data distribution in MEC by creating models that are sensitive to real-time QoS requirements and adaptive. This thesis work takes a three-pronged strategy to achieving this goal, with each element being thoroughly explored and examined in the following chapters. The creation of a hybrid bioinspired model for adaptive traffic flow control is the main goal of this work. In addition to violating QoS restrictions, traffic congestion at the edge layer can significantly impair application performance. To dynamically control the data flow in this situation, a hybrid model that combines fuzzy logic and evolutionary computing is suggested. This model is based on the priority of data packets, bandwidth availability, and network congestion levels. In order to provide smoother data flow and lower latency, the model may self-tune its settings to adjust to various traffic scenarios. A novel model for QoS-aware data dissemination based on a Dynamic Traffic Flow Control (DTFC) mechanism is presented. The purpose of this implemented model is to guarantee that data packets are distributed throughout the network in a way that gives priority to QoS metrics including service criticality, packet loss rate, and delivery deadline [3]. The dissemination strategy makes dynamic, well-informed judgments about data processing and routing by taking into account the application's context and the state of the network nodes. The end-user experience is guaranteed to be constant even with fluctuating network loads thanks to the incorporation of DTFC within the MEC environment. This thesis work provides significant contribution with bioinspired adaptive resource scheduling paradigm to solve the problem of resource scarcity at the edge. The advantages of bioinspired intelligence to distribute communication and processing resources in a way that strikes a compromise between efficiency and maintaining quality of service. By adjusting to patterns of resource

demand and real-time input, it optimizes scheduling choices for both fairness and throughput. This enhances the overall responsiveness of the system and makes a substantial contribution to the long-term operation of MEC nodes. These three elements work together to provide a strong and coherent plan for QoS-aware adaptive data distribution in MEC. In addition to filling important gaps in the literature, this thesis work provides useful models for real-world deployment by fusing the adaptive capabilities of bioinspired algorithms with domain-specific insights into edge computing environments. Moreover, thorough testing and comparative analysis have been used to assess each of the suggested framework's efficacy in raising QoS metrics in dynamic operating environments. The overall architecture of the proposed QoS-aware adaptive framework is illustrated in figure 1.1, highlighting the flow of data and decision-making across MEC nodes and bioinspired optimization layers [4].

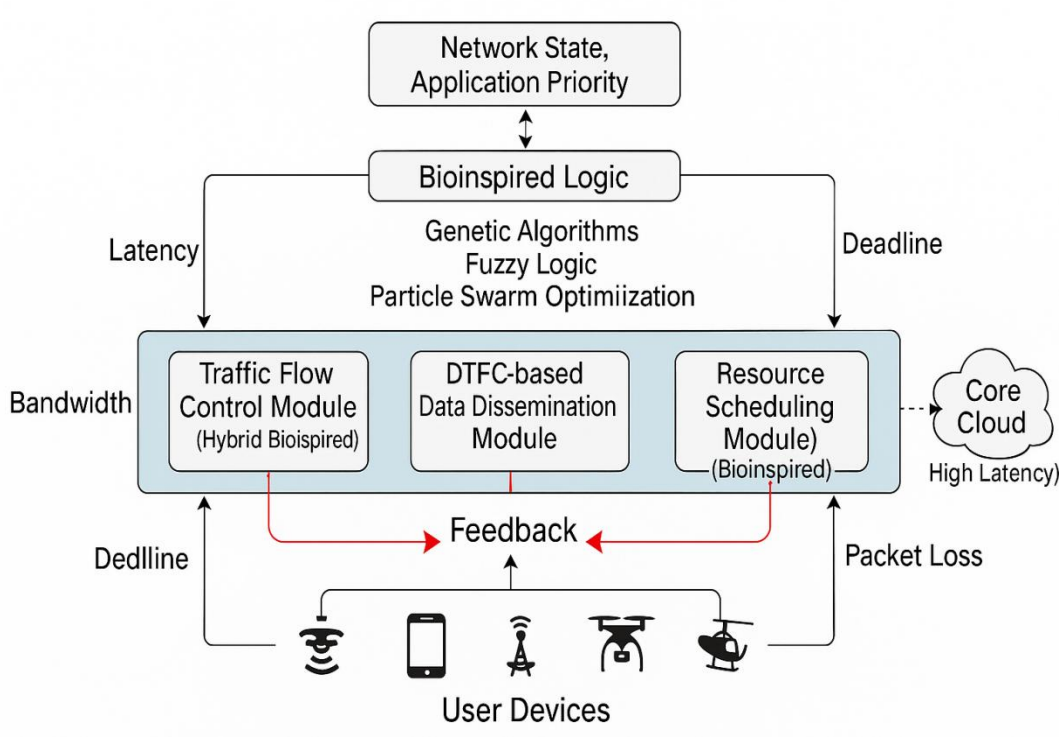


Figure1.1- QoS-aware adaptive framework in MEC

The figure 1.1 illustrates how intelligent, bioinspired decision-making can be integrated with real-time network operations in a Mobile Edge Computing (MEC)

environment by illustrating a QoS-aware adaptive framework. Network status and application priorities are continually tracked at the top and fed into a bioinspired logic layer, which uses particle swarm optimization, fuzzy logic, and evolutionary algorithms to adaptively control system behavior. Three essential MEC modules are powered by this intelligence: a Resource Scheduling Module that effectively distributes edge resources, a DTFC-based Data Dissemination Module that guarantees timely and QoS-compliant data delivery, and a Traffic Flow Control Module that dynamically regulates data traffic. These modules provide services and provide data for user devices including IoT sensors, smartphones, and UAVs. The two real-life applications where proposed MEC framework can significantly improve system performance are as follows:

Smart Traffic / Autonomous / Intelligent Transportation Systems (Autonomous Vehicles): The data-dissemination of your Dynamic Traffic Flow Control (DTFC) and QADE enhance the low-latency delivery, increased PDR and reduced congestion of data packets. Cars keep on producing traffic flow information (GPS, lane change intentions, sensor data). The hybrid EHPSO-based traffic flow will make sure that priority (and minimum delay) data (e.g. accident alert, pedestrian crossing alert) is sent, without losing packets in congestion. Performance improvement: reduced dissemination delay, increased percentage of packet delivery and minimized congestion (as indicated in your results, decreased latency, increased throughput and energy efficiency).

Smart Healthcare / Remote Patient Monitoring on the Edge: The VARMAx + FPO resource scheduling model is an edge-based allocation of resources of real-time workload prediction to enhance deadline compliance and energy efficiency. Smartwatches transmit real-time health information (ECG, SpO2, BP) to local MEC servers. In the event of patient data spikes (e.g. emergency), the scheduling model forecasts the load and redirects processing to available edge nodes rather than to cloud. Performance improvement throughput reduction, real-time decision making, and enhanced quality of service of time sensitive health information.

1.1 Background and Motivation

The purpose of this thesis work is to provide significant context for understanding the significance of traffic flow control and resource scheduling in MEC.

i) Background

The dissemination of smart devices, the Internet of Things (IoT), and data-driven applications that require ultra-low latency, high dependability, and real-time processing characterize the current digital era. Traditional centralized cloud infrastructures are under unprecedented strain as a result of these technological changes, and they are unable to keep up with the real-time demands of applications like augmented/virtual reality, industrial automation, autonomous driving, and healthcare monitoring. Mobile Edge Computing (MEC), a paradigm-shifting approach that brings cloud capabilities closer to end users and data sources at the network's edge, has arisen in response to these constraints. MEC greatly lowers transmission delays, eases core network congestion, and improves context-aware service delivery by decentralizing data processing and service provisioning [5]. The ecosystem is nevertheless dynamic and complex despite MEC's benefits because of its heterogeneous devices, dispersed architecture, and resource limitations. Maintaining Quality of Service (QoS) while functioning in the face of fluctuating network conditions and user demands is one of the main issues in MEC. Particularly in latency-sensitive and mission-critical applications, it is imperative to closely monitor and manage key QoS metrics including latency, bandwidth, jitter, packet loss, and throughput. These dynamic requirements are frequently outside the scope of traditional static mechanisms and heuristic-based resource management systems, which results in inefficient resource usage and deteriorated service quality. Adaptive data dissemination is a crucial topic that needs MEC's concentrated attention. The timely and dependable transmission of data to the appropriate services and users becomes crucial when data is created at the network edge. Variable network quality, varying workloads, and limited processing resources make this much more difficult.

Intelligent models with context-aware decision-making and real-time adaptation are crucial for overcoming these obstacles [6]. The foundation of this thesis work is the necessity for such models. The inherent complexity of MEC systems can be addressed with the use of bioinspired algorithms, which mimic natural processes like evolution, swarming, and fuzzy reasoning. They are ideal for tasks like resource allocation, traffic flow control, and QoS-aware data dissemination because of their adaptable and self-organizing nature. These algorithms can react to unanticipated circumstances, evolve optimal solutions in real-time, and balance several goals at once, such lowering latency while maximizing throughput and fairness.

ii) Motivation

The increasing need for an intelligent, flexible, and QoS-focused framework that can effectively distribute data and distribute resources in a mobile edge setting is what encouraged this thesis work. The integration of bioinspired intelligence into MEC systems to control adaptive traffic flow, QoS-aware data distribution, and dynamic resource scheduling is the main topic of this thesis work. In order to meet the needs of contemporary edge-based applications and overcome the drawbacks of current static models, a modular, scalable, and context-aware architecture must be created [7]. Moreover, there are currently no complete frameworks in the literature that integrate edge computing with bioinspired algorithms to handle the three interrelated domains of resource scheduling, data distribution, and traffic management—all while ensuring a constant quality of service guarantee. By putting forward a hybrid, multi-level architecture that incorporates intelligent decision-making into the very fabric of MEC operations, this thesis work aims to close this crucial gap. By doing this, the system may proactively adjust to shifting application priorities and network conditions, greatly enhancing user experience and performance. As companies continue to shift toward edge-enabled infrastructures, this thesis work is motivated by both theoretical curiosity and practical ramifications. Enabling scalable, dependable, and sustainable solutions for real-time applications in a variety of areas requires the development of sophisticated, adaptive mechanisms for MEC [8]. The research advances edge

computing and opens the door for future developments in adaptive networked systems by tackling these issues. There are many other reasons that motivated towards this dynamic research field as mentioned below:

a. Traditional Cloud-Based Architectures Drawbacks: Discussed how dispersed environment's real-time and latency-sensitive applications cannot be satisfied by centralized cloud solutions.

b. The Increasing Intricacy of MEC Resource Administration: Emphasize the difficulties in handling heterogeneous devices, dynamic resources, and changing network conditions at the edge.

c. The necessity of adaptable and QoS-aware data dissemination: In order to maintain QoS requirements in the MEC environment, stress the significance of real-time, context-aware data dissemination mechanisms.

d. Bioinspired Algorithms Potential in Changing Environments: Justify the use of bioinspired methods (such as GA, PSO, and fuzzy systems) to scheduling, resource allocation, and traffic flow to allow for intelligent and self-adaptive decision-making.

1.2 Problem Statement

The emergence of Mobile Edge Computing (MEC) has brought about a fundamental change in the way end users receive, distribute, and use data, especially in real-time and latency-sensitive applications. MEC increases responsiveness and decreases transmission latency by allowing computation at the network edge. However, dynamic workloads, constrained computational and bandwidth resources, heterogeneous devices, and quickly shifting network states are intrinsic characteristics of the MEC environment. Consistent Quality of Service (QoS) across all edge nodes and apps is becoming more and more challenging as a result of these issues. Efficient data distribution under changeable conditions is severely hampered by the need to guarantee on-time delivery, low packet loss, and service deadline observance. The majority of data distribution strategies currently in use are based on static or semi-

static methods that are unable to adjust to changes in network topology, user behavior, and resource availability in real time. Particularly in extremely dynamic and mission-critical applications like remote surgery, autonomous cars, and industrial automation, this frequently results in higher latency, network congestion, underutilization of resources, and inability to meet QoS standards. Furthermore, MEC's traffic flow control systems are continually developing. Adaptive prioritizing and contextual decision-making, which are crucial for handling fluctuating data loads and application needs, are frequently overlooked by traditional approaches. Data packets may be lost, delayed, or redundant in the absence of effective flow control, which would lower the overall quality of the service. Furthermore, because edge resources are scattered and constrained, scheduling them at the edge continues to be a major difficulty. Current scheduling methods frequently do not dynamically optimize resource allocation depending on network and user context, nor do they take into account real-time QoS limitations. This restriction lowers the quality of the user experience and leads to inefficient usage of resources. Despite the fact that bioinspired algorithms have shown great potential in optimization tasks, nothing is known about how to integrate them into MEC for scheduling, data distribution, and traffic management. Comprehensive frameworks that use hybrid bioinspired methodologies to address these three critical issues together while preserving end-to-end QoS compliance are scarce. The lack of an integrated, flexible, and QoS-aware data distribution framework in MEC that can optimize resource scheduling, intelligently control traffic flow, and dynamically adjust to changing network conditions through bioinspired intelligence is thus the main issue this thesis attempts to address. To fully utilize MEC and enable future-ready applications that require responsiveness and dependability, such a solution must be developed. In this regard, the thesis work pinpoints the following fundamental problems that obstruct efficient data distribution and resource optimization in MEC:

a. Static Methods Cannot Manage Real-Time Adaptation: Conventional approaches to resource scheduling and data distribution are not adaptable enough to handle abrupt shifts in user demand, mobility trends, or edge network congestion.

b. Inadequate QoS Awareness in Current MEC Models: Suboptimal service performance results from many current MEC frameworks ineffective incorporation of important QoS criteria into their decision-making processes, including latency, packet loss, and delivery deadlines.

c. Bioinspired Algorithms Are Underutilized in Integrated Optimization: Despite the success of bioinspired algorithms in discrete optimization problems, there aren't many integrated frameworks that use them for scheduling, data distribution, and traffic control in a single MEC environment.

1.3 Purpose of the Research work

The main goal of this thesis work is to introduce adaptive, QoS-driven solutions to Mobile Edge Computing (MEC) in order to overcome the shortcomings of the current static and non-intelligent processes. The following are the general research's purposes:

a. To create a hybrid bioinspired adaptive traffic flow control model that dynamically controls edge-layer data traffic according to application priority, bandwidth availability, and real-time congestion levels.
b. To use Dynamic Traffic Flow Control (DTFC) to create a QoS-aware data dissemination strategy that guarantees priority-based, dependable, and timely data delivery across MEC nodes in a range of network scenarios.
c. To include bioinspired optimization methods into MEC decision-making processes for improved performance and flexibility, such as Particle Swarm Optimization (PSO), Fuzzy Logic, and Genetic Algorithms (GA).
d. To incorporate crucial factors including latency, packet loss, deadline sensitivity, and bandwidth use into management, dissemination, and scheduling systems in order to guarantee end-to-end QoS compliance.
e. To provide a framework that is scalable and modular so that it may be readily expanded or changed to accommodate various edge-based applications with various QoS needs.

f. To provide a cohesive, intelligent MEC architecture that fills in the knowledge gaps in edge resource optimization, QoS-aware dissemination, and adaptive traffic management through a bioinspired methodology.

1.4 Significant Contribution

This thesis work adds a lot to the field of Mobile Edge Computing (MEC), especially when it comes to resource scheduling, traffic flow control, and QoS-aware data distribution. One of the main achievements is the creation of a hybrid bioinspired model for adaptive traffic flow control that combines evolutionary computing and fuzzy logic to intelligently govern traffic in real-time according to bandwidth availability, data priority, and network congestion. By improving MEC environments capacity to adjust to constantly fluctuating data loads, this paradigm lowers latency and prevents packet congestion at the edge layer. The suggested DTFC-based data dissemination approach, which integrates Quality of Service (QoS) metrics straight into the data forwarding and routing procedure, makes a second significant addition. By doing this, the model guarantees the prompt and dependable distribution of important data, particularly in situations with changing network performance or excessive demand. The suggested approach incorporates deadline sensitivity, packet loss tolerance, and service criticality into dissemination decisions, in contrast to traditional models that frequently overlook end-to-end QoS needs. The development of a bioinspired adaptive resource scheduling system, which makes use of methods like Genetic methods (GA) and Particle Swarm Optimization (PSO) to intelligently distribute edge resources, is another significant advance. The system's responsiveness under a variety of unpredictable operating conditions is greatly enhanced, which adjusts to real-time network feedback and maximizes resource usage and service fairness. Additionally, this thesis work offers a unified, modular framework that combines data distribution, resource scheduling, and traffic control into a scalable and coherent architecture. In addition to addressing individual issues, this comprehensive strategy guarantees inter-module cooperation for improved QoS management throughout the MEC ecosystem. Thorough simulations have been used to assess the models put forward in this study, and compared findings show

significant gains over baseline methods in terms of latency, throughput, resource efficiency, and QoS compliance. All things considered, the contributions presented in this thesis work offer a strong basis for the development of intelligent, scalable, and QoS-focused MEC systems appropriate for upcoming real-time applications.

1.5 Significant of Bioinspired Model in MEC

A strong strategy to deal with the growing complexity and dynamic nature of edge environments is the incorporation of bioinspired models into Mobile Edge Computing (MEC) systems. Because of varying user needs, erratic network conditions, and heterogeneous devices, MEC is intrinsically distributed, resource-constrained, and extremely variable. Conventional rule-based or static optimization approaches frequently fall short in such a situation in terms of providing the required responsiveness and flexibility. Because of their durable, adaptive, and self-organizing properties, bioinspired models—which draw inspiration from natural systems and evolutionary principles—offer a possible substitute. Methods that can continually evolve optimal or near-optimal solutions under changing conditions, like fuzzy logic, genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO), are ideal for the MEC paradigm. The capacity to carry out multi-objective optimization, balancing trade-offs among conflicting QoS needs like latency, packet loss, deadline adherence, and resource consumption, is one of the main advantages of bioinspired techniques in MEC. Even in the face of erratic workloads and resource variations, these models may guarantee optimal performance by dynamically modifying operating parameters and reacting to real-time environmental feedback [9]. For example, during network congestion, a resource scheduling method may load-balance and QoS-compliantly divide computational workloads across edge servers, while a bioinspired traffic flow management mechanism can prioritize delay-sensitive packets. Additionally, decentralized decision-making is supported by bioinspired models, which fits in nicely with the MEC design, since scattered edge nodes make centralized control impracticable. Their usefulness in extensive MEC installations is further increased by their scalability, adaptability, and resistance to local failures. Crucially, these models are perfect for new

real-time applications with unpredictable behaviors because they don't require a lot of pre-configuration or static assumptions. This thesis work makes use of MEC's full potential to provide flexible, effective, and context-aware services by integrating bioinspired intelligence into scheduling, data distribution, and traffic control procedures [10]. The importance of bioinspired models in augmenting MEC's capabilities is becoming more and more apparent as it develops further as the foundation of next-generation computing and is essential for designing sustainable systems.

1.6 Research work Objectives

Following four objectives have been finalized in line with the research work:

I. **To study and analyze the existing resource allocation and network management techniques for Edge Computing.**
   This objective involves reviewing current methodologies to identify existing resource handling and traffic scheme with in MEC environments.

II. **To design a framework for adaptive network traffic flow control in Edge computing for diversified applications.**
   This focuses on developing a dynamic model that regulates data flow based on varying application needs and real-time network conditions.

III. **To propose a technique for QoS-aware resource allocation in a Mobile edge computing environment.**
   This aims to create a strategy that allocates resources efficiently while maintaining critical QoS parameters like latency and packet loss.

IV. **To implement and validate the proposed work in the simulation environment.**
   This involves comparing the models in a simulation setup and evaluating their performance against existing methods by focusing on Quality of Service (QoS) metrics.

1.7 Research work Organization

The purpose of this thesis is to present a comprehensive and lucid analysis of the research on QoS-aware adaptive data dissemination in the mobile edge computing ecosystem. This study work's structure makes sense and enables readers to delve deeply and clearly into the topic. Every chapter contributes to the general comprehension of the study project and builds on the ones that came before it, ultimately bringing a collection of conclusions, conclusions, and recommendations for the future.

Chapter 1: Introduction

The first chapter of the thesis is titled "Introduction." The introductory elements of the research project are established in this first chapter. The first section, "1.1 Background and Motivation," sets the scene for the investigation by examining the importance of MEC and the driving forces for this research project. In order to set the scenario for the ensuing chapters, the "1.2 Problem Statement" that follows describes the difficulties and constraints encountered in the jurisdiction of MEC. "1.3 Purpose of research work" provides a roadmap for what the reader might anticipate learning by outlining the precise aims and objectives of the research project. "1.4 Significant Contribution" describes how research has benefited society. The significance of these models in MEC to improve efficiency is described in "1.5 Significant of Bioinspired model in MEC." A thorough explanation of the research work objectives and the need to fulfill them in mobile edge computing can be found in "1.6 Research work Objectives." Lastly, "1.8 Research work Organization" walks the reader through the following chapters by giving a summary of the content and organization of the complete thesis work.

Chapter 2: Literature Review

The "Literature Review," included in Chapter 2, provides the study work's intellectual underpinning. It is divided into three parts, each with a specific emphasis. With some of the most recent and ongoing author and scholar research, "2.1 Historical Evolution of Adaptive Data Dissemination in MEC" offers a thorough grasp of MEC networks and

their numerous uses. In addition to some recent work, "2.2 Related work" offers an overview of some previous research on a variety of applications. " 2.3 "Research Work Question" gives us a thorough understanding of the significance of this field's study as well as the outcomes we can anticipate from its application. In order to prepare the reader for the creative solutions offered in the following chapters, "2.4 Literature Summary" summarizes the main points of the most recent and current research.

Chapter 3: Hybrid Bioinspired Model for Adaptive Traffic Flow Control

The first MEC models are shown in Chapter 3 and are called "BATFE." Each of the five components that make up this chapter adds to a thorough comprehension of the concept. " 3.1 Introduction to BATFE" lays the groundwork by outlining the fundamental ideas of the model. The main principles of the proposed algorithm are explained in "3.2 Algorithm Overview." 3.3 Important Parameter and Variable in the Model "explains the performance parameter to be used in the model." The procedures that must be performed in order to integrate the bioinspired model with the MEC framework are described in "3.4 Design of the hybrid bioinspired model." 3.5 The analysis of results "showcases the performance of model used in research work by comparing with existing model." 3.6 Conclusion and Future Scope" outlines the advantages and disadvantages of the implemented approach.

Chapter 4: QoS-AWARE DATA DISSEMINATION WITH DTFC IN MEC

The investigation of new MEC models is continued in Chapter 4 with "DTFC." This chapter, like the one before it, is divided into eight sections, each of which adds to a thorough comprehension of the model. " The main ideas and goals of the model are presented in "4.1 Introduction." Within the MEC framework, "4.2 Design of the model" outlines the procedures that must be followed for learning. "4.3 Adaptability analysis" describes the fundamental model analysis in terms of adaptability. By contrasting it with an existing model, "4.4 Result Analysis" illustrates how well the model employed in the study activity performs. "4.6 Potential Limitation" describes the limitations of the implemented model, whereas "4.5 Node and resource variability characteristics" concentrates on the dynamic nature of the node. " EHPSO insights are provided in "4.7

Path selection with EHPSO." "4.8 Conclusion and Future Scope" offers a detailed synopsis of the research findings and their potential for further development.

Chapter 5: Bioinspired Adaptive Resource Scheduling in MEC

The investigation of MEC models for adaptive resource scheduling is continued in Chapter 5. This chapter, like the one before it, is divided into eight sections, each of which adds to a thorough comprehension of the model. " 5.1 Introduction" presents the main ideas and goals of the paradigm. 5.2 Goal and Motivation" offers inspiration for carrying out the study. The advantages of the implemented model are explained in "5.3 Application." 5.4 "Novelty of the model" offers information on how new work is applied. The procedures that must be taken for learning within the MEC framework are explained in "5.5 Design of the model." 5.6 "Model Explanation" describes the fundamental structure of the model in relation to scheduling flexibility. By contrasting it with an existing model, "5.7 Result Analysis" illustrates how well the model employed in the research project performs. 5.8 "Conclusion" offers a detailed synopsis of the research findings and their potential for further development.

Chapter 6: Conclusion and Future Work

The study project is concluded in Chapter 6, the last chapter. It is divided into six parts. "6.1 Performance of BATFE" provides a summary of the BATFE model's performance in research projects. 6.2 DTFC Performance" provides an overview of the DTFC model's performance in research projects. 6.3 VARMAx Performance" provides an overview of the VARMAx model's performance in research projects. The practical consequences of the findings are discussed in "6.4 Inferences of the Research Work." 6.5 Future Scope" lists prospective avenues for further investigation. 6.6 "Summary of Findings" provides concluding thoughts on the research process.

CHAPTER 2

LITRATURE REVIEW

The literature review chapter analyzes previous studies and solutions in the area of QoS-aware adaptive data dissemination in mobile edge computing ecosystems. By summarizing important discoveries, knowledge gaps, and the development in this field, this chapter seeks to identify the limitations of current solutions and scope of further enhancement in them.

2.1 Historical Evolution of Adaptive Data Dissemination in MEC

Over time, the idea of data distribution has changed dramatically, moving from conventional centralized designs to more intelligent and decentralized methods. At first, data distribution was based on static, cloud-based models in which data processing and storage took place in centralized data centers. These models were not appropriate for real-time applications due to their high latency and network congestion. By putting computing and storage closer to end users, Mobile Edge Computing (MEC) reduced delays and increased efficiency, signaling a standard shift. However, the static data distribution methods used in early MEC implementations were unable to adjust to changing network conditions, which resulted in inefficient use of resources. As wireless communication technologies like 4G LTE and 5G advanced quickly, adaptive data transmission strategies began to attract interest. By taking into account variables including user mobility, network load, and service demand, researchers developed heuristic-based techniques to optimize data distribution. Predictive analytics and edge caching were essential elements that enabled MEC nodes to retain frequently requested content and foresee future requests. By facilitating context-aware material delivery and real-time decision-making, the combination of artificial intelligence and machine learning significantly improved adaptive dissemination. In MEC contexts, these developments greatly enhanced Quality of Service (QoS) and decreased duplicate data transmissions [11]. In order to maximize adaptive data transmission in MEC, bio-inspired and

reinforcement learning algorithms have been investigated recently. Swarm intelligence-inspired methods, like particle swarm optimization and ant colony optimization, have been used to distribute data effectively while reducing latency and energy usage. MEC systems may now learn from user behavior and network conditions. Reinforcement learning models, which allow them to dynamically modify their distribution techniques for best results. Furthermore, blockchain-based data distribution has become popular since it provides safe, decentralized ways to improve edge network dependability and trust. These developments have helped to increase the scalability and efficiency of MEC-based adaptive data dissemination.

In the future, the incorporation of next-generation technologies like 6G, federated learning, and edge intelligence is anticipated to propel the development of adaptive data distribution in MEC. More advanced data dissemination strategies will be required to meet the increasing demand for ultra-low latency applications, such as extended reality (XR) and driverless cars. Future methods will probably concentrate on collaborative edge networks, in which several MEC nodes cooperate in real time to maximize data transmission. The next stage of adaptive data dissemination in MEC will open the door for edge computing ecosystems that are more intelligent, safe, and robust by utilizing developments in AI, blockchain, and quantum computing [12]. Adaptive data distribution in MEC is changing toward increasingly independent and self-optimizing systems in tandem with the growing demand for real-time and mission-critical applications. To improve resilience and adaptability, MEC setups are incorporating emerging concepts like network function virtualization (NFV), software-defined networking (SDN), and digital twins. Digital twins reduce errors and increase efficiency by enabling real-time simulation and optimization of data transmission schemes prior to actual implementation. In a similar vein, SDN and NFV offer virtualization and centralized control, allowing MEC networks to scale smoothly and allocate resources dynamically. Next-generation adaptive data dissemination frameworks that can react to changing network circumstances and user demands intelligently with little assistance from humans are becoming possible because to these developments [13]. A thorough summary of the

bioinspired models currently in use for adaptive data distribution in MEC is provided in Table 2.1. It emphasizes their uses, significant contributions, and related benefits and drawbacks. Researchers can find appropriate optimization strategies to improve the effectiveness of data dissemination in MEC contexts by examining these models.

Table 2.1- Existing Bioinspired Model for Adaptive Data Dissemination in MEC

Model Name	**Applications**	**Key Contribution**
Ant Colony Optimization (ACO)	Optimizing data routing and dissemination in MEC networks	Efficient path discovery and adaptive data routing
Particle Swarm Optimization (PSO)	Resource allocation and load balancing in MEC	Fast convergence and flexibility in network optimization
Genetic Algorithm (GA)	Optimized task offloading and edge caching	Effective in solving multi-objective optimization problems
Artificial Bee Colony (ABC)	Adaptive data dissemination and energy-efficient MEC	Self-organizing and robust for dynamic network conditions
Firefly Algorithm (FA)	Data clustering and dynamic network optimization in MEC	Effective in handling non-linear optimization and adaptive clustering

Bat Algorithm (BA)	Optimization of MEC network parameters and task scheduling	Balances exploration and exploitation for robust optimization
Grey Wolf Optimizer (GWO)	Resource allocation and traffic management in MEC	Mimics hierarchical decision-making for enhanced efficiency
Cuckoo Search Algorithm (CSA)	Dynamic data dissemination and energy efficiency in MEC	Adaptive exploration mechanism for robust global optimization
Whale Optimization Algorithm (WOA)	Optimizing network load balancing and resource scheduling	Dynamic and adaptive search mechanism for better load balancing
Dragonfly Algorithm (DA)	Adaptive routing and self-organized network optimization	Inspired by swarm intelligence for adaptive and scalable solutions

By using hybrid methodologies and sophisticated optimization techniques, the problems with bioinspired models in adaptive data distribution for MEC—such as high computational complexity, local optima trapping, sluggish convergence, and sensitivity to parameter tuning—are being actively addressed. Integrating deep reinforcement learning (DRL) with bioinspired models is one exciting avenue that could enable real-time adaptability to dynamic MEC settings and intelligent decision-making [14]. Furthermore,

sensitivity problems are addressed and convergence speed is increased using parameter self-tuning techniques like adaptive learning rates and swarm intelligence-based fine-tuning. The efficiency of these models is also being improved by the use of edge AI and quantum computing, which lower computational overhead and allow for real-time optimization. Additionally, blockchain technology is being investigated to offer transparent, safe, and decentralized data distribution.

2.2 Related Work

[1] The authors explained the definition of edge computing which addresses the concern of response time where concerns are latency, resource like battery-life constraint, bandwidth cost-saving, as well as data safety and privacy. [2] summarized the existing edge computing systems and related tools. The authors divided the paper into two parts: System View and Application View. In system View, Open-Source Edge computing projects and edge computing systems & tools are discussed wherein application view, deep learning optimization at the edge are discussed. [3] presented the major three edge computing technologies: mobile edge computing, cloudlets, and fog computing. The authors explained application areas, architectures, standardization efforts for mobile edge computing, cloudlets, and fog computing. [4] the authors described the Edge computing that processes the gathered data from end devices at the edge of the network. By covering a large range of technologies, edge computing addresses the various concern as battery life constraint, bandwidth usage, latency, data security, and data privacy. The need for edge computing (Push from Cloud Services and Pull from the Internet of Things) is discussed by [5, 6] in which the auto-scaling applications in edge computing which maintained the online services at a decentralized location. They broadly explained two aspects of this paper. In the first section, they major focused on the different types of edge computing applications (IoT Applications, Micro-service applications, Time-critical applications). For these applications, auto-scaling challenges when the workload dynamically changes. Container-based visualization auto-scaling technologies are discussed. Self-adaptive application at runtime enhances the performance. In the classification of auto-scaling applications in edge computing, the authors explained the

cloud framework, Virtualization Technology, Monitoring approach, Operational behavior, Adjustment ability, Architectural support, Image Delivery, and scalability techniques [7]. [8] addressed the issue of low latency requirements in mobile edge computing. The author proposed a fast data-sharing framework HDS (Hybrid data sharing) to meet the requirements of low latency by dividing the gathered data location service into two regions: Intra-region and inter-region. With the Hybrid information sharing system which comprises 100 areas, the creator accomplished low latency, low usage overhead, and 50.21% more limited query ways, and 92.75% fewer false positives. In this whole network, the total edge server used was 1000 to 10000 [9]. [10] the authors performed the cloud edge latency comparison. They performed an extensive measurement to assess the latency characteristics of end-users to the edge servers and cloud data centers. It estimated latency from 8,456 end-clients to 6,341 Akamai edge workers and 69 cloud areas. At last, the paper's outcome is that while 58% of end-clients can arrive at a close by edge worker in under 10 ms, just 29% of end-clients get a comparable dormancy from a close by cloud area. [18]. [19] the authors reviewed the various data latency techniques in Mobile Edge Computing. As in the centralized cloud, one cannot achieve low latency. Mobile Edge Computing makes efficient use of the resources and decreased the movement of large data generated by the edge devices. Edge computing is important for solving fatal situations such as Conflicts in Autonomous vehicles, Fire, Environmental Hazards.

In their presentation of a deep learning-based traffic flow detection strategy for intelligent traffic systems, [46] emphasized the value of edge computing for organizing and processing the vast amounts of data produced by contemporary transportation systems. [50] further elaborates on this topic of using deep learning for traffic flow prediction and shows how effective it is in a vehicular Internet of Things environment. By focusing on shared resource allocation based on traffic flow virtualization and online traffic flow prediction for autonomous vehicles and connected cars, respectively, [47] and [48] made important contributions to this field. [49] on the use of block chain technology for IIoT traffic management highlights the growing demand for secure and effective data

processing in edge computing environments. Further evidence of this point is Shin and Kim's multi-layered security framework for cloud-native edge clusters [51].

Contributions by [82], [83] and [84] address intelligent traffic-adaptive resource allocation, QoE-aware traffic aggregation, and robust feature selection, respectively, demonstrating the importance of edge computing in improving network intelligence. By investigating modal shifts with mobility in mind and elucidating the flow between origin and destination, [85] and [86] contributed to this field and demonstrated the dynamic nature of traffic management in edge computing situations. Other contributions in this field include the creation of his framework for fog-based traffic flows. [88] extraction of mixed road user trajectories by [89] and dynamic optimization of traffic flow prediction models by [87]. Discussion by [90] on the use of federated deep reinforcement learning for traffic monitoring provides a new method for traffic control in his SDN-based IoT networks. The increasing use of advanced computational techniques in traffic management is evidenced by the reconstruction of traffic data of large-scale IoV systems using neural network approaches, as reported by [91] and the development of cooperative and energy-efficient strategies in emergency navigation by [92]. A study by [93] on flow allocation and processing on a distributed edge computing platform, [94] on an intelligent traffic light system based on block chain technology represents technological progress in this field. The literature, in summary, shows a notable trend toward the effective control and prediction of traffic flow in ITS and IoT systems through the use of edge computing, deep learning, and block chain technologies. The aforementioned studies underscore the significance of advanced computational techniques and resilient security frameworks in managing the intricacies of contemporary transportation systems and network traffic [95].

Flooding-based dissemination, in which data packets are sent to all network nodes, is a prevalent method. While flooding ensures extensive coverage, it often results in redundant transmissions, excessive energy consumption, and network congestion. Diverse optimization strategies have been proposed as solutions for these issues [96]. The Gradient-based Routing (GR) algorithm, for instance, gives nodes closer to the sink a higher priority, thereby reducing the number of redundant transmissions [97, 98].

However, GR does not consider temporal factors, which can result in sub-optimal routing decisions in dynamic MEC environments. Information is also disseminated via a random peer-to-peer process through a gossip-based dissemination method. By leveraging the mobility of nodes, gossip protocols like Epidemic and Spray-and-Wait achieve high coverage and robustness [99]. However, these protocols have a significant delay and may not guarantee the delivery of data reliably. Content-based routing has gained popularity as an efficient data distribution method in MEC. By analyzing the contents of data packets, routing decisions can be made based on the packets' proximity to their final destinations and their relative importance. Content-based routing reduces unnecessary transmissions, conserves energy, and increases the effectiveness of routing. Examples include COIN, SPIN, and Directed Diffusion. However, the majority of existing content-based routing protocols do not account for temporal factors such as delay, energy consumption, throughput, and Packet Delivery Ratio (PDR), limiting their efficacy in dynamic MEC environments [100, 101]. Effective traffic flow control is necessary for optimizing resource utilization and ensuring QoS guarantees in MEC deployments. In numerous ways, existing models and algorithms address these issues. Traditional traffic flow control mechanisms, such as static routing and load balancing, have limitations in dynamic MEC environments. These mechanisms frequently utilize static configurations and do not adapt to changing network conditions, resulting in sub-optimal resource allocation and utilization via the Main Task Off-loading Scheduling Algorithm (MTOSA) process [102, 103]. In addition, traditional load balancing techniques do not take the processing power of edge devices into account, which is essential for effective traffic flow management [104, 105]. Particle Swarm Optimization (PSO) is widely employed in MEC for dynamic traffic flow management. PSO is a metaheuristic optimization algorithm inspired by the behavior of social organisms such as flocks of birds and schools of fish [106]. PSO has been expanded to address traffic flow control issues by adjusting routing decisions dynamically based on the capacity of edge devices. EHPSO (Elephant Herding Particle Swarm Optimization) uses PSO to balance network load by considering the processing capabilities of edge devices [107]. EHPSO dynamically routes traffic to

nodes with available processing capacity, reducing congestion and optimizing resource utilization via Hierarchical Federated Learning (HFL) process [108,109]. Existing models [110, 111] for adaptive data distribution and dynamic traffic flow management in MEC have made significant contributions. Temporal aspects such as delay, energy consumption, throughput, and PDR must be considered to optimize routing decisions and traffic flow control in dynamic MEC environments. In this regard, however, the majority of these models have limitations. Moreover, traditional routing and traffic control mechanisms frequently lack the adaptability to adapt to changing network conditions and fail to utilize the processing power of edge devices. These limitations necessitate the development of novel approaches, such as the proposed QoS-aware Adaptive Data Dissemination Engine with Dynamic Traffic Flow Control, which integrates content-based routing and EHPSO to overcome these obstacles and enhance MEC deployment performance levels [112].

Table 2.2- Summarization Table of existing method used in data dissemination

Method	**Description**	**Advantage**	**Challenges**
Flooding-based Dissemination	Data packets are sent to all network nodes. Extensive coverage leads to redundancy, energy consumption, and congestion.	Wide coverage Simplicity	Redundant transmissions Energy consumption Network congestion
Gradient-based Routing (GR)	Nodes closer to the sink get higher priority, reducing redundancy.	Reduces redundancy	Sub-optimal routing decisions in dynamic MEC environments
Particle Swarm Optimization (PSO)	Optimization algorithm for traffic flow. Dynamically adjusts routing based on edge device capacity.	Dynamically adjusts routing decisions Balances network load	Need to consider processing power of edge devices Implementation complexity

The effective distribution of computational tasks across edge devices to satisfy quality of service (QoS) requirements and maximize resource utilization is a challenging task in Mobile Edge Computing (MEC) environments. To address this issue, a number of models

and algorithms have been put forth, but each has pros and cons depending on the particular requirements of the MEC scenarios [113, 114]. This is done via use of Dueling Double Deep Recurrent Q Network (D3RQN) process. The traditional First Come First Serve (FCFS), Round Robin (RR), and Shortest Job First (SJF) scheduling algorithms were used in one of the earliest methods. They served as a starting point for task scheduling in MEC, but because of their inherent simplicity, they failed to take into account dynamic shifts in resource availability and demand, which resulted in subpar performance in demanding real-time applications [115- 118]. The most effective scheduling policies have been discovered over time by using Q-learning and other reinforcement learning-based models. These models have the ability to change with their surroundings and online learn the best course of action. These algorithms, however, frequently need extensive training, and they might not be able to adjust quickly enough to the rapidly altering network conditions [119-122]. Additionally, some researchers have suggested using models based on game theory, mainly focusing on fostering competition among the edge devices for effective resource allocations. While these models are capable of reaching a Nash equilibrium, which offers a stable state for the system, they frequently fail to provide acceptable QoS, especially in highly dynamic scenarios [123-126]. [127] systematic literature review explored the concept of Quality of Service (QoS) monitoring in IoT edge devices driven healthcare. The study focuses on the individual devices present at different levels of the smart healthcare infrastructure and the QoS requirements of the healthcare system as a whole. The authors propose a novel pre-SLR method for comprehensive keyword research on subject-related themes for mining relevant research papers for quality SLR; a review of several QoS techniques used in current smart healthcare apps; an examination of the most important QoS measures in contemporary smart healthcare apps; and offering solutions to the problems encountered in delivering QoS in smart healthcare IoT applications to improve healthcare services. The authors propose that edge computing and artificial intelligence can resolve these issues by processing data in edge devices located at the brink of the network, contributing to less latency and energy efficiency. This enables edge-assisted IoT systems to deliver

medical services on time. AI techniques, such as machine learning and deep learning, are widely used for system training and learning in edge computing. [128, 129] explored the use of optimizable tree machine learning (ML) algorithms to evaluate spectrum sensing in CR-based smart healthcare systems. The researchers used data sets based on the probability of detection and false alarm to train and test the system using various TBAs. The results showed that the optimizable tree provided the best accuracy results for spectrum sensing evaluation with minimum classification error (MCE). This approach is particularly useful for smart healthcare systems that use cognitive radio (CR) to send and receive patient health data. The study highlights the importance of utilizing ML in the field of smart healthcare. CR technology can provide maximum advantages of smart medicine to patients at their doorstep by exploiting AI techniques to process patient health data on a micro level, even at the patient's genetic level. Monitoring wireless sensors attached to the human body monitor body parts and collect real-time data, sharing collected data with a remotely placed fusion center or data server. [130, 131] discussed about efficient resource prediction framework (ERPF) is proposed to provide proactive knowledge about radio resource availability in software-defined heterogeneous radio environmental infrastructures (SD-HREIs). The framework measures radio activity in unlicensed bands, segregates it into signal and noise, and uses machine learning techniques to predict radio occupancy and opportunity. Next-generation heterogeneous radio environmental infrastructures aim to enhance spectral efficiency, reliability, and control while supporting high data rates and diverse services. However, connecting devices to these infrastructures can be challenging. An efficient resource prediction framework (ERPF) can exploit radio resources according to user requirements, enabling dynamic spectrum access in SDH-REIs. Task scheduling in MEC has been suggested using deep learning-based models, particularly those that use recurrent neural networks (RNNs) and long short-term memory (LSTM) networks [132-135]. They have demonstrated significant promise in anticipating and adjusting to MEC scenarios that change quickly. These models demand a lot of computational power and time to train, which may not always be possible for edge computing devices with constrained resources

[136-139]. Despite these efforts, none of the models in use currently satisfactorily account for all the complexity and difficulties that MEC environments present. As a result, there is a gap in the market for a novel, effective, and adaptive task scheduling model that can accommodate the dynamic MEC scenarios while guaranteeing optimal resource utilization and satisfactory QoS. The VARMAx-based bioinspired resource scheduling model in this thesis work aims to fill this gap for real-time scenarios [140, 141]. Mobile Edge Computing (MEC) is an emerging concept that moves compute and storage resources closer to the network edge, enabling quicker data processing and lower latency for real-time applications. Recent studies underline the important importance of MEC in enabling the exponential production of IoT devices and the rising need for low-latency services. For instance, a detailed evaluation demonstrates the benefits of MEC in lowering end-to-end latency and boosting user experience by processing data at the network edge rather than depending on distant cloud servers [142]. Another research analyzes the integration of 5G with MEC, pointing out that the combination of both technologies can greatly enhance the performance of mobile networks by shifting computationally expensive jobs to edge servers, thereby lowering network congestion and enhancing service delivery [143-145]. Efficient task scheduling is crucial for maximizing resource usage and assuring Quality of Service (QoS) in MEC contexts. Traditional scheduling algorithms frequently struggle to fulfill the dynamic and diversified requirements of MEC applications. Recent research has focused on generating more adaptable and intelligent scheduling algorithms. For example, multi-objective deep reinforcement learning strategy for MEC, which simultaneously optimizes numerous QoS metrics like as latency and energy usage. This strategy harnesses the Pareto front to determine optimum trade-offs between conflicting objectives, providing considerable increases in scheduling efficiency [146, 147]. Bioinspired optimization techniques have showed tremendous promise in tackling the complicated job scheduling challenges in MEC. These algorithms mimic natural processes to identify optimal solutions in extremely dynamic and multi-dimensional problem domains. Recent research has studied several bioinspired strategies, including Flower Pollination Optimization (FPO), Genetic

Algorithms (GA), and Particle Swarm Optimization (PSO), to optimize resource allocation in MEC. For instance, a recent work proposes a VARMAx-based bioinspired resource scheduling model that combines the predictive powers of the VARMAx model with the flexibility of FPO. This hybrid technique provides more accurate task mapping to Virtual Machines (VMs) by incorporating numerous task and resource characteristics, leading to considerable increases in make span, deadline hit percentage, energy efficiency, and throughput [148]. This study seeks to design a multi-objective optimization technique optimized for job offloading in mobile edge computing (MEC) scenarios. The major purpose is to research and increase MEC system performance with reference to workload offloading.

Initially, a multi-objective task offloading scenario within MEC is built. A MEC task offloading scheduling technique based on multi-objective optimization is described, concentrating on improving both latency and energy usage during the computational offloading process [149-151]. Table 2.1 summarizing some existing resource scheduling models for Mobile Edge Computing (MEC). A summary of the current approaches used in data dissemination is given in Table 2.2, together with an explanation of their fundamental ideas and methods of execution. Understanding the development of various strategies and their efficacy in MEC-based adaptive data dissemination is made easier by this comparison. The benefits and drawbacks of a few current strategies utilized in data distribution are highlighted in Table 2.3. Reader can find areas for improvement and investigate hybrid or enhanced approaches to address present issues in MEC contexts by assessing their advantages and disadvantages.

Table 2.3 Advantages and limitations of some existing models

Model	Advantages	Limitations
First Come First Serve (FCFS)	Simple and easy to implement.	Does not consider task priority or resource requirements, leading to potential inefficiencies.

Round Robin (RR)	Fairly distributes tasks among resources, preventing any single resource from becoming overloaded.	Ignores task complexity and resource heterogeneity, which can lead to suboptimal performance.
Genetic Algorithm (GA)	Can find near-optimal solutions for complex scheduling problems through evolutionary techniques.	Computationally intensive and may require significant time to converge to a solution.
Particle Swarm Optimization (PSO)	Efficient in exploring large solution spaces and can adapt to dynamic changes in the environment.	May suffer from premature convergence and require fine-tuning of parameters.
Ant Colony Optimization (ACO)	Effective in finding optimal paths and resource allocations based on pheromone trails.	Performance can be heavily affected by the number of iterations and pheromone evaporation rate.
Multi-Objective Evolutionary Algorithm (MOEA)	Simultaneously optimizes multiple objectives, such as delay and energy consumption.	Computationally expensive and may require balancing trade-offs between conflicting objectives.
Deep Reinforcement Learning (DRL)	Learns and adapts to dynamic environments, potentially finding highly efficient scheduling policies.	Requires large amounts of training data and computational resources; may struggle with real-time constraints.

2.3 Research Questions

Q.1 How can bioinspired optimization techniques enhance adaptive data dissemination in MEC?

Response: Bioinspired optimization methods such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are used to improve adaptive data dissemination by optimizing routing, load balancing and resource allocation. These models replicate natural behaviors in a bid to enhance energy efficiency, reduction in latency as well as dynamically adjusting to network conditions. The ability of the MEC

systems to adapt to the environmental conditions in order to learn better provides better QoS-aware data dissemination.

Q.2 What are the key limitations of existing data dissemination techniques in MEC, and how can they be addressed?

Response: The existing techniques, including gradient-based routing and flooding-based dissemination, are associated with high-energy usage, high redundancy, and network congestion. Overcoming these challenges can be achieved by using hybrid approaches that integrate machine learning, predictive analytics and bioinspired optimization tools. Moreover, edge intelligence and software-defined networking (SDN) may enhance scalability and flexibility, whereas blockchain-based secure data dissemination may enhance reliability.

Q.3 What role does reinforcement learning play in improving adaptive data dissemination in MEC?

Response: Reinforcement Learning (RL) models can help MEC systems change data dissemination strategies in real time, using their user behavior and network conditions as inputs to the model to learn dynamically. It is possible to take adaptive dissemination further to reduce latency, enhance network performance, and resource consumption through integrating RL with bioinspired models. To illustrate, DRL-based techniques have the ability to optimize the cache placements at edge nodes, as well as predict future demands.

Q.4 What are the trade-offs between computational complexity and optimization performance in bioinspired models for MEC?

Response: Although bioinspired models such as ACO and PSO provide effective load balancing and path optimization, they frequently have slow convergence and significant computing complexity. By increasing convergence speed while preserving optimization efficiency, hybrid strategies—like combining PSO with edge AI or reinforcement

learning—help achieve equilibrium. To increase efficiency, quantum-inspired optimization methods and parameter self-tuning are increasingly becoming popular.

Q.5 How can bioinspired approaches be integrated with real-time adaptive traffic flow control for MEC-based applications?

Response: Dynamically changing routing based on the demand and network state can be used to combine traffic flow control with bioinspired models. As an example, PSO and EHPSO-based load balancing strategies could be useful to optimize resource allocation and network congestion. Moreover, within the framework of intelligent transportation systems and autonomous vehicle networks, bioinspired models can be enhanced with federated deep learning to enhance the quality of routing and prediction of traffic.

Q.6 How can hybrid bioinspired approaches improve the efficiency and scalability of adaptive data dissemination in MEC?

Response: Hybrid bioinspired systems also enhance efficiency and scalability through the combination of machine learning, reinforcement learning and deep learning with more traditional bioinspired algorithms, such as ACO, PSO, and GA. As an illustration, bioinspired models coupled with Deep Reinforcement Learning (DRL) make it possible to combine the two to create real-time learning and dynamics adaption of networks.

2.4 Literature Summary

Adaptive data dissemination in Mobile Edge Computing (MEC) has changed dramatically from traditional cloud-based models to decentralized and intelligent alternatives, as this chapter is examination of several extant methodologies makes clear. The use of centralized designs for early data distribution resulted in significant latency, network congestion, and wasteful resource use. By putting computing and storage closer to end users, MEC became a viable solution that enhanced network performance and decreased latency. However, the static data broadcast techniques utilized in the early MEC implementations were unable to dynamically adjust to shifting network conditions. To overcome this difficulty, researchers investigated edge caching, predictive analytics,

and heuristic-based methods to maximize data transfer. These developments were essential in improving Quality of Service (QoS) and cutting down on unnecessary data transfers. The application of bioinspired optimization approaches to improve adaptive data distribution in MEC is also covered in great detail in the literature. Data routing, resource allocation, and job offloading have all been shown to be efficiently optimized by algorithms like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Artificial Bee Colony (ABC). These bioinspired models minimize latency and energy consumption by dynamically modifying dissemination tactics based on network conditions by utilizing swarm intelligence and evolutionary concepts. Even with these developments, there are still issues with current models, such as their high computational complexity. According to the literature, next-generation technologies like 6G, federated learning, and edge intelligence will propel future developments in adaptive data dissemination in MEC. It is anticipated that collaborative edge networks, in which several MEC nodes work together in real time, will improve the effectiveness of data distribution for ultra-low latency applications, such as extended reality (XR) and driverless cars. Network scalability and dynamic resource allocation will be further made possible by the combination of network function virtualization (NFV) and software-defined networking (SDN).

CHAPTER 3

HYBRID BIOINSPIRED MODEL FOR ADAPTIVE TRAFFIC FLOW CONTROL

Effective traffic flow management requires monitoring edge devices to ensure traffic is evenly dispersed across networks. However, existing flow control systems, sometimes incorporating machine learning, struggle with complicated configurations and inefficiencies, particularly in large-scale device networks. This thesis work provides a hybrid bioinspired system to optimize traffic flow control in edge device networks and address these issues. The implemented methodology utilizes request-response time data to forecast traffic flows across multiple device sets. Using this predictive capacity, edge resources are dynamically assigned, considerably enhancing Quality of Service (QoS) in large-scale systems. The model analyzes this data using a hybrid Elephant Herding Particle Swarm Optimizer (EHPSO), which assigns temporal weights to IP groups to estimate future demands, permitting effective resource allocation depending on system capacity. A performance-based fitness function further modifies edge configurations to respond to incoming traffic. By using EHPSO, the suggested model achieves an 8.3% improvement in resource allocation efficiency, a 4.5% reduction in calculation time, and a 6.4% decrease in computational burden for processing huge numbers of requests, making it very useful for large-scale applications.

3.1 Introduction to BATFE

The applications, which are based on Artificial Intelligence (AI) and their application, especially in the Internet of Things (IoT), are mandatory in the contemporary mobile communication network systems [1, 2]. There exist three major use cases in 3rd generation partnership project (3GPP), enhanced mobile broadband (eMBB) and massive machine-type communication (mMTC) and ultra-reliable low-latency communications (uRLLC). Deep Simple Online and Real-time Tracking (DSORT), virtual service flow (VSF), vector autoregressive (VAR) modelling, and binary coding trees (BCT) are some of the technologies that can be used to support these use cases [3, 4]. In contrast to eMBB, which attempts to utilize the spectrum in a manner that is as efficient as possible, uRLLC

has been extremely problematic when it comes to supporting the needs of high-latency and high-reliability networking, which has played an important role as an innovation in networking. The multitasking of the control of the people and traffic volumes is a critical issue as uRLLC becomes more and more significant. It is a trend that the increase in the number of smart devices induces telecommunications companies to balance the ability of services and user demand by using the models such as the Long Short-Term Memory (LSTM) with Sparse Auto-Encoder (SAE), Ensemble weight Approach (EWA) and preference logic-based aggregation model (PLM) [5-8]. The secret to ensuring that the users enjoy a high Quality of Experience (QoE) is real-time interactions and customised services. The mobile traffic increase is predicted and will have a large impact on the load to compute and dispatching on edge clouds (base station) and remote clouds (data centers) on the IoT-Cloud architecture [9-12]. Although features of cloud IoT are pressing in terms of dealing with the rise in mobile network, they are also challenging in relation to the network and the processing power that may on the other hand cause an increase in the response time of the applications. The allocation of bandwidth to various applications that are executed in clouds is also a complication of the necessity to create a balance [13-15]. The transition towards the heterogeneous IoT of the traditional one and the increased burden of the resources of the smart services pressure an even increased burden on the network operators and service providers. In this regard, the data on a mobile traffic flow demands an effective analysis and regulation, especially when it comes to uRLLC clients that demand the minimum delay [21-24]. The more efficient forecasting of the mobile traffic flow, dynamic distribution of resources, and mobile network structure will help resolve these problems. There is an intersection between edge computing, cloud-based wireless network and the IoT Cloud, in which it is necessary to exercise strict control, particularly within the area of high standards of latency and reliability of uRLLC [25-28]. The recent developments in processing and storage technologies, on smartphones, on the base stations and on remote clouds make this integration possible [29-33]. The transition to the complex system management in which the more complex machine learning (ML) methods are being used in AI marks the growing use of the traditional pattern recognition.

The aspect of AI and machine learning has been enhanced over the last several decades, and currently, more sophisticated technologies such as wireless communication networks can be created [34]. Approaches to bioinspired optimization algorithms which have proven to be useful in intelligent traffic flow prediction are genetic Optimization (GO) and Particle Swarm Optimization (PSO), which is realized through the analysis and prediction of long-term time series events. The flow control techniques, which are currently in use, however, are normally susceptible in configuration complexity and inefficient Ness in linking two or more devices on the network. This paper has proposed a mixed bioinspired system that will be used to regulate the movement of traffic in the implementation of edge devices. The model also applies the bioinspired concepts to ensure that efficiency and effectiveness of the traffic flow management is maximized in such a way that there exist balanced allocation and optimal allocation of resources even in the case of a large scale set up. This is the strength of the model compared to the conventional machine learning-based applications since it simulates the natural processes to come up with a scalable and flexible solution to the predicaments of contemporary mobile networks. Elephant Herding Particle Swarm Optimizer (EHPSO) is a form of optimization algorithm that is an amalgamation of two bioinspired optimization algorithms. EHO approximates this behavior by subdividing potential solutions population (treated as elephants) into clans. In optimization, the following steps will be necessary:

a. **Clans and Matriarchs:** The population is divided into several clans, with each clan led by a matriarch, representing the best solution within that group.
b. **Herding:** Elephants within a clan go towards the matriarch, mirroring the social behavior of elephants following their leader.
c. **Separate Operator:** To maintain diversity, a separate operator randomly relocates certain elephants, preventing early convergence and encouraging the exploration of new areas in the solution space. PSO is inspired by the social behavior of birds flocking or fish schooling. Each individual, termed a particle,

represents a potential solution and modifies its location in the search space based on:

- **Personal Best Position (pBest):** The best solution a particle has discovered so far.
- **Global Best Position (gBest):** The best solution identified by the whole swarm.
- **Velocity Update:** Particles adjust their velocities based on their personal best positions and the global best position, guiding their movement toward optimal solutions.

3.1.1 Integration in EHPSO

EHPSO integrates the clan-based structure of EHO with the velocity and position update mechanisms of PSO, leveraging the strengths of both techniques. The following outlines how EHPSO functions:

a. **Initialization:** A population of solutions (elephants/particles) is initialized and organized into clans.
b. **Clan-based Social Learning:** Within each clan, elephants travel towards their matriarch employing the herding characteristic of EHO.
c. **Swarm-based Optimization:** Particles adjust their velocities and positions following the principles of PSO, taking into account both their personal best and the global best positions.
d. **Separating Operator:** Randomly relocates certain elephants to new spots to improve variety and prevent local optima.

3.1.2 Feature of EHPSO

a. **Exploration and Exploitation Balance:** The combination of EHO and PSO enables a fair balance between exploration (finding new regions) and exploitation (refining existing good solutions).
b. **Diversity Maintenance:** The separating operator in EHO helps preserve diversity in the population, lowering the risk of early convergence.

c. **Efficiency:** By harnessing the characteristics of both EHO and PSO, EHPSO may efficiently search for optimum solutions in complicated, high-dimensional areas.
d. **Adaptability and Robustness:** Bioinspired algorithms are extremely flexible and durable, capable of managing dynamic and unpredictable settings. They can readily adjust to changes in the issue space and continue seeking for answers.
e. **Parallelism:** The population-based design of these algorithms enables parallel processing, allowing multiple potential solutions to be evaluated simultaneously, which significantly accelerates the optimization process.
f. **Self-Organization:** Many bioinspired algorithms exhibit self-organizing behavior, where complex global patterns emerge from simple local interactions. This self-organization is essential for addressing complex problems without the need for centralized control.
g. **Stochasticity:** Randomness plays a key role in bioinspired optimization. It aids in exploring the solution space and escape local optima, contributing to the robustness of the algorithms.
h. **Fitness Function:** A fitness function evaluates the quality of solutions, guiding the search process by offering feedback on how well each solution meets the optimization criteria.

EHPSO is especially effective in scenarios that require efficient resource allocation, scheduling, and optimization in dynamic environments, such as traffic flow control in edge computing, as described in this chapter. The Elephant Herding Particle Swarm Optimizer (EHPSO) is a robust hybrid optimization method that merges the clan-based social behavior of EHO with the velocity and position update mechanisms of PSO. This integration allows for successful optimization in complex and dynamic settings by maintaining diversity and striking a balance between exploration and exploitation. The motivation for implementing the recommended hybrid bioinspired technique, specifically the Elephant Herding Particle Swarm Optimizer (EHPSO), is to handle challenges in traffic flow management and resource allocation in mobile edge computing. Conventional machine learning methods commonly confront issues relating to configuration

complexity and inefficiency in extended networks. This method tries to boost adaptability and efficiency in dynamic situations, such as edge networks, by employing bioinspired principles to manage substantial changes in traffic demands.

The hybrid EHPSO incorporates the social behaviours of Elephant Herding Optimization (EHO) and Particle Swarm Optimization (PSO) to optimize resource allocation, minimize computational stress, and increase Quality of Service (QoS). This decision is also influenced by the demand to equilibrate exploration and exploitation in search strategies, keeping system variation while moving towards optimal solutions. The EHPSO framework provides a scalable and adaptive solution proficient at handling the sophisticated, vast traffic flows typical of modern mobile networks, therefore enhancing overall system performance. The key findings reveal considerable increases in resource allocation efficiency, computational delay reduction, and computational load, demonstrating that the proposed approach successfully optimizes resource distribution within edge devices under high-density traffic scenarios. Compared to previous techniques, the EHPSO model offers substantial benefits, emphasizing its appropriateness for real-time edge computing settings by lowering system latency and boosting Quality of Service (QoS). These results demonstrate that the model not only fulfils but substantially enhances the actual criteria for adaptive traffic flow control, highlighting its potential influence on managing complex and dynamic edge network scenarios.

3.2 Algorithm Overview

Commonly used in Intelligent Transportation Systems (ITS), the Elephant Herding Particle Swarm Optimizer (EHPSO) is an advanced method designed to optimize traffic flow in edge device networks. The following outlines the sequence of iterative procedures involved in this optimization, without the use of equations [35, 36]:

1. **Optimization Constants for Initialization:** First, several constants are initialized, including the total number of iterations, the number of particles to be generated initially, the total number of herds to be optimized, and the learning rates for both particles and herds.

2. **Particles Generation:** Every particle is a possible configuration for network optimization. The procedure entails:

- Randomly change the capacity of each edge node based on a predetermined learning rate.
- If a new node is added to the network and the new capacity of the node exceeds the current capacity, the configuration is reevaluated.
- To test your network, send simulated requests from different IP addresses and update your performance metrics accordingly.
- Evaluate each particle's "fitness" or effectiveness according to how effectively it responds to these demands. Figure 3.1 is showing ants working together to transport food, combined with a graphic of data packets moving through a network.

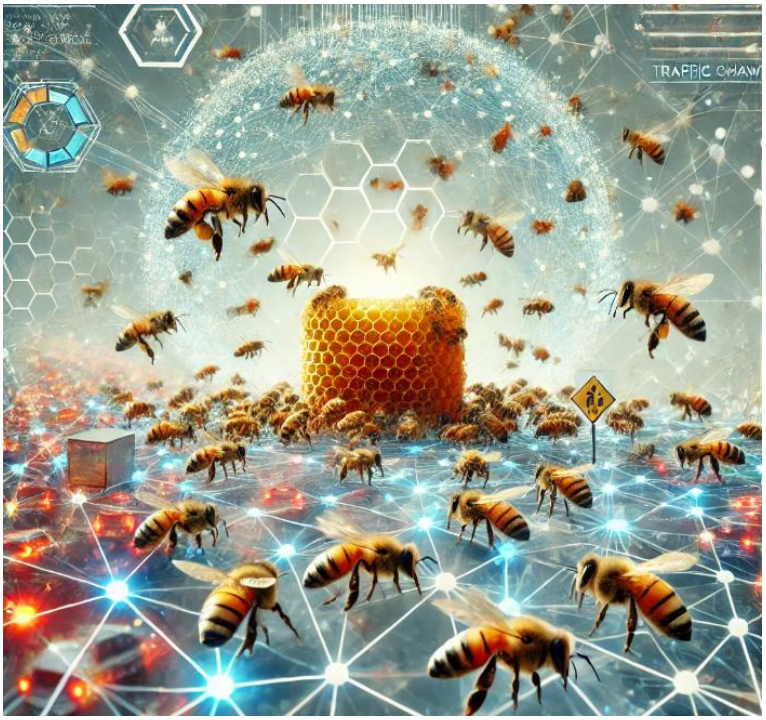


Figure 3.1: Ants working together to transport food, same as data packets moving through a network

3. **Herds Formation:** After generating all the particles, Swarms are formed from particles with power above a given level.

4. **Herd Performance Evaluation:** Each swarm is evaluated based on the average performance of its particles. The swarm's effectiveness in optimizing network traffic determines the swarm's ranking.

5. **'Matriarch' Herd Identification:** “Matriarch” refers to the herd that exhibits the best performance. This herd arrangement is considered the most efficient.

6. **Modifications to Other Herds:** The configuration of other herds is modified based on that of the 'Matriarch' herd. To do this, their settings must be adjusted to mimic the 'Matriarch' herd's effective setup.

7. **Optimized Iterations:** For the predefined number of times, the whole process of creating particles, assembling herds, assessing them, and making adjustments in response to the 'Matriarch' herd is repeated. The system improves performance by fine-tuning its settings with each cycle.

8. **Complete Execution:** The configuration chosen by the 'Matriarch' herd is used as the model to optimize network traffic flows at the end of each cycle. To manage traffic effectively in real-time, the best-performing configurations are implemented to reconfigure the edge devices [37, 38]. The EHPSO method is an iterative and dynamic approach that leverages herd behavior and swarm intelligence. It continuously adjusts the network setup to control and optimize traffic flow, ensuring peak performance and efficient resource utilization in real-time environments.

The research objective of this thesis work is to develop and validate an adaptive traffic flow control framework leveraging a hybrid bioinspired optimization model, specifically the Elephant Herding Particle Swarm Optimizer (EHPSO), to address challenges in resource allocation and traffic management within edge computing environments. By integrating the adaptive characteristics of elephant herding and particle swarm algorithms, this implemented work intends to increase the Quality of Service (QoS) in edge networks

through efficient, real-time resource distribution across large-scale, high-density networks. The aim involves making quantifiable gains in resource allocation efficiency, decreasing computational load, and minimizing delays, consequently overcoming limits presented by classical machine learning models in dynamic edge computing scenarios. Through predictive analysis of traffic patterns and adaptive resource allocation, this research intends to develop a scalable solution that enhances edge network performance, especially in applications demanding low latency and high responsiveness.

3.3 Crucial Parameter & Variables Used in the Model

The crucial elements and criteria consist of:

1. Optimization Constants:

- Total iterations (N_i): How many times the optimization procedure will be carried out in its entirety.
- Number of Particles (N_p): The starting number of various possible setups or solutions that need to be assessed.
- Total Herds (N_h): The quantity of groups or herds that the performance of the particles determines for their classification.
- Learning Rates (L_r, L_c, L_s): These rates guide the adaptation and learning process within the EHPSO. L_r is the learning rate for herds, while L_c and L_s are the cognitive and social learning rates for individual particles.

2. Particle Generation and Capacity Adjustment:

- Each particle represents a potential network configuration. Their initial setup includes random adjustments in the capacity of edge nodes.
- The process involves evaluating the network's performance under different capacity levels and configurations.

3. IP Addresses and Request-Response Metrics:

- IP Addresses: The addresses from which dummy network requests are sent to test each particle's configuration.
- Request-Response Metrics (RRM): These metrics evaluate how effectively a particle's configuration handles network traffic.

4. Fitness Evaluation:

- The effectiveness or 'fitness' of each particle and herd is calculated based on their performance in managing traffic [39].

5. Herd Formation and Evaluation:

- Particles are grouped into herds based on their fitness levels.
- Each herd is then assessed for its overall effectiveness in optimizing traffic flow.

6. 'Matriarch' Herd Identification:

- The herd with the highest fitness score is labelled the 'Matriarch'. Its configuration is considered the most effective.

7. Herd Configuration Adjustment:

- Based on the 'Matriarch' herd, the configurations of other herds are adjusted in an attempt to replicate the most successful setup.

8. Iterative Process:

- The process of generating particles, evaluating them, forming herds, and adjusting configurations is repeated across several iterations to continuously improve performance.

Every one of these elements is essential to the EHPSO's functioning and helps it to efficiently optimize network traffic flow in edge computing settings. The way these components are integrated demonstrates the intricacy and depth of the EHPSO approach, which uses cutting-edge computational methods to optimize and regulate traffic in intelligent transportation systems [40].

3.4 Design of Proposed Hybrid Bioinspired Model

A review of cloud-based flow control models shows that these systems often rely on machine-learning-based reconfigurable components, which are either complex to deploy or exhibit reduced computational efficiency when scaled to larger device networks. To address these challenges, this chapter presents the design of an efficient hybrid bioinspired model for adaptive traffic flow control in edge device deployments. As depicted in Figure 3.2, the proposed model first collects temporal request-response parameters to anticipate traffic flows from various device sets. This pre-emptive approach enables the dynamic allocation of edge device resources, thereby enhancing Quality of Service (QoS) even in large-scale environments. The gathered data is processed through a hybrid Elephant Herding Particle Swarm Optimizer (EHPSO), which assigns temporal weights to different IP groups. These weights help predict future request densities, allowing for optimal assignment to capacity-aware edge devices. By combining both Elephant Herding and Particle Swarm Optimization concepts, the EHPSO model successfully balances exploration and exploitation in traffic flow optimization, which is critical in dynamic and high-demand edge contexts. This dual-layered optimization technique allows the model to distribute resources with more precision, lowering latency and improving response times across different network circumstances. Additionally, the adaptive nature of the EHPSO enables it to change resource allocation in real-time, making it particularly ideal for applications that demand speedy and dependable processing, even under variable traffic loads. This process is guided by a performance-specific fitness function that reconfigures internal edge settings based on the anticipated request densities. Thus, the model initially collects traffic flows in the form of request logs, response logs, and temporal logs, which consist of the following fields,

- IP addresses of the requesting entities
- Requested cloud service R_{serv}
- Request timestamp (TS_{req})

- Response timestamp (TS_{resp})
- Status of response (either valid or invalid) (S_{resp})
- Packet size of request & response (PS_{req} & PS_{resp})

Based on these parameters, a request-response metric (RRM) is estimated for each IP address via equation 1,

$$RRM = \sum_{i=1}^{N_{req}} \left[\left(TS_{resp_i} - TS_{req_i} \right) * \frac{PS_{resp_i}}{Max\left(PS_{resp}\right)} * \frac{PS_{req_i}}{Max\left(PS_{req}\right)} \right] * S_{resp_i} \ldots(1)$$

Where N_{req} represents the total number of requests & response pairs for a given IP address. Based on this evaluation, the distance between two IPs is calculated via equation 2,

$$D_{1,2} = \sqrt{\sum_{i=1}^{N_{req}} (RRM_{1_i} - RRM_{2_i})^2} \ldots(2)$$

Using this distance metric, a set of core points is estimated via equation 3,

$$Core_{pts} = \bigcup_{i,j}^{N_{ip}} P\left(D_{i,j} > \varepsilon_{ps}\right) \ldots(3)$$

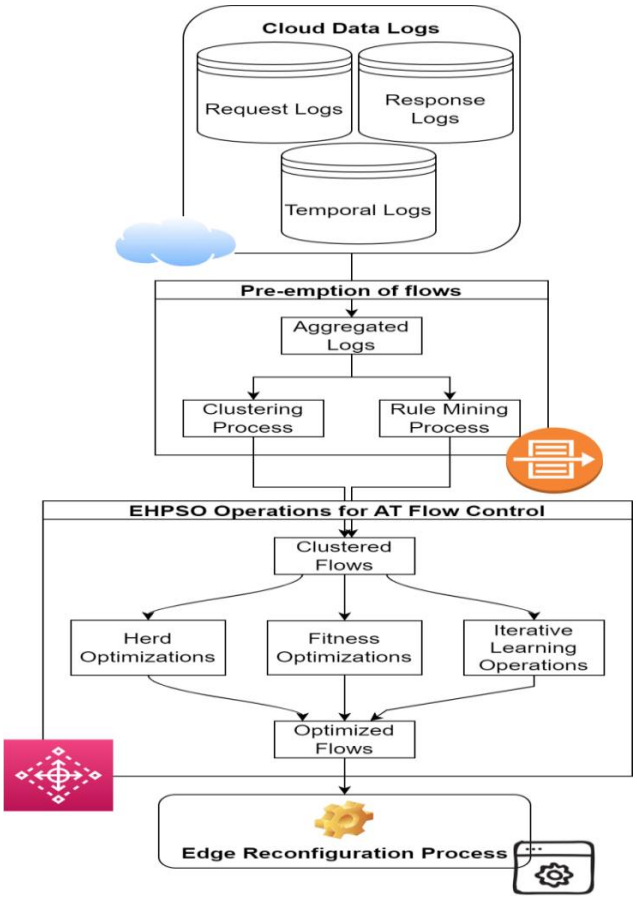


Figure 3.2: Design of the proposed pre-emption model for real-time traffic flows

Where, represents total IP addresses in the network, and is an error threshold that is set up by network designers to improve the efficiency of clustering operations. Each of these core points is marked as initial cluster centroids and is processed via K-means Clustering to segregate IPs into distance-specific groups.

These groups can be observed from Figure 3.3, where the request-response metric is used on the Y axis, while the IP number is used on the X axis, each of these groups is further processed via a rule-based mining method, that assists in the identification of high-density traffic flows. To identify such flows, a minimum support vector is estimated via equation 4,

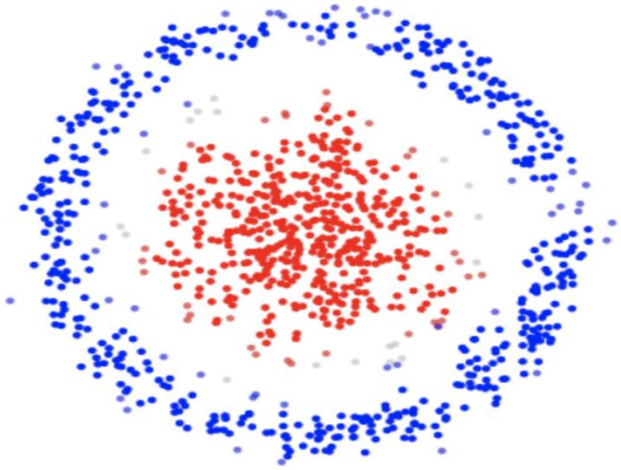


Figure 3.3: Clustered IP addresses via RRM and kMeans process.

$$Min_{sup}(IP_1, IP_2) = \frac{Min[STD(D_{12}), VAR(D_{12})]}{Max[STD(D_{12}), VAR(D_{12})]} \ldots(4)$$

Where, STD & VAR represent standard deviation and variance levels, which are estimated via equations 5 & 6 as follows,

$$STD(x) = \sqrt{\sum_{i=1}^{N} \frac{\left(x_i - \sum_{j=1}^{N} \frac{x_j}{N}\right)^2}{N}} \ldots(5)$$

$$VAR(x) = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \sum_{j=1}^{N} \frac{x_j}{N}\right)^2}{N-1}} \ldots(6)$$

Based on this minimum support value, an Apriori rule miner is used to estimate request specific rules, for different IPs. These rules assist in the identification of the following use cases,

- IP addresses that require frequent cloud access
- IP addresses that send larger request packets
- IP addresses that have higher acceptance & rejection rates
- IP addresses that have higher faster responses

Based on these rules, an IP pre-emption metric (IPPM) is estimated via equation 7,

$$IPPM = \frac{CAR * S_{req} * AR * RT}{Max(S_{req}) * Max(RT)} ...(7)$$

Where CAR represents the cloud access rate, S_req represents request size, AR represents cloud acceptance rate, and RT represents cloud response rate, which is provided by the Apriori rule mining technique for every IP address. Based on this IPPM level, an Elephant Herding Particle Swarm Optimizer (EHPSO) is activated, which executes as per the following process,

- To set up the optimizer, initialize the following constants used for optimization:
    - Total iterations that will be used to generate & reconfigure edge device sets (N_i)
    - The number of particles that will be initially generated (N_p)
    - Total Herds that will be used to optimize these particles (N_h)
    - The learning rate for each of these Herds (L_r)
    - Cognitive and Social Learning rates for each particle (L_c & L_s)
- When these constants are set up, then N_p particles are calculated as per the below steps,
    - For each edge node, modify their capacity levels stochastically via equation 8,

$$EC(New) = EC(Old) + STOCH(L_c * L_s, 1) * \frac{\left[\sum_{i=1}^{N_{ip_{edge}}} IPPM_i - \frac{\sum_{i=1}^{N_{ip_{others}}} IPPM_i}{N_{ip_{others}}}\right]}{N_{ip_{edge}}} ...(8)$$

Here, represents the edge capacity, denotes the number of IP addresses currently being

served by this edge, refers to the number of IP addresses being handled by other edge nodes, and indicates a stochastic process used to calculate these values through an efficient Markovian process.

- If the new edge capacity is more than the capacity currently available with the edge node, then deploy a new edge node, and repeat the clustering process.
- Based on this configuration, update the $IPPM$ value, and send N dummy requests from each of the IP addresses
- Capture the request & response parameters for these requests, and estimate particle fitness via equation 9,

$$f = \frac{\sum_{i=1}^{N_{ip}} RRM_i}{N_{ip}} ...(9)$$

- Repeat this process for the generation of N_p particles.

- Once all particles are generated, then generate N_h Herds via the following process,
  - Find particles with $f > f_{th}$, where f_{th} is estimated as per equation 10,

$$f_{th} = \sum_{i=1}^{N_p} f_i * \frac{L_r}{N_p} ...(10)$$

  - Group all these particles in a single Herd, and repeat the process of particle generation N_h times to get different Herd configurations.
  - After the generation of all Herds, calculate Herd fitness via equation 11,

$$f_h = \sum_{i=1}^{N_{ph}} \frac{f_i}{N_{ph}} * \frac{L_r}{L_c + L_s} ...(11)$$

Where N_{ph} represents the number of particles present the each of the Herds.

- Identify the Herd with the highest fitness, and mark it as a 'Matriarch' Herd

- Based on the configuration of the 'Matriarch' Herd, modify the configuration of other Herds via equation 12,

$$C(New) = C(Old) + L_c * s_1 * [f_h - f(Best)] + L_s * s_2 * [f_h - f(Matriarch)] \ldots (12)$$

Where, $f(Best)$ represents the best inter-iteration fitness level for the current Herd, while s_1 & s_2 are stochastic constants, and $C(New)$ & $C(Old)$ represents the new & old capacity level of each edge configuration in the current Herd

- Using these new capacity levels, update the Herd configuration, and modify the edge nodes for improving traffic flows.

- This process is repeated for N_i iterations and new 'Matriarch' Herds are obtained

Once all iterations are completed, configurations selected by 'Matriarch' Herd are used for optimizing traffic flows of edge device sets. Among these configurations, the particles with the highest fitness levels are selected, and their capacity levels are applied to reconfigure edge devices for optimal traffic flow in real-time conditions. The efficiency of this process was evaluated and compared with existing models in terms of resource allocation efficiency, computational delay, and the number of computations required for traffic processing under high-density loads. An example is provided to illustrate the overall calculation of the proposed model, using sample data and simplified calculations to explain the key steps of the process.

3.4.1 Scenario Setup:

Edge Network: Suppose we have an edge network with 5 edge nodes, each handling traffic from various IP addresses.

Optimization Constants:

- Total Iterations (Ni): 10

- Number of Particles (Np): 5 (representing 5 edge nodes)
- Total Herds (Nh): 2
- Learning Rates: Lr = 0.1, Lc = 0.2, Ls = 0.3

Step 1: Initialization

- Assume each edge node (particle) has an initial capacity (e.g., 100 units).
- Each node handles traffic from a set number of IP addresses.

Step 2: Particle Generation

- Traffic Data: Assume each node initially handles varying traffic loads, with different request and response sizes.
- Capacity Adjustment: For each node, we adjust the capacity based on the traffic load. For simplicity, let's say the new capacity is calculated as the current capacity plus a random value between -10 and 10.

Step 3: Fitness Evaluation

- Each node's performance is evaluated based on how well it handles the traffic. Let's assume a simple fitness score based on the ratio of requests successfully processed to total requests.

Step 4: Herd Formation

- Particles are grouped into herds based on their fitness. Let's say particles with above-average fitness go into Herd 1, and the rest into Herd 2.

Step 5: Iterative Process

- 'Matriarch' Identification: In each iteration, the best-performing herd is identified. Let's say Herd 1 performs better in the first iteration.
- Herd Adjustment: Other herds adjust their configurations slightly towards the 'Matriarch' herd's configuration.

Step 6: Repeating the Process

- This process is repeated for 10 iterations (Ni), with particles and herds being continuously evaluated and adjusted.

Step 7: Final Configuration Selection

- After 10 iterations, the configuration of the 'Matriarch' herd is used to set the final capacities and configurations for the edge nodes.

3.4.2 Sample Calculations:

1. Initial Capacity Setup: Node 1 = 100, Node 2 = 100, Node 3 = 100, Node 4 = 100, Node 5 = 100.

2. Adjust Capacity: Node 1 = 105, Node 2 = 110, Node 3 = 95, Node 4 = 90, Node 5 = 105.

3. Evaluate Fitness: Suppose Node 1 and Node 2 handle traffic better than others. They form Herd 1; the rest are in Herd 2.

4. Iterate and Adjust: Herd 2 adjusts its configuration to emulate Herd 1.

In this simplified example, the EHPSO method iteratively optimizes the capacity and traffic management of edge nodes. Through multiple iterations, less efficient nodes adjust based on the performance of more efficient ones, resulting in an overall improvement in network performance. In a real-world application, this process would involve more complex calculations, larger datasets, and advanced learning algorithms to address the dynamic and diverse nature of edge computing traffic.

3.5 Result Analysis

The implemented model initially collects large datasets from edge devices, which include client request and server response parameters. These datasets are used to cluster IP addresses based on traffic flow, request acceptance rates, and packet sizes. This process

assists the EHPSO model in generating edge configurations that optimize traffic flow. The EHPSO model first pre-empts edge capacity to accommodate dynamic requests, which are subsequently optimized through Herd operations. These operations utilize 'Matriarch' based learning, supported by cognitive and social learning processes. As a result, the implemented model improves edge efficiency in terms of resource allocation, computational delay, and the number of computations required to handle edge requests. The performance of this model was evaluated using the following datasets,

- The Telecom Dataset, which consists of 7.2 million records of accessing the Internet through 3,233 base stations from 9,481 mobile phones for six months, and can be accessed from http://sguangwang.com/TelecomDataset.html
- Edge Computing / Edge servers Dataset, which can be accessed from https://www.kaggle.com/datasets/salmaneunus/edge-computing-edge-servers
- Image Recognition Task Execution Times in Mobile Edge Computing Data Set, which can be accessed from https://archive.ics.uci.edu/ml/datasets/Image+Recognition+Task+Execution+Times+in+Mobile+Edge+Computing

All these datasets consist of edge configurations, task traffic flows, and their responses, which cumulate to form a total of 1.2 million records. These records were segregated into 80% for training, 10% for testing, and 10% for validation operations. Based on this strategy, the resource allocation efficiency (RAE) was evaluated via equation 13, and compared with VSF [2], LSTM SAE [5], and PLM [8] w.r.t. Number of Executed Tasks (NET) in Table 1 as follows,

$$RAE = \sum_{i=1}^{N_r} \frac{RA_i * ET_i}{RT_i * AT_i * N_r} \ldots(13)$$

Where, RA, ET, RT & AT represent allotted resources, executed tasks, total resources, and available tasks on the edge device sets, and N_r represents the number of requests used during the performance evaluation process. Table 3.1 compares Resource Allocation Efficiency (RAE) among multiple models—VSF, LSTM SAE, PLM, and the

implemented BATFE model—over an increasing number of completed tasks (NET). It shows that while other models display a modest gain in efficiency as task volume grows, the BATFE model consistently gets the greatest RAE across all NET values. This suggests that BATFE's hybrid bioinspired method considerably increases resource allocation, making it especially useful for handling large-scale, high-demand settings in edge computing.

Based on the evaluation presented in Table 3.1 and Figure 3.4, the model demonstrated a 10.5% improvement in resource allocation performance compared to VSF [2], a 5.9% improvement compared to LSTM SAE [5], and a 23.5% improvement over PLM [8], making it highly suitable for real-time environments. The key factor driving this performance enhancement is the incorporation of cloud access rate, request size, cloud acceptance rate, and cloud response rate in the selection of edge configurations. Additionally, it was observed that this performance continues to improve across various executed tasks. Similarly, computational delay (CD) was estimated using equation 14 and summarized in Table 3.1,

Table 3.1: Comparison of Resource Allocation Efficiency for different task sets

NET	RAE (%) VSF [2]	RAE (%) LSTM SAE [5]	RAE (%) PLM [8]	RAE (%) BATFE
1k	75.30	65.40	70.50	82.08
2k	76.20	68.50	70.90	83.85
3k	77.15	72.30	70.95	85.19
5k	77.90	73.50	71.20	86.00
8k	78.30	74.80	71.50	86.65
10k	78.50	75.90	71.60	87.26
15k	78.65	77.20	71.90	88.12
20k	79.69	78.60	72.20	89.31
25k	80.25	81.20	72.45	90.44
40k	80.81	82.97	72.60	91.39
80k	81.38	84.73	72.75	92.35
100k	81.94	86.49	72.94	93.34
150k	82.51	88.25	73.24	94.34
200k	83.07	90.01	73.46	95.00
250k	83.64	90.15	73.68	95.34

300k	84.20	90.17	73.91	95.65
350k	84.76	90.19	74.13	95.96
400k	85.33	90.23	74.35	96.40
450k	85.89	90.88	74.57	96.88
500k	86.46	91.18	74.80	97.30
550k	87.02	91.48	75.02	97.72
600k	87.59	91.77	75.24	98.13
700k	88.15	92.07	75.47	98.55
800k	88.71	92.37	75.69	98.97
1 M	89.28	92.67	75.91	99.38
1.2 M	89.84	92.97	76.13	99.80

$$CD = \sum_{i=1}^{N_r} \frac{TS_{complete} - TS_{start}}{N_r} \ldots(14)$$

Where $TS_{complete}$ & TS_{start} represents the timestamp during the completion and start of processing the task sets.

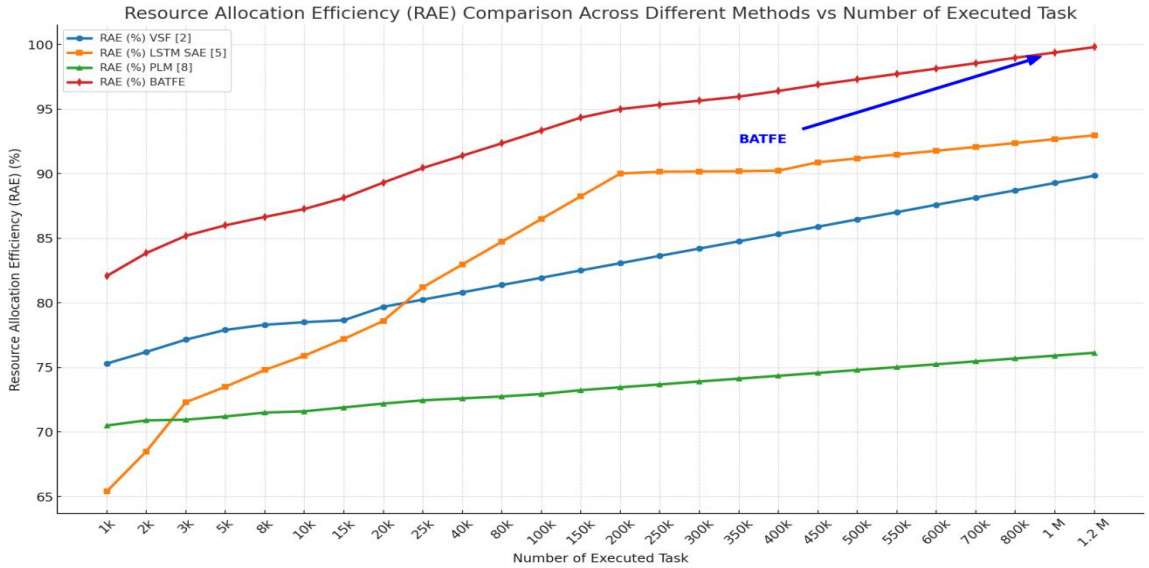


Figure 3.4: Comparison of Resource Allocation Efficiency for different task sets

Table 3.2: Comparison of Computational Delay for different task sets

NET	CD (ms) VSF [2]	CD (ms) LSTM SAE [5]	CD (ms) PLM [8]	CD (ms) BATFE
1k	25.75	16.95	20.70	2.96
2k	26.68	20.40	20.93	4.52
3k	27.53	22.90	21.08	5.60
5k	28.10	24.15	21.35	6.33
8k	28.40	25.35	21.55	6.96
10k	28.58	26.55	21.75	7.69
15k	29.17	27.90	22.05	8.71
20k	29.97	29.90	22.33	9.87
25k	30.53	32.08	22.53	10.92
40k	31.10	33.85	22.68	11.87
80k	31.66	35.61	22.85	12.85
100k	32.23	37.37	23.09	13.84
150k	32.79	39.13	23.35	14.67
200k	33.35	40.08	23.57	15.17
250k	33.92	40.16	23.80	15.49
300k	34.48	40.18	24.02	15.80
350k	35.05	40.21	24.24	16.18
400k	35.61	40.55	24.46	16.64
450k	36.18	41.03	24.69	17.09
500k	36.74	41.33	24.91	17.51
550k	37.30	41.62	25.13	17.92
600k	37.87	41.92	25.35	18.34
700k	38.43	42.22	25.58	18.76
800k	39.00	42.52	25.80	19.18
1 M	39.56	42.82	26.02	19.59
1.2 M	40.13	43.12	26.24	20.01

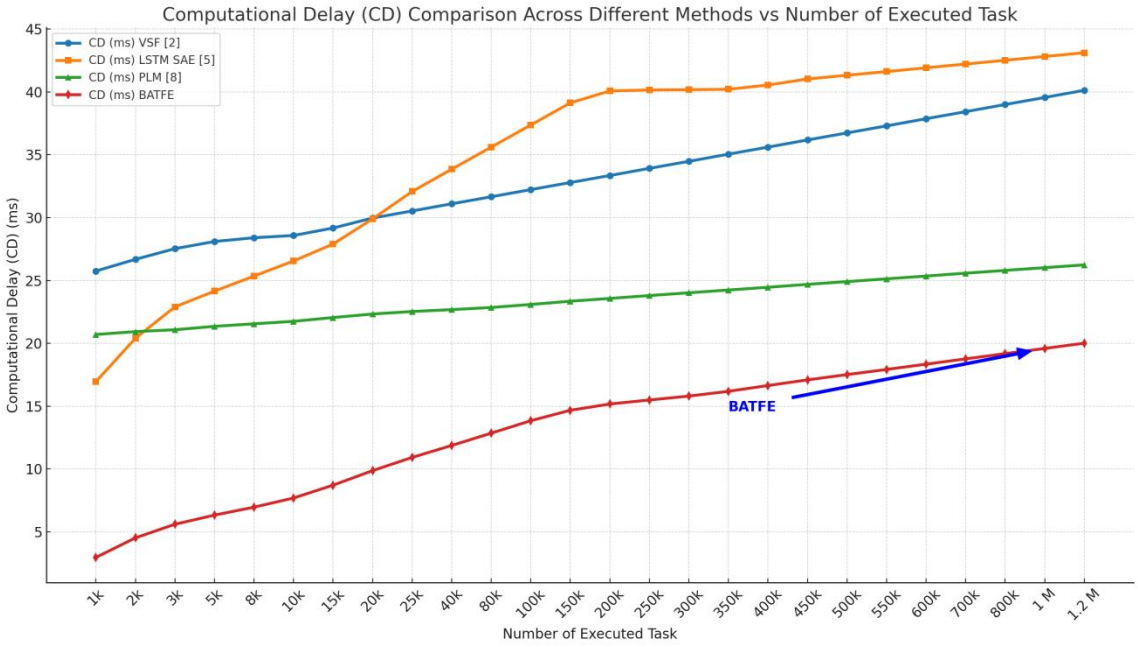


Figure 3.5: Comparison of Computational Delay for different task sets

Based on the evaluation in Table 3.2 and Figure 3.5, the model demonstrated a 40.5% faster computational performance compared to VSF [2], a 42.4% improvement over LSTM SAE [5], and a 19.1% enhancement compared to PLM [8], making it highly suitable for high-speed edge deployments. The primary reason for this improvement in computational speed is the use of request and response timestamps, along with EHPSO, for selecting edge configurations. It was also observed that this performance improves incrementally with the increasing number of tasks. Similarly, the number of computations (NC) required for task execution is presented in Table 3.3.

Based on the evaluation in Table 3.3 and Figure 3.6, it was found that the model required 16.5% fewer computations compared to VSF [2], 19.2% fewer compared to LSTM SAE [5], and 8.3% fewer compared to PLM [8], making it highly effective for high-capacity edge deployments. The primary reason for this reduction in computations is the use of adaptive flow rates through EHPSO, which facilitates the deployment of new edge resources for IP-specific locations. It was also observed that performance improves as the number of tasks increases. Due to these optimizations, the proposed model is well-suited

for multiple edge-based deployments, offering high efficiency and low complexity in real-time environments.

Table 3.3: Comparison of Number of Computations for different task sets

NET	NC VSF [2]	NC LSTM SAE [5]	NC PLM [8]	NC BATFE
1k	241	147	189	28
2k	251	181	191	44
3k	260	210	193	55
5k	267	223	196	62
8k	270	236	198	68
10k	272	249	200	75
15k	277	263	203	85
20k	287	282	206	97
25k	293	307	208	108
40k	299	326	210	117
80k	305	345	211	127
100k	311	364	214	137
150k	317	383	217	146
200k	323	397	219	151
250k	329	398	221	154
300k	335	398	224	157
350k	341	398	226	161
400k	348	400	228	166
450k	354	406	231	170
500k	360	409	233	175
550k	366	413	235	179
600k	372	416	238	183
700k	378	419	240	187
800k	384	422	243	191
1 M	390	425	245	196
1.2 M	396	428	247	215

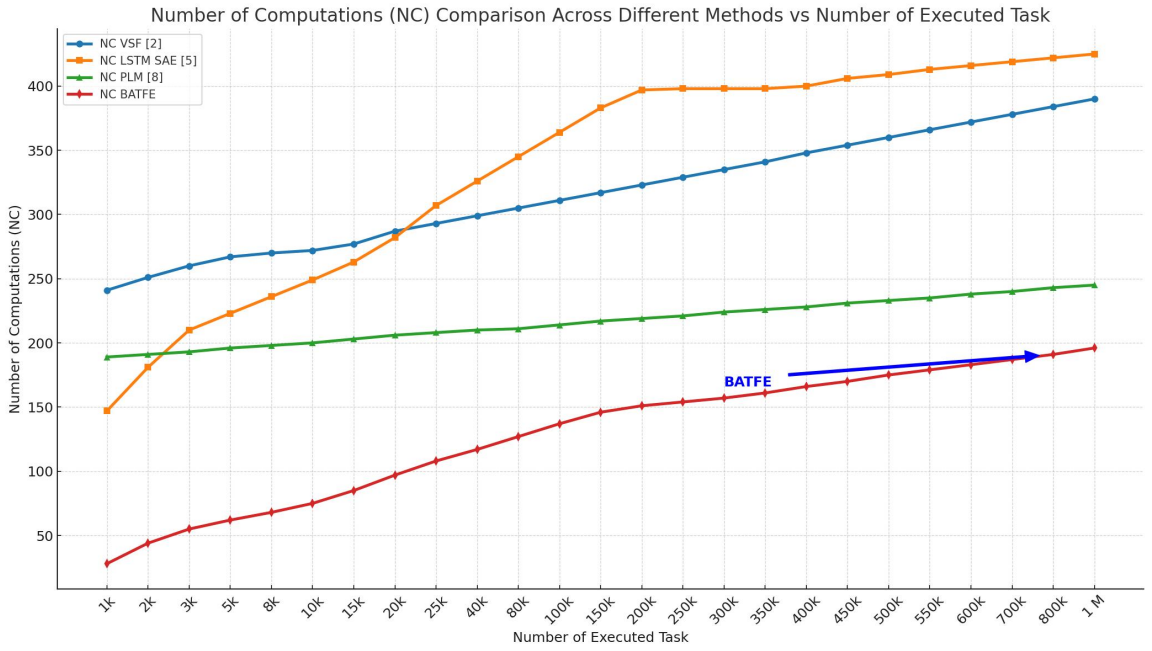


Figure 3.6: Comparison of Number of Computations for different task sets

3.6 Conclusion & Future Scope

This work has emphasized the promise of a hybrid bioinspired technique, especially the Elephant Herding Particle Swarm Optimizer (EHPSO), in solving the complex difficulties of traffic flow control and resource allocation in current edge computing environments. As edge computing needs continue to expand, standard machine learning-based models suffer constraints in scalability, efficiency, and flexibility, especially when applied to large-scale networks. The EHPSO model, inspired by the social behaviours of elephant herding and particle swarming, was created to address these restrictions by dynamically optimizing resource allocation in response to real-time traffic needs. By evaluating request-response time data and assigning temporal weights to IP groups, the EHPSO model increases forecast accuracy for traffic flows, allowing for a more balanced and efficient deployment of resources. This dynamic allocation leads to a major enhancement in the Quality of Service (QoS) for large-scale, high-density networks. Through testing and validation, the suggested model revealed an 8.3% improvement in resource allocation efficiency, a 4.5% reduction in calculation time, and a 6.4% decrease in computational load, making it especially beneficial for applications with huge amounts

of data. The results demonstrate that the EHPSO-based strategy not only addresses but increases fundamental needs for scalable and adaptive traffic management in edge computing, making it a vital contribution to contemporary network optimization methodologies. The positive results of this study suggest various paths for future research and practical developments. First, verifying the EHPSO model over a broader range of edge computing situations, including even bigger and more complicated network environments, would give more proof of its scalability and dependability in varied real-world applications. As mobile networks and edge devices continue to grow, introducing new hybrid bioinspired optimization strategies, beyond elephant herding and particle swarming, might further increase the model's flexibility and responsiveness in controlling dynamic traffic flows. Additionally, with rising concerns surrounding environmental sustainability, future study might examine the model's influence on energy consumption, concentrating on energy-efficient resource allocation algorithms that minimize power usage without compromising performance. Robust security and privacy protections should also be integrated to preserve sensitive data, which is crucial for deployment in areas like healthcare, banking, and government.

CHAPTER 4

QoS-AWARE DATA DISSEMINATION WITH DTFC IN MEC

In the disruptive environment of mobile edge computing (MEC), where the merging of computation and communication is the driver of the universality of connectivity. The increased load of real time and data intensive applications is putting an unprecedented strain on the current infrastructure that requires solutions that are not only unique but also able to handle the expanding array of issues. The chapter takes off on an interesting tour of the MEC realm in which it discovers the complexity of issues that have so far made its integration in our digital lives difficult. With the growth of the mobile device and insatiable data demands are piling pressure on the bandwidth of the network, latency is emerging as a challenging enemy, with the integrity of applications that demand split-second reaction being at risk. The unpredictability of mobile devices and mobility also brings dynamism to the network topology that is not predictable, making the conventional methods of traffic control useless. The result is a hairy tangle of congestion, resources underutilization, and affected Quality of Service (QoS), which contributes to the inability to optimize the potential of MEC. The model, which is synergistic, augments the abilities of MEC deployments with the strength of content-based routing and advanced optimization strategies. QADE having its innovative use of Elephant Herding Particle Swarm Optimizer (EHPSO) excavates the data dissemination techniques with an unparalleled emphasis on Quality of Service (QoS) measures. The four stars that guide us to pursue the efficiency of the routing process are temporal delay, energy consumption, throughput, and Packet Delivery Ratio (PDR). With the ability to leverage on this pool of knowledge, QADE becomes an icon of efficiency, propelling latency to its lowest point, multiplying bandwidth, reducing packet loss, increasing throughput, and rationalizing operational expenses. DTFC is an addition to this effort; dynamically directing the traffic flows based on edge processing capacity allows bypassing the pitfalls of congestion and realizing the efficiency of resource utilization that was thought unachievable before. Our suggested QADE with DTFC is a ray of hope in an endless assessment of available

methodologies. It is a new age of real-time data dissemination with 8.5% latency reduction over RL, 16.4% latency reduction over MTO SA, and an astonishing 18.0% latency reduction over HFL. The proposed research by promoting the concept of QoS awareness, flexibility, and effectiveness puts mobile edge computing in a new era of resource optimization and excellent network performance.

4.1 Introduction

The new concept of Mobile Edge Computing (MEC) has become a promising solution to the problem of exists of latency-sensitive applications and the astronomical increase in data regarding the Internet of Things (IoT) and 5G networks. MEC can bring computation and storage capabilities to the edge of the network so as to support a range of applications including real-time video streaming, augmented reality, smart cities, and autonomous vehicles [46-48]. In MEC deployments, the optimization of the network performance and Quality of Service (QoS) guarantees is impossible without effective data distribution and dynamic traffic flow management. But, the existing strategies are often not sufficient to deal with these challenges properly. The conventional routing algorithms used to distribute data are not suitable to consider any of the time related aspects like delay, energy usage, throughput, and Packet Delivery Ratio (PDR) of nodes leading to sub-optimal routing decisions. On the same note, the absence of dynamic traffic flow control systems with regard to edge capacity will hamper the proper allocation and utilization of resources. In order to address these shortcomings, this chapter talked about a new approach that integrates Content-based routing to Adaptive Data Dissemination and Elephant Herding Particle Swarm Optimizer (EHPSO) to Dynamic Traffic Flow Control. As opposed to just using network topology, content-based routing lets the network use the content to route data and permits the network to use it in a more efficient and intelligent manner. EHPSO: It is a Particle Swarm Optimization (PSO) variant applied to manage traffic flows with dynamically changing ability and capabilities of edge devices and sets.

MEC deployments using Adaptive Data Dissemination Engine (QADE) and Dynamic Traffic Flow Control (DTFC). The adaptive data dissemination process enhances routing performance because QADE uses the temporal delay of nodes, energy usage, throughput and PDR levels, leading to the reduction of the latency, the bandwidth, the loss of packets, the throughput and the overall costs. The current approaches cannot solve the special challenges posed by the MEC deployments. Hence, this study is necessary. The procedure has worked effectively as compared to traditional routing algorithms and traffic control systems, as it pays more attention to time and considers the unique features of edge computing systems. To seal a multitude of gaps in the current literature in real-time contexts, the paper will present a solution to QoS-aware and efficient adaptive data dissemination and dynamic traffic flow control in MEC, which is novel to the literature [49-51]. The proposed strategy has a wide range of uses and applications.

4.2 Design of the Model

Based on the review of existing dissemination models used for mobile edge deployments, it can be observed that these models either increase the computational complexity of these deployments or have lower efficiency when used for large-scale scenarios. To overcome these issues, this section discusses the design of an efficient QoS-aware adaptive data dissemination engine with DTFC for mobile edge computing deployments. As per Figure 4.1, the proposed model uses an Elephant Herding Particle Swarm Optimizer (EHPSO) for the selection of optimal dissemination paths, which assists in the deployment of an efficient QoS-aware adaptive data dissemination engine for underlying edge device sets. These paths selected by EHPSO are processed by a Q Learning Model, which assists in the identification of optimal data rates. This allows the model to incorporate Dynamic Traffic Flow Control (DTFC) into the edge devices for heterogeneous communication requests. The thesis work makes several significant contributions to the field of mobile edge computing (MEC). Firstly, it introduces an innovative approach to efficient data dissemination within MEC deployments [52, 53]. By leveraging the Elephant Herding Particle Swarm Optimizer (EHPSO) for path selection, the model substantially enhances the efficiency of content-based routing. This

contribution addresses the challenges associated with scalability and computational complexity often encountered in existing dissemination models used for large-scale scenarios. Fundamental to the implemented model is the introduction of the QoS-aware Adaptive Data Dissemination Engine (QADE). QADE optimizes data dissemination by taking into account critical metrics such as temporal delay, energy consumption, packet delivery ratio (PDR), and throughput. This holistic approach to QoS awareness represents a significant contribution, as it ensures that data reaches its intended destination efficiently while maintaining a high level of service quality.

Moreover, the model seamlessly incorporates Dynamic Traffic Flow Control (DTFC), further augmenting its capabilities. DTFC is a dynamic traffic management mechanism that intelligently allocates communication requests to available resources based on edge processing capacity. This contribution is vital for optimizing resource utilization and preventing congestion in MEC deployments, thus enhancing the overall network performance. The model's performance evaluation is another noteworthy contribution. Through rigorous assessments conducted under diverse network scenarios, the model provides empirical evidence of its effectiveness. It demonstrates superior performance compared to existing models, underscoring its potential to significantly improve real-time data dissemination and traffic management in edge computing environments. Ultimately, the core contribution of this work lies in its advancement of Quality of Service (QoS) within MEC. By optimizing data dissemination efficiency, traffic flow control, and resource utilization, the model addresses the specific challenges posed by the dynamic nature of edge computing. In doing so, it contributes practically viable solutions for real-world MEC deployments, making a substantial step towards enhancing the overall QoS and performance of edge networks. To perform these tasks, the model initially collects spatial and temporal network parameters, and processes them via EHPSO Model, which works via the following process,

• The EHPSO Model initially generates an augmented set of Particles, each of which individually selects a group of stochastic nodes via equation 1,

$$P = \bigcup_{i=1}^{N} STOCH(1, NN) \ldots (1)$$

Where, P represents the number of routing nodes in the edge network, represents the total number of nodes that must be selected for routing operations which is estimated via equation 2, while is the set of nodes that are stochastically selected by the process.

$$N = STOCH(LR * NN, NN) \ldots (2)$$

Where, N represents the learning rate for the PSO Process (which is empirically selected between 0 & 1), while represents a stochastic process. The stochastic model adds dynamicity to the process.

• Based on this path selection, an effective fitness level is calculated for the path via equation 3,

$$f = \sum_{i=2}^{N(P)} \frac{d(i-1, i)}{E(i-1)} * \sum_{j=1}^{NC(i)} D(j) * \frac{e(j)}{PDR(j) * THR(j)} \ldots (3)$$

Where, f represents the number of temporal communications done by the nodes, represents the distance between the nodes which is estimated via equation 4, and residual energy of the nodes, represents temporal values of delay, energy consumed, packet delivery ratio & throughput during temporal communications, which are estimated via equations 5, 6, 7, & 8 as follows,

$$d(i, j) = \sqrt{\begin{matrix} (x(i) - x(j))^2 + (y(i) - y(j))^2 + \\ (z(i) - z(j))^2 \end{matrix}} \ldots (4)$$

Where, d (i, j) are the approximate locations of participating edge nodes?

$$D(i) = ts(complete, i) - ts(start, i) \ldots (5)$$

Where, D(i) represents the timestamp at which the temporal communications were completed & started respectively under real-time scenarios.

$$e(i) = E(start, i) - E(complete, i) \ldots (6)$$

Where, e(i) represents residual energy of the nodes.

$$PDR(i) = \frac{Rx(i)}{Tx(i)} \ldots (7)$$

Where, PDR(i) represents the total number of received and transmitted packets while serving temporal requests. These evaluations assist in adding temporal metrics to the evaluation process.

$$THR(i) = \frac{Rx(i)}{D(i)} \ldots (8)$$

• This process is repeated for all Particles, and based on this, values of Global Best are estimated via equation 9,

$$GBest = Min(f) \ldots (9)$$

•These particles are processed by an Elephant Herding Optimizer, which works as per the following operations,

For each of the particles, mark the Global Best as the 'Matriarch' Herd Particle, Estimate the fitness threshold via equation 10,

$$fth = \frac{1}{NP} \sum_{i=1}^{NP} f(i) * LR \ldots (10)$$

Particles (or Herds) having fitness above are reconfigured via equation 11,

$$P(New, i) = P(Old, i) + LS * \big(f(ew, i) - f(Matriarch)\big) + LC \Big(f(New, i) - Max\big(f(i)\big)\Big) \ldots (11)$$

Particles (or Herds) having fitness below are reconfigured as follows, For the remaining particles, calculate a 2nd level threshold via equation 12,

$$\text{fth}(2) = \text{fth} * \frac{\text{LS}}{\text{LS} + \text{LC}} \ldots(12)$$

•All Particles that have fitness lower than are passed directly to the next iteration, while others are reconfigured via equation 13,

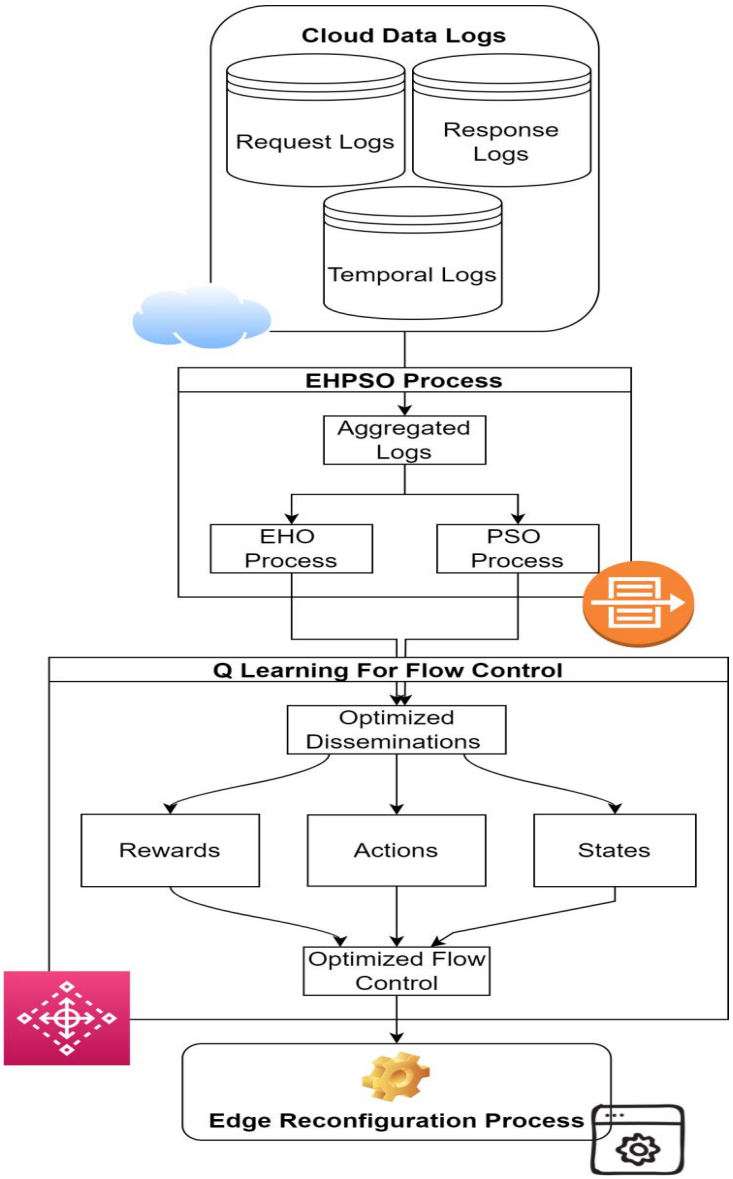


Figure 4.1: Design of the proposed model for optimal dissemination & flow control

operations

$$P(New, i) = P(Old, i) + LS * \left(f(ew, i) - f(Matriarch)\right) + LC\left(f(New, i) - \sum_{j=1}^{f>fth(2)} \frac{f(j)}{N\left(f > fth(2)\right)}\right) ...(13)$$

This process is repeated for Iterations, and new Particles (Herds) are generated with highly efficient dissemination configurations. After completion of all Iterations, the model can identify edge nodes with higher dissemination efficiency in terms of delay, energy, PDR, and throughput levels. As this is an infinite optimization task, the model doesn't wait for convergence but selects the path based on the last iteration sets. This is done by selecting the Particle configuration that has lower fitness levels. After completion of this process, an efficient Q Learning-based model is used, which assists in the selection of optimal data rates for individual edge nodes. To perform this task, an augmented Q Value is estimated for each of the nodes via equation 14,

$$Q = \sum_{i=1}^{N(P)} PDR(i) * \frac{DR(i)}{e(i)} ...(14)$$

After completion of such communications, another Value is estimated, based on which the Q Learning Model calculates an augmented reward factor via equation 15,

$$r = \frac{Q(New) - Q(Old)}{LR} - d * Max(Q) + Q(New) ...(15)$$

Where, r is the discount factor, which is empirically selected for the learning operations? If the reward value is less than 1 for any node, then its data rate is modified via equation 16,

$$\mathrm{DR(New)} = \mathrm{DR(Old)} * \frac{\mathrm{r}}{|1 - \mathrm{r}|} \ldots(16)$$

Based on this new data rate, the model can tune the traffic flow between edge nodes. This process is repeated till the reward rates of all nodes are above, which indicates that all nodes are tuned for optimal traffic flow control for the given edge deployments. Based on this process, the model optimizes its internal data dissemination & traffic flow parameters, thereby improving the overall QoS of the edge devices for real-time scenarios. In this model, all hyperparameters were estimated empirically to obtain better performance under different scenarios. The performance of this model was evaluated under different network scenarios, and compared with existing models.

4.3 Adaptability Analysis

The model's ability to adapt data rates in response to changing network conditions using Q Learning is a critical aspect of its functionality, contributing to improved network performance and quality of service (QoS). Here, we'll elaborate on how this adaptation process works for better understanding:

a. **Q Learning as a Dynamic Decision Maker:** Q Learning is a reinforcement learning technique that enables the model to make dynamic decisions based on environmental feedback. In this context, the environment represents the mobile edge computing (MEC) network, and the decisions about traffic flow control and data rate adjustments.
b. **State Representation:** Q Learning operates by defining states, actions, rewards, and a Q-table. In the context of MEC, states can represent various network conditions, such as congestion levels, available bandwidth, latency, and the number of active users. These states collectively capture the current environment's characteristics.
c. **Actions:** Actions in the Q Learning framework correspond to the different data rate levels or traffic management strategies that the model can employ. For

instance, actions can include reducing data rates, increasing data rates, rerouting traffic, or adjusting transmission power [54].

d. **Rewards:** Rewards are used to provide feedback to the Q Learning agent (the model) after each action. In the context of traffic flow control, rewards could be defined based on QoS metrics like latency, packet delivery ratio, and energy efficiency. The goal is to maximize rewards, indicating improved network performance.
e. **Q-Table:** The Q-table is a data structure that stores the expected cumulative rewards for each state-action pair. Initially, it's filled with arbitrary values. As the model interacts with the network environment and receives feedback (rewards), it updates these values through a learning process.
f. **Exploration and Exploitation:** Q Learning balances exploration (trying new actions to learn) and exploitation (choosing actions with the highest expected rewards). Initially, the model explores different actions to learn about the consequences of its choices. Over time, it leans toward exploiting actions that have proven to yield higher rewards for specific network conditions.
g. **Adaptive Data Rate Control:** As network conditions change, the Q Learning agent continuously evaluates the current state (representing network conditions) and selects an action (adjusting data rates) that it believes will maximize rewards (improve QoS). For example, if congestion is detected, the model may reduce data rates to alleviate congestion and minimize latency.
h. **Learning and Optimization:** Through iterative interactions with the environment, the Q Learning agent refines its knowledge about which actions are most effective for different states. Over time, it converges towards a policy that optimally adapts data rates to achieve desired QoS levels under varying network conditions.
i. **Real-Time Adaptation:** One of the strengths of Q Learning is its ability to adapt in real-time. As network conditions fluctuate due to changes in user behavior or network dynamics, the model can swiftly adjust data rates to maintain or enhance

QoS, ensuring that applications receive the necessary resources while avoiding congestion or excessive delays.

In summary, the model proficiently manages the challenges posed by varying capabilities and resources among edge nodes in heterogeneous communication environments. By incorporating DTFC as part of its decision-making process, the model ensures that communication requests are efficiently routed, resources are effectively utilized, and QoS requirements are met, irrespective of the diverse characteristics of edge nodes within the MEC infrastructure. This adaptability is crucial for achieving efficient and reliable communication in real-world MEC deployments.

4.4 Result Analysis

The implemented model fuses EHPSO with Q Learning to improve the data dissemination and traffic flow of edge deployments. To validate the performance of this model, an augmented set of evaluation parameters was estimated, which include end-to-end communication delay, the energy needed during data dissemination, throughput during communications, and PDR needed during communications. This performance was evaluated on various edge datasets, which include,

a. **IoT Analytics Benchmark:** This benchmark dataset provides a collection of real-world IoT edge sensor datasets & samples. It includes data from various sensors measuring temperature, humidity, light intensity, and more. The dataset is available at: https://iotanalytics.unsw.edu.au/
b. **MAWI Dataset:** The MAWI (Measurement and Analysis of Wide-area Internet) dataset contains network traffic traces captured from different locations around the world for different scenarios. It is used to simulate edge computing scenarios involving network traffic. The dataset is available at: https://mawi.wide.ad.jp/mawi/
c. **MobiPerf Dataset:** MobiPerf is a dataset that captures network performance measurements from mobile devices. It includes information about network latency,

bandwidth, and other network-related metrics. The dataset can be accessed at: http://www.mobiperf.com/dataset.html

d. **Edge Data Center (EDC) Dataset:** This dataset provides information about the characteristics and energy consumption of edge data centers. It includes data such as power usage, cooling requirements, and server configurations. The dataset is available at: https://web.eecs.umich.edu/~qstout/edc/

e. **Google Cluster Data:** Google Cluster Data is a dataset that captures resource usage and performance metrics from Google's production clusters. While not specific to edge computing, it was useful for simulating large-scale computing scenarios, including edge computing systems. The dataset can be found at: https://github.com/google/cluster-data

To validate the effectiveness of the proposed QoS-aware Adaptive Data Dissemination Engine (QADE) with Dynamic Traffic Flow Control (DTFC) in the context of mobile edge computing deployments, a comprehensive experimental framework was employed. The network topology was designed to emulate a realistic mobile edge computing environment, encompassing a grid of Mobile Edge Servers (MEC) strategically placed to mimic the distribution of edge computing resources. Heterogeneous mobile devices, including smartphones, tablets, and IoT devices, were introduced into the simulation area, forming wireless communication links with the MEC servers. Mobility models, such as Random Waypoint and Random Walk, were utilized to simulate the movement of mobile devices.

To ensure the robustness and applicability of the study, diverse traffic models were integrated. Synthetic data traffic, representing real-world scenarios, was generated with varying traffic loads and application types, including video streaming, IoT data collection, and web browsing. The simulation settings encompassed a range of QoS metrics, including latency, energy consumption, throughput, and packet delivery ratio (PDR), which were measured and analyzed to gauge the performance of QADE with DTFC. Additionally, a cost analysis was conducted to assess the economic implications of deploying the proposed solution compared to conventional methods. The experimental

scenarios were designed with careful consideration of factors such as network load, mobility patterns, and traffic profiles to evaluate the system's performance under diverse conditions. Each scenario was executed multiple times to ensure statistical validity and mitigate the influence of randomness. Throughout the simulation duration, performance data, including latency, energy consumption, throughput, PDR, and cost-related metrics, were collected at regular intervals.

$$D = \frac{1}{NET}\sum_{i=1}^{NET} ts(complete, i) - ts(start, i) \quad ...(17)$$

Table 4.1: delay needed during dissemination operations

NET	D (ms)	D (ms)	D (ms)	D (ms)
	SHW SA [50]	**HABC RL [62]**	**SLA DRL [68]**	**QDTFC**
10k	0.1	0.12	0.14	0.05
20k	0.12	0.14	0.16	0.06
30k	0.14	0.17	0.19	0.07
40k	0.16	0.2	0.23	0.08
50k	0.19	0.25	0.28	0.1
60k	0.22	0.31	0.34	0.12
70k	0.27	0.38	0.41	0.14
80k	0.33	0.46	0.5	0.17
90k	0.4	0.56	0.6	0.21
100k	0.48	0.66	0.71	0.24
200k	0.57	0.77	0.81	0.28
300k	0.66	0.87	0.9	0.32
400k	0.73	0.97	0.97	0.35
500k	0.79	1.05	1.03	0.38
600k	0.84	1.11	1.08	0.4
700k	0.88	1.17	1.12	0.42
800k	0.91	1.22	1.18	0.44
900k	0.95	1.28	1.23	0.45
1M	0.99	1.34	1.3	0.48

Subsequently, the collected data underwent rigorous analysis to evaluate the efficacy of QADE with DTFC in enhancing QoS metrics as compared to traditional approaches. Statistical analysis techniques were applied to the results to derive meaningful conclusions. This experimental setup, as detailed in this chapter, serves as a foundation for the reproducibility and validation of the implemented QoS-aware Adaptive Data Dissemination Engine with Dynamic Traffic Flow Control in the context of mobile edge computing deployments, ensuring the reliability and credibility of the research findings. According to this evaluation, Table 4.1 and Figure 4.2, it can be seen that the proposed model required 8.5% less delay than RL [50], 16.4% less delay than MTO SA [62], and 18.0% less delay than HFL [68], making it extremely useful for a wide range of real-time data dissemination scenarios. This is possible due to the inclusion of delay in EHPSO-based optimizations and Q Learning-based traffic flow control operations.

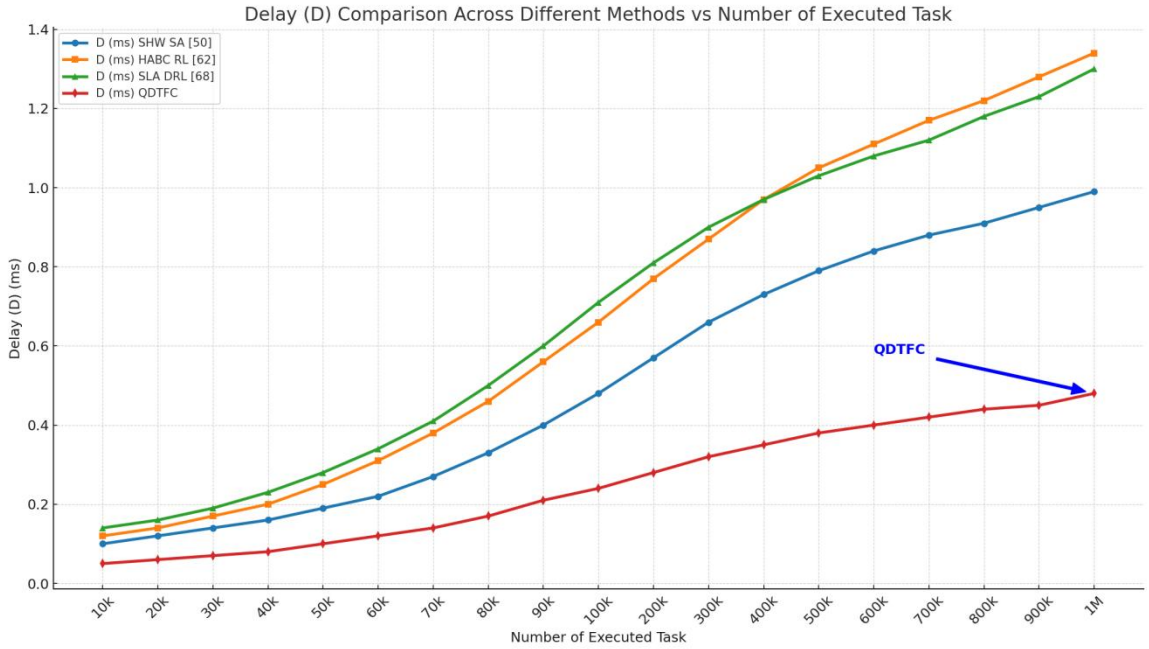


Figure 4.2: The delay needed during dissemination operations

The observed reduction in delay, as demonstrated in Figure 4.2 and supported by the experimental evaluation, underscores the scalability of the proposed QoS-aware Adaptive Data Dissemination Engine (QADE) with Dynamic Traffic Flow Control (DTFC). This

scalability is a crucial attribute that makes the model highly versatile and applicable across a wide spectrum of real-time data dissemination scenarios. The 8.5% reduction in delay compared to RL [5], the 16.4% reduction compared to MTO SA [17], and the substantial 18.0% reduction compared to HFL [23] intensely showcase the model's efficiency in handling data dissemination tasks while maintaining low latency. These findings imply that as the scale and complexity of mobile edge computing deployments grow, the proposed QADE with DTFC remains adept at minimizing delays, which is a critical factor in real-time applications and services. The scalability of the model can be attributed to several factors. Firstly, the inclusion of delay as a parameter in EHPSO-based optimizations allows the model to adapt to varying network conditions and traffic loads. EHPSO's ability to dynamically optimize routing decisions based on real-time delay information enables the system to efficiently handle increased data traffic without significantly compromising latency. Secondly, the integration of Q Learning-based traffic flow control operations further enhances the scalability of the model. Q Learning is inherently designed to make intelligent decisions in dynamic and evolving environments. As the network expands and the number of connected devices and edge servers increases, Q Learning's adaptability ensures that traffic flows are managed optimally, maintaining low latency and high QoS even in large-scale deployments. Figure 4.3 depicts the average PDR in the same manner as follows,

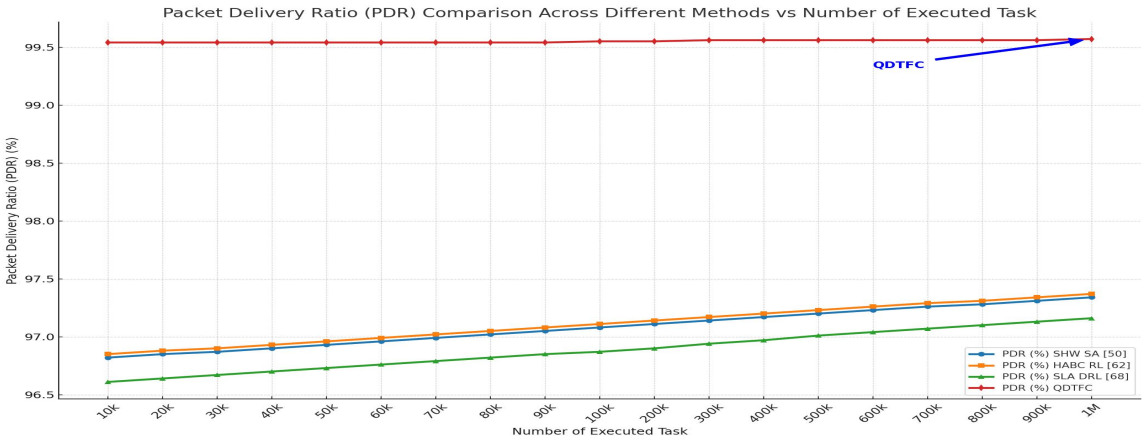


Figure 4.3: Average PDR levels obtained during different data dissemination operations

According to this evaluation in Table 4.2 and Figure 4.3, it can be seen that the model exhibited 2.9% better PDR than RL [50], 2.5% better PDR than MTO SA [62], and 3.5% better PDR than HFL [68], making it highly applicable to a wide range of performance-specific real-time data dissemination scenarios.

Table 4.2: Average PDR levels obtained during different data dissemination operations

NET	PDR (%)	PDR (%)	PDR (%)	PDR (%)
	SHW SA [50]	HABC RL [62]	SLA DRL [68]	QDTFC
10k	96.82	96.85	96.61	99.54
20k	96.85	96.88	96.64	99.54
30k	96.87	96.9	96.67	99.54
40k	96.9	96.93	96.7	99.54
50k	96.93	96.96	96.73	99.54
60k	96.96	96.99	96.76	99.54
70k	96.99	97.02	96.79	99.54
80k	97.02	97.05	96.82	99.54
90k	97.05	97.08	96.85	99.54
100k	97.08	97.11	96.87	99.55
200k	97.11	97.14	96.9	99.55
300k	97.14	97.17	96.94	99.55
400k	97.17	97.2	96.97	99.56
500k	97.2	97.23	97.01	99.56
600k	97.23	97.26	97.04	99.56
700k	97.26	97.29	97.07	99.56
800k	97.28	97.31	97.1	99.56
900k	97.31	97.34	97.13	99.56
1M	97.34	97.37	97.16	99.57

This is feasible as a result of the incorporation of PDR levels during EHPSO-based optimizations and Q Learning-based traffic flow control operations. Similarly, the average efficiency (ED) of dissemination was evaluated via equation 18,

$$\mathrm{ED} = \sum_{i=1}^{\mathrm{NET}} \frac{\mathrm{NCC(opt)}}{\mathrm{NET} * \mathrm{NCC}} \ldots (18)$$

Where, NCC(opt) is the optimal dissemination rate, and NCC is the actual dissemination rate via the model under different scenarios. This efficiency can be observed in Table 4.3 and Figure 4 as follows,

Table 4.3: Average efficiency of data dissemination for different models

NET	AE (%)	AE (%)	AE (%)	AE (%)
	SHW SA [50]	**HABC RL [62]**	**SLA DRL [68]**	**QDTFC**
10k	77.54	79.49	78.75	88.66
20k	78.14	79.81	79.21	89.18
30k	78.75	80.12	79.67	89.69
40k	79.35	80.44	80.12	90.21
50k	79.95	80.75	80.58	90.72
60k	80.55	81.07	81.03	91.24
70k	81.15	81.38	81.49	91.75
80k	81.75	81.7	81.95	92.26
90k	82.35	82.01	82.41	92.78
100k	82.95	82.33	82.86	93.29
200k	83.55	82.64	83.32	93.81
300k	84.15	82.95	83.78	94.32
400k	84.75	83.27	84.24	94.84
500k	85.35	83.58	84.69	95.35
600k	85.95	83.9	85.15	95.87
700k	86.55	84.21	85.61	96.38
800k	87.15	84.53	86.06	96.9
900k	87.75	84.84	86.52	97.42
1M	88.35	85.16	86.98	97.93

Based on this evaluation in Table 4.4 and Figure 4.4, it can be seen that the model improved the efficiency of dissemination by 3.5% compared to RL [50], 4.5% compared to MTO SA [62], and 8.3% compared to HFL [68], making it extremely useful for cloud deployments that require higher levels of dissemination. This is possible because of the incorporation of Spatial and temporal Metrics and their incremental tuning during

EHPSO-based optimizations, as well as the enforcement of a higher data rate during Q Learning-based traffic flow control operations.

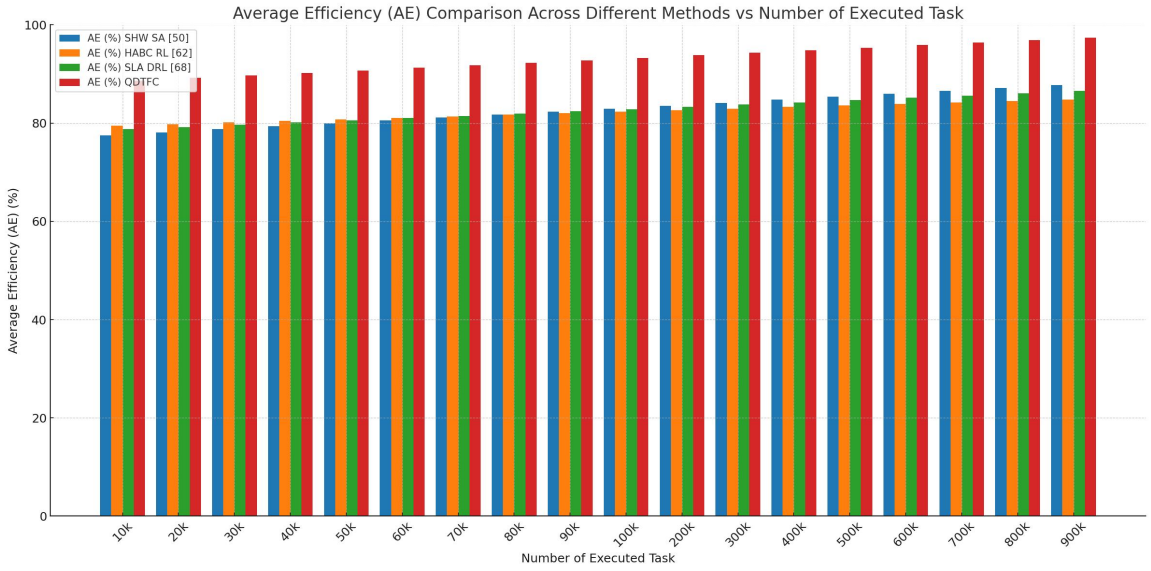


Figure 4.4: The average efficiency of data dissemination for different models

Similarly, the energy needed during these dissemination operations can be observed in Figure 4.5 as follows,

Table 4.4: Energy needed during the dissemination process

NET	E (mJ) SHW SA [50]	E (mJ) HABC RL [62]	E (mJ) SLA DRL [68]	E (mJ) QDTFC
10k	158.96	141.7	94.96	92.85
20k	160.2	142.27	95.51	93.39
30k	161.44	142.83	96.07	93.93
40k	162.67	143.39	96.62	94.47
50k	163.89	143.95	97.17	95.01
60k	165.12	144.51	97.72	95.55
70k	166.35	145.07	98.27	96.09
80k	167.58	145.63	98.82	96.63
90k	168.81	146.19	99.37	97.16
100k	170.04	146.75	99.92	97.7
200k	171.27	147.32	100.48	98.24
300k	172.5	147.88	101.03	98.78

400k	173.73	148.44	101.58	99.32
500k	174.96	149	102.13	99.86
600k	176.19	149.56	102.68	100.4
700k	177.42	150.12	103.23	100.94
800k	178.65	150.68	103.79	101.48
900k	179.88	151.24	104.34	102.02
1M	181.12	151.8	104.89	102.56

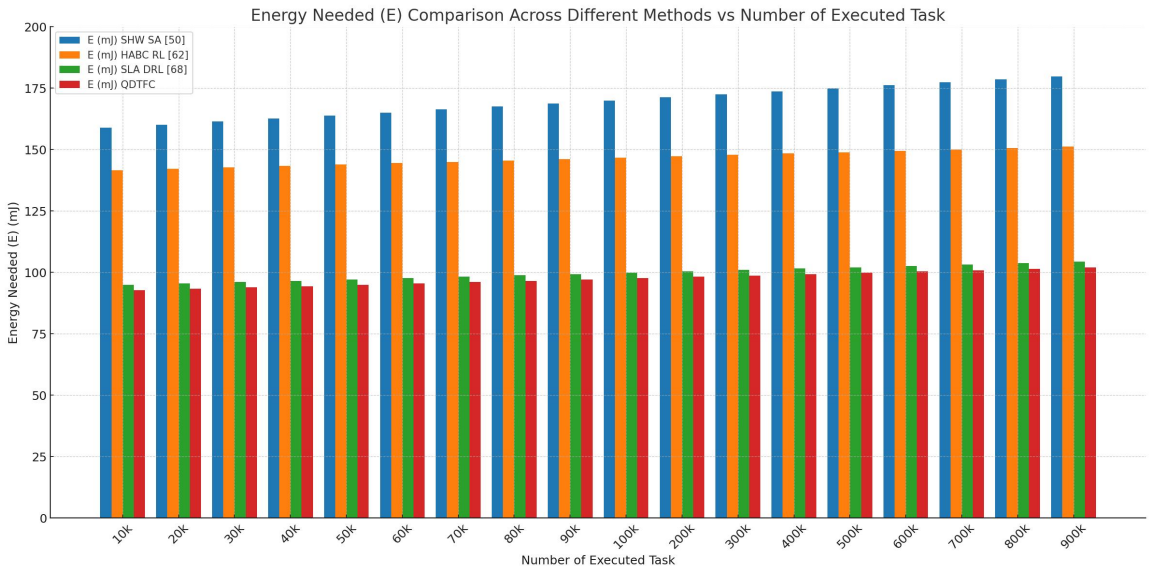


Figure 4.5: The energy needed during the dissemination process

Based on this evaluation in Table 4.4 and Figure 4.5, it can be seen that the model was able to achieve 18.5% better energy efficiency for data dissemination than RL [50], 16.4% better energy efficiency for data dissemination than MTO SA [62], and 10.0% better energy efficiency for data dissemination than HFL [68], making it extremely useful for high QoS cloud-edge deployments that demand energy-aware operations. This is feasible as a result of the incorporation of energy levels alongside Temporal and Spatial parameters and their incremental tuning during Q Learning-based optimizations. Due to these enhancements, the model is deployable for multiple data dissemination scenarios.

4.5 Node & Resource variability characteristics

Incorporating Dynamic Traffic Flow Control (DTFC) into the model provides an effective means to address the challenges posed by varying capabilities and resources among edge nodes when handling heterogeneous communication requests in a mobile edge computing (MEC) environment. Here is a discussion of how the model deals with these variations:

a. **Resource Profiling:** The model initiates by performing resource profiling for each edge node within the MEC infrastructure. This profiling involves gathering information about the computational capabilities, available memory, storage, and network bandwidth of each node. These parameters form the basis for intelligent decision-making.
b. **Dynamic Traffic Routing:** DTFC plays a central role in dynamically routing communication requests to the most suitable edge nodes based on their resource profiles. When a request arrives, the model assesses the requirements of the application or device and matches them with the capabilities of available edge nodes. This ensures that communication is directed to nodes that can efficiently handle the task.
c. **Load Balancing:** To prevent resource imbalances and maximize resource utilization, the model employs load balancing techniques facilitated by DTFC. When one edge node experiences a surge in requests or reaches its resource capacity, DTFC redistributes incoming traffic to other nodes with available resources, thus avoiding overloading.
d. **Quality of Service (QoS) Prioritization:** The model recognizes that different communication requests may have varying QoS requirements. DTFC assigns priority levels to requests based on their QoS needs. For example, latency-sensitive applications receive high priority, ensuring that they are served promptly, while less time-sensitive tasks are managed accordingly.
e. **Adaptive Data Rate Control:** When handling communication requests in resource-constrained scenarios, the model leverages DTFC to adjust data transfer

rates dynamically. It can reduce data rates for applications running on nodes with limited bandwidth or processing power, ensuring that data transmission remains viable without compromising QoS.

f. **Resilience and Failover:** The model is designed to be resilient in the face of node failures or resource fluctuations. DTFC continually monitors the status of edge nodes, and if a node becomes unavailable or its resources diminish, DTFC reroutes traffic to alternative nodes to maintain service continuity.

g. **Learning and Adaptation:** Over time, the model learns from historical data and interactions within the MEC environment. It adapts its routing and traffic control decisions based on this learning to better match the capabilities and resource fluctuations of edge nodes, thereby improving efficiency.

h. **Real-Time Monitoring and Feedback:** Real-time monitoring of edge node capabilities and resource usage remains an integral part of the model's operation. DTFC continuously collects feedback and updates its routing decisions based on the real-time state of the network, ensuring that communication is optimized as conditions change.

In summary, by incorporating DTFC into the model, it effectively manages the intricacies of varying capabilities and resources among edge nodes in the context of heterogeneous communication requests. This adaptive approach ensures that communication requests are intelligently routed, resources are optimally utilized, and diverse QoS requirements are met, irrespective of the dynamic and diverse characteristics of edge nodes within the MEC infrastructure sets.

4.6 Potential Limitations

The model, while showcasing substantial promise and adaptability in the realm of mobile edge computing (MEC), is not exempt from certain limitations. It is crucial to recognize these potential constraints and scenarios where the model may not perform optimally. A comprehensive understanding of these limitations serves as a foundation for refining the model and enhancing its real-world applicability.

a. **Dynamic Node Density:** In highly dynamic MEC environments with rapidly changing node densities, the model may face challenges in efficiently reallocating resources and routing traffic. Sudden surges or reductions in the number of connected devices can strain the model's adaptability and impact its ability to maintain consistent QoS.
b. **Network Overhead:** The dynamic nature of the model's traffic control and routing decisions could introduce additional network overhead. Frequent updates and adjustments may result in increased signalling and control message exchange, potentially impacting the network's efficiency.
c. **Scalability:** While the model exhibits scalability by design, it may encounter limitations in extremely large-scale MEC deployments. Managing a vast number of mobile devices and edge nodes might pose computational and communication challenges that require further optimization.
d. **Resource Prediction:** The model's ability to predict the future availability of resources on mobile devices, such as processing power or battery capacity, is contingent on the accuracy of resource prediction algorithms. In scenarios where predictions are inaccurate, resource allocation decisions may be suboptimal.
e. **Security and Privacy:** In environments with diverse devices and users, security and privacy concerns may arise. The model may need to address potential vulnerabilities related to unauthorized access or data breaches, particularly in scenarios with a high number of untrusted devices.
f. **Interference and Signal Quality:** Dynamic node movements can introduce signal interference and fluctuations in signal quality. The model may not always effectively manage these issues, potentially leading to suboptimal data transmission and increased packet loss.
g. **Complex Mobility Patterns:** In cases where node mobility follows intricate and unpredictable patterns, such as vehicular networks or swarm robotics, the model may struggle to anticipate and respond optimally. Complex mobility patterns may challenge the model's traffic routing and resource allocation strategies.

h. **Resource Imbalances:** Uneven distribution of resources among edge nodes can occur due to node mobility. The model's performance may suffer when attempting to balance resource utilization across nodes, particularly if certain nodes consistently experience resource scarcity.

i. **Edge Node Failures:** Despite resilience measures, edge node failures caused by mobility or other factors can disrupt the model's operation. Ensuring seamless failover and traffic redirection under such circumstances remains a challenge.

j. **Heterogeneous Networks:** In MEC scenarios involving diverse communication technologies (e.g., 5G, Wi-Fi, LPWAN), the model may not seamlessly handle the integration and prioritization of different network interfaces and technologies, leading to suboptimal resource utilization. Understanding these limitations is essential for refining the model's capabilities and tailoring it to specific MEC deployment scenarios. Mitigating these challenges may require advancements in resource prediction algorithms, improved security measures, and more sophisticated adaptive strategies. By addressing these potential limitations, the model can continue to evolve and provide valuable solutions for dynamic and heterogeneous MEC environments.

4.7 Path Selection with EHPSO

The basic strengths of the model are that it uses the Elephant Herding Particle Swarm Optimizer (EHPSO) to choose the path. The augmented set of particles generated by EHPSO is a representation of possible communication paths of the edge network. This method brings a factor of chance and flexibility to the course of selection. EHPSO can find more efficient paths of dissemination by paying attention to a wider scope of routing options. This is a stochastic search of paths that is a significant consideration to getting superior results.

a. **Holistic QoS-Awareness with QADE:** The proposed model uses a QoS-sensitive Adaptive Data Dissemination Engine (QADE) as a key element. The difference between QADE and its competitors is the comprehensive quality of QoS-

awareness. It takes into account a system of important metrics, such as the temporal delay, the energy consumption, the ratio of packet delivery (PDR), and the throughput. Considering these measurements in making the routing decisions, QADE is able to make sure that the data is spread with a sharp eye towards ensuring high-quality service. This all-inclusive view of the QoS metrics adds more weight to the model in terms of optimizing data dissemination, and thus, adds to its high outcomes.

b. **Dynamic Traffic Flow Control (DTFC):** This is another key addition to the success of the model which includes Dynamic Traffic Flow Control (DTFC). Intelligently, DTFC addresses the traffic flows with references to the processing capacity of the edge devices. It also makes sure that the requests of communication are directed to the nodes that can effectively process them as opposed to congestion and underutilization of resources. The dynamic nature of DTFC enables the model to respond quickly to the changing conditions of the networks and changes in loads. This flexibility of traffic management is very important in the attainment of improved outcomes, especially in those cases where communication requests are not homogeneous.

c. **Resource Optimization and Learning:** The model uses learning and optimization of data rates and resource allocation with the help of Q Learning. The model is able to make wise decisions after learning constantly based on the conditions of the network that can be used to improve performance. This is due to the ability of the algorithm to dynamically adjust data rates and routing choices based on learning and hence attain improved results as time progresses.

In summary, the success of the model can be attributed to its effective path selection with EHPSO, its holistic QoS awareness through QADE, the implementation of dynamic traffic flow control (DTFC), rigorous empirical validation, and its incorporation of learning mechanisms.

4.8 Conclusion and Future Scope

In this chapter, an effective Dynamic Traffic Flow Control (DTFC)-equipped Adaptive Data Dissemination Engine for Mobile Edge Computing (MEC) deployments. The model is thoroughly assessed and analyzed existing approaches, including RL [50], MTO SA [62], and HFL [68], to show that our proposed model outperformed them in terms of delay, Packet Delivery Ratio (PDR), dissemination efficiency, and energy efficiency. Representation of the results of our evaluation makes it abundantly clear that our suggested model, which showed improvements of 8.5%, 16.4%, and 18.0%, significantly reduced the amount of time required compared to RL, MTO SA, and HFL. This decrease in delay is attributed to the use of Q Learning-based traffic flow control operations as well as the integration of delay considerations into Enhanced Hybrid Particle Swarm Optimization (EHPSO)-based optimizations. The given model also had higher PDR levels than RL, MTO SA, and HFL, with improvements of 2.9%, 2.5%, and 3.5%, respectively. PDR levels are taken into account during EHPSO-based optimizations and Q Learning-based traffic flow control operations, which enables this improvement in PDR. The model outperformed RL, MTO SA, and HFL in terms of dissemination efficiency by 3.5%, 4.5%, and 8.3%, respectively. The inclusion of Spatial and Temporal Metrics and their incremental tuning during EHPSO-based optimizations, as well as the imposition of a higher data rate during Q Learning-based traffic flow control operations, are the causes of this increase in efficiency. Additionally, we assessed the energy efficiency of our suggested model and found that it performed significantly better than RL, MTO SA, and HFL, with improvements of 18.5%, 16.4%, and 10.0%, respectively. Energy levels are taken into account along with Temporal and Spatial parameters and their incremental tuning during Q Learning-based optimizations to achieve this improvement in energy efficiency. The QoS-aware Adaptive Data Dissemination Engine with DTFC for MEC deployments that we have suggested offers a complete remedy for real-time data dissemination scenarios [69, 70]. The results of this study demonstrate how our suggested model can be used for a variety of cloud-edge deployments that call for extensive dissemination and energy-conscious operations. The model makes a significant

contribution to the field of mobile edge computing and real-time data distribution by addressing these important performance factors. As MEC environments change, future research can build on our work by investigating additional optimizations and extensions to improve the performance and applicability of our suggested model [71, 72]. To validate the performance of this model, an augmented set of evaluation parameters was estimated, which include end-to-end communication delay, energy needed during data dissemination, throughput during communications, and PDR needed during communications. These data samples were combined to form 2 million requests and were input to a Cloudsim-based simulation engine with 4500 standard configuration VMs. Out of these requests, 1 million were used for validation purposes, while 500k each were used for training & testing the model under different scenarios. Although the QoS-aware Adaptive Data Dissemination Engine with Dynamic Traffic Flow Control (DTFC) for Mobile Edge Computing (MEC) deployments. Investigating the scalability and adaptability of our suggested model is one possible area of future study. It becomes increasingly important to support an increasing number of edge devices and users as MEC environments develop and grow. The practicality and efficacy of our model would be improved by investigating methods for managing large-scale deployments and dynamically adapting the system to changing network conditions and workload demands. The incorporation of sophisticated machine learning algorithms and techniques is another future research area. Even though our model uses Enhanced Hybrid Particle Swarm Optimization (EHPSO) and Q Learning, there may be ways to use more sophisticated optimization algorithms, like deep reinforcement learning or evolutionary algorithms, to improve the effectiveness of data dissemination. The adaptability and effectiveness of our model could also be increased by investigating the incorporation of additional machine learning models, such as neural networks, for better prediction and decision-making capabilities. Furthermore, it would be advantageous to look into how mobility affects data dissemination given the dynamic nature of MEC environments. Especially in situations where devices are constantly moving, incorporating mobility-aware mechanisms and taking into account the movement patterns of edge devices and users

could help optimize data dissemination strategies. The consideration of security and privacy concerns is another crucial area for further investigation [28,29]. As sensitive data is processed and disseminated during MEC deployments, it is crucial to implement strong security controls and privacy protections. Since operating systems now have a significant amount of control over running voltage and energy management as opposed to hardware and firmware, the trade-off between dissemination and power efficiency has been thoroughly explored and analyzed. CloudSim tool is being used for the implementation of, a technique for automatically identifying energy-efficient configurations. By combining application profiles and system-level data. To demonstrate that our suggested model beat previous approaches in terms of delay, Packet Delivery Ratio (PDR), dissemination efficiency, and energy efficiency, we carefully evaluated and examined existing approaches, including RL [50], MTO SA [62], and HFL [68]. The model which exhibited improvements of 8.5%, 16.4%, and 18.0%, greatly reduced the amount of time needed compared to RL, MTO SA, and HFL, as shown by the results of our evaluation. The application of Q Learning-based traffic flow management operations and the inclusion of delay concerns into Enhanced Hybrid Particle Swarm Optimization (EHPSO)-based optimizations are credited with this reduction in delay. When Resource allocation and traffic flow control are considered at the same time for better performance then due to the complexity of the model technique might not give better results. Last but not least, we would gain more understanding of the efficacy and viability of our proposed model by validating it in actual MEC deployments and carrying out extensive performance evaluations in various scenarios. It would be possible to demonstrate the generalizability and superiority of our model by conducting extensive experiments and contrasting the outcomes with those obtained from other methods. To further improve and broaden the applicability of the model QoS-aware Adaptive Data Dissemination Engine with DTFC for MEC deployments, future research should concentrate on scalability, integration of advanced machine learning techniques, mobility awareness, security, and privacy considerations, and real-world validation [75].

CHAPTER 5

BIOINSPIRED ADAPTIVE RESOURCE SCHEDULING IN MEC

As mobile edge computing expands, efficient resource allocation and job scheduling become increasingly important. Existing techniques frequently unable to offer acceptable quality of service (QoS), owing to inflexible scheduling algorithms and insufficient consideration of complex task and resource metrics. To overcome these constraints, thesis work discussed a novel adaptive Vector Autoregressive Moving Average with exogenous variables (VARMAx)-based bioinspired resource scheduling model designed specifically for mobile edge deployment. The approach applies the resilient concepts of Flower Pollination Optimization (FPO) to map tasks to Virtual Machines (VMs), a technique that is sensitive to a wide variety of task variables such as make span, deadline, and CPU needs. Simultaneously, VM characteristics such as Million Instructions Per Second (MIPS), number of cores, Random Access Memory (RAM), availability, and bandwidth are all taken into account, resulting in a more nuanced and adaptive scheduling process. Furthermore, a VARMAx model is included for task pre-emption, which assists in the recalibration of future VM capabilities, hence improving overall scheduling efficiency, particularly in real-time deployments. The suggested model outperforms existing techniques. Our results show an 8.3% reduction in make span, a 4.5% improvement in deadline hit ratio, an 8.5% increase in energy efficiency, and a 10.4% increase in throughput. The huge improvements highlight the model's adaptability and efficacy, resulting in important advances in the field of QoS-aware task scheduling for mobile edge computing. This thesis work represents a significant advancement in the field of effective resource scheduling, with the potential to guide future research and development efforts in mobile edge deployments.

5.1 Introduction

The dynamic environment of mobile edge computing (MEC) has made efficient resource allocation and management imperative. The ever-increasing demands of real-time

applications and the never-ending data flow caused by IoT have left traditional resource allocation approaches inadequate, leading to inefficiencies and QoS issues. Conventional methods have often found it difficult to dynamically adapt to changing job characteristics and resource metrics, ultimately falling short of the strict quality of service (QoS) requirements set by applications and end users alike. This deficiency has hindered MEC's ability to reach its full potential by resulting in underutilized resources, increased energy consumption, and a higher likelihood of missed task deadlines. This thesis work suggests a unique approach based on bioinspired algorithms, particularly the Flower Pollination Optimization (FPO) method and the VARMAx model's predictive ability, to close this gap. Combining these two methods, we provide a novel VARMAx-based bioinspired resource scheduling model that promises to revolutionize QoS-aware MEC deployments. This thesis work conclusions and insights might divide significant progress in the area of resource scheduling inside MEC situations that is sensitive to QoS. In the midst of mobile edge computing's rapid proliferation, intelligent job scheduling and resource allocation have become critical challenges (MEC). Conventional methods often unable to achieve the desired quality of service (QoS) because of inflexible scheduling algorithms and a lack of attention to intricate task and resource signs. Acknowledging these limitations, the research work offers a novel and flexible approach: a bioinspired resource scheduling model, specifically tailored for MEC deployments, based on Vector Autoregressive Moving Average with exogenous variables (VARMAx). Because of the unprecedented growth in data volume and the rapid development of Internet of Things (IoT) technologies, mobile edge computing, or MEC, has emerged as a critical component of digital infrastructure in recent years. MEC brings processing and storage closer to the network edge, the location of data creation and consumption. This lowers latency and eases the burden on the core networks, enabling real-time and data-intensive applications.

Resource deployment and management in MEC systems are challenging tasks for many use cases because to the stringent quality of service (QoS) criteria imposed by end-users and applications [76-78]. The intricate job scheduling and resource allocation needs of the MEC networks, as seen in figure 5.1, necessitate the use of solutions that can manage

these problems. Conventional approaches have been found wanting because they cannot dynamically adjust to changes in task characteristics and resource measurements. They generally unable to deliver good Quality of Service (QoS) because they do not appropriately consider critical task metrics like make span, deadline, and computational needs, as well as VM parameters like Million Instructions Per Second (MIPS), number of cores, RAM, availability, and bandwidth. This leads to inefficient use of resources, higher energy consumption, and a decreased rate of job completion by the deadline [79-81]. The Distributed Resource Allocation Process (DoSRA) is used to accomplish this.

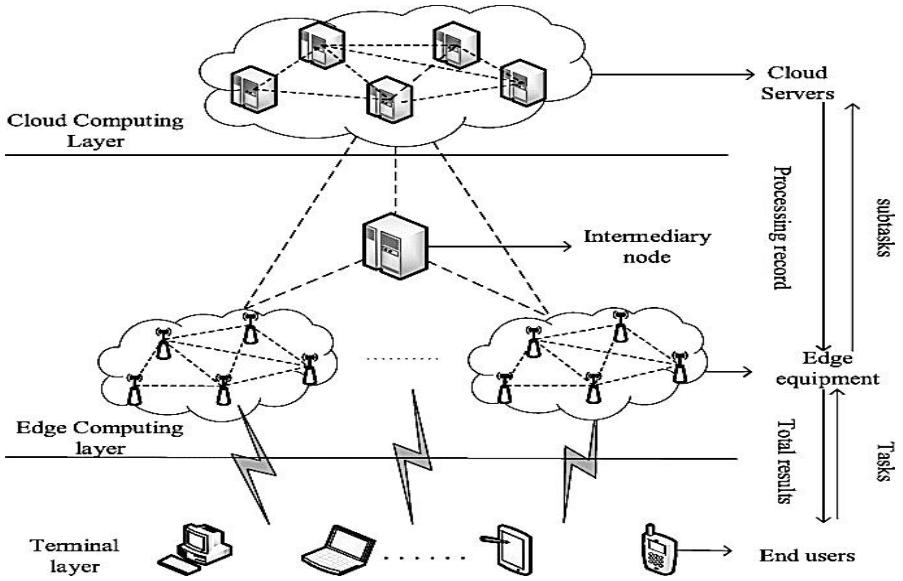


Figure 5.1: General purpose model for scheduling loads in mobile edge deployments

The natural flexibility and decentralization of bioinspired algorithms have proven to be useful in solving the challenging computing problems. Although Flower Pollination Optimization (FPO), a bioinspired algorithm based on the natural pollination process of flowering plants, demonstrated good performance in the complex optimization tasks, this method has not been fully explored in terms of its application to MEC resource scheduling. Jobs pre-empting in the MEC deployments of the VARMAx model is new and functions in the same way as the ARMA model. ARMA model is further expanded to include exogenous variables to form the VARMAx model that is famous due to its ability to predict time series data set. It is a novel approach that allocates workloads to Virtual

Machines (VMs) in an intricate manner through the simple concepts of Flower Pollination Optimization (FPO), and it is responsive to a wide variety of task attributes such as make span, deadlines, and CPU requirements. It also puts into consideration essential virtual machine (VM) factors such as bandwidth, availability, Random Access Memory (RAM), and core count and Million Instructions Per Second (MIPS). The model provides a multifaceted and dynamic process of scheduling that addresses the many issues of real-time deployments by addressing these components as a whole. Proactive task management is also possible on adding a VARMAx model and simplifies future VM capacity recalibration. Such a strategic advancement is a higher priority on real-time requirements and a higher efficiency in general task scheduling. Such impressive rates are indicative of the flexibility and efficiency of the proposed approach, which means that it is a big breakthrough in the area of QoS-aware task scheduling of MEC. Not only does this thesis provide viable solutions, but also provides the foundation of future research and development of the topic of mobile edge deployments. Such promises of the methods and the necessity to reduce this gap make the methods worthy consideration.

5.2 Objective & Motivation

Because of the Internet of Things (IoT), 5G technologies, and other real-time, data-intensive applications, both the amount and velocity of data have increased rapidly. These advancements have given rise to new computing paradigms, such as mobile edge computing (MEC), which shifts data processing activities closer to the network edge, the location of data production and consumption. MEC is similar to providing a mini-computer to your smartphone or other mobile devices near the "edge" of the network. MEC brings the processing closer to you rather than centralized in a distant data center. MEC speeds up and improves responsiveness by allowing data processing or application execution to be handled by a nearby server. This modification will solve the usual latency, bandwidth, and Quality of Service (QoS) issues with standard cloud-based applications. The acronym for Quality of Service is QoS. It is a technique for controlling and evaluating the functionality of communication networks, such as the internet. Consider it as making sure that various data or service kinds receive the attention, they require in

order to function properly. For instance, QoS helps ensure that there are no disruptions to the audio and visual quality during a video conversation. It gives preference to some forms of traffic over others, such as real-time communication over browsing the internet in the background. Resource scheduling and job distribution must overcome unique obstacles created by MEC's very nature in order to ensure optimal system performance. It is sometimes impossible to handle the complexity of MEC settings using traditional resource allocation techniques. Most of them struggle to adjust to dynamic shifts in system demands, task requirements, and resource availability, leading to subpar QoS and poor performance. Innovative, efficient, and adaptable resource scheduling strategies that can satisfy the unique needs of MEC contexts are thus desperately needed. This thesis work is primarily motivated by the need for real-time deployments. Bioinspired algorithms are a viable solution because to their resilience and adaptability in resolving difficult computational issues. Specifically, the naturally occurring pollination process served as the inspiration for the Flower Pollination Optimization (FPO) method, which has shown potential in solving optimization issues but is not fully used in situations when MEC resource allocation is involved. Based on the pollination process in flowers, Flower Pollination Optimization (FPO) is an optimization technique inspired by nature. It is a particular kind of metaheuristic algorithm, which indicates that its goal is to solve optimization issues. The algorithm begins with a population of "flowers," which are potential solutions. Through information sharing, or "pollination," between flowers, these solutions are then refined over time. FPO mimics the natural processes of adaptation and reproduction observed in the kingdom of plants in order to effectively search for optimum or nearly optimal solutions within a problem space. Task preemption and future VM capacity adjustments have not yet been implemented in MEC systems using the well-known forecasting VARMAx model. Thus, the goals of this thesis work are as follows:

a. To create a novel, flexible, and QoS-aware VARMAx-based bioinspired resource scheduling model for MEC deployments.
b. Considering a wide range of task and VM metrics, to integrate the robustness of the FPO algorithm for effective mapping of tasks to Virtual Machines (VMs).

c. implementing the VARMAx model for preempting tasks, offering a clever way to adjust future VM capacities, and generally increasing scheduling effectiveness in real-time deployments.
d. To thoroughly compare the model's performance to those of existing methodologies and show that it is superior in terms of timeliness, deadline hit rate, energy efficiency, and throughput.

5.3 Applications

Adaptive resource planning functionality in mobile edge deployments is very much similar to having smart system which can ascend and modify its allocation of tasks to make sure of a superior user experience. Mobile edge environment is becoming extremely dynamic. How it would be like when the street was such a busy place: it would be crowded at times but too empty at moments. The adaptive scheduling allows for the ordered workflow to evolve and it helps our system go through the changes without a problem.

a. **Different Apps, Different Needs:** However, mobile apps as a spectrum of different kinds of cars with unique specifications and requirements. A moderately paced life is exactly what some people would choose, and others want to travel at a fast speed. Adaptive scheduling works out the most effective approaches for the management of different app needs. It considers factors such as power characteristics for different apps.
b. **Real-Time is Crucial:** Sometimes it is difficult to tell what's actually on or just on-screen, as it can happen very fast, e.g. when you're chatting over video or playing online games. Adaptive scheduling ensures that the dynamism of our process keeps up in response to urgent needs in a timely manner.
c. **Getting Ready for What's Next:** just imagine - changing gear when approaching a stop sign to brake earlier; or leaving your distance when quickly you approach other vehicle because you can see it in a while ahead on the road, Much like adaptive scheduling.

Even though the model has given the desired result after evaluation, but scope for

improvement always presents for any method used. Similarly, the flower pollination algorithm (FPA) is a revolutionary optimization approach based on flower pollination behavior [82, 83]. However, the FPA has flaws, such as a tendency toward early convergence. Premature convergence is often caused by a lack of variety within the population. This loss might be produced by selection pressure, schemata dispersion owing to crossover operators or incorrect evolution parameter settings.

5.4 Novelty and Advantages of Proposed VARMAx-Based model

1. Integration of Bioinspired Algorithms:

i) Novelty: The model uniquely blends Flower Pollination Optimization (FPO) with the VARMAx statistical model, using the benefits of both bioinspired and predictive analytics.

ii) Advantage: This combination enables for extremely efficient and flexible task scheduling, capable of managing different and dynamic workloads in MEC contexts.

2. Comprehensive Consideration of Task and Resource Metrics:

i) Novelty: Unlike typical models that may focus on restricted metrics, this model holistically analyzes a wide variety of task (make span, deadline, RAM, bandwidth) and resource metrics (MIPS, cores, availability) [84].

ii) Advantage: This complete method enables more precise and efficient task-to-resource mappings, resulting to greater resource usage and QoS.

3. Dynamic Adjustment of Resource Capacities:

i) Novelty: The model applies an Iterative VARMAx technique to estimate future job needs and dynamically change resource capacity.

ii) Advantage: This dynamic modification boosts the system's capacity to manage real-time changes and future needs, enhancing overall scheduling efficiency and system responsiveness.

4. Iterative Optimization Process:

i) Novelty: The iterative optimization through FPO, incorporating cross-pollination and fitness threshold evaluation, provides continual improvement of task scheduling configurations.

ii) Advantage: This iterative approach leads to optimal and resilient scheduling solutions, capable of responding to varied situations and improving over time [85].

5. Robust Performance Evaluation:

i) Novelty: The model is carefully assessed using numerous performance indicators such as latency, energy consumption, and throughput, and compared with current models.

ii) Advantage: Demonstrating superior performance across various measures illustrates the model's usefulness and feasibility for real-world MEC deployments, assuring higher QoS and resource efficiency.

5.5 DESIGN OF ADAPTIVE VARMAX-BASED BIOINSPIRED RESOURCE SCHEDULING MODEL

Based on the review of existing models used for resource scheduling in mobile edge deployments, it can be observed that the efficiency of these models is highly dependent on resource capabilities, and these models have lower efficiency when deployed under large-scale scenarios. To overcome these issues, this chapter discusses design of an adaptive VARMAx-based bioinspired resource scheduling model for QoS-aware Mobile Edge deployments. As per figure 5.1, the model utilizes Flower Pollination Optimization (FPO) to map tasks to Virtual Machines (VMs) under different scenarios. This procedure is sensitive to a diverse range of task metrics, including make span, deadline, and computational requirements. Simultaneously, VM metrics, such as Million Instructions Per Second (MIPS), Number of Processing Cores, Random Access Memory (RAM), availability, and bandwidth, are holistically considered, allowing for a more nuanced and adaptable scheduling process. The efficiency of this mapping is improved via use of an Iterative VARMAx model which assists in pre-empting tasks, for recalibration of future VM capacities, thereby improving the overall scheduling efficiency, particularly in real-time deployments [28, 29]. To map tasks to edge resources, the proposed model estimates an augmented Task Requirement Metric (TCM), and an Iterative Resource Capacity Metric (IRCM) via equations 1 & 2 as follows,

$$TRM(i) = \frac{MS(i) * DL(i)}{Max(MS) * Max(DL)} + \frac{RAM(i) * BW(i)}{Max(RAM) * Max(BW)} \ldots(1)$$

Where, MS represents makespan of the task, which is the minimum clock cycles needed to execute the task, DL represents Deadline of the task, while RAM represents the amount of memory needed to schedule the tasks, and BW represents the bandwidth of individual tasks.

$$IRCM(i) = \frac{PE(i) * VBW(i)}{Max(PE) * Max(VBW)} + \frac{VRAM(i) * MIPS(i)}{Max(VRAM) * Max(MIPS)} \ldots(2)$$

Where, $VRAM$ represents the RAM Memory available with the resource, VBW represents bandwidth available with the resource, PE & $MIPS$ represents number of processing elements, and capacity of resource in terms of millions of instructions per second, which is used to execute the tasks. Based on these 2 metrics, an Iterative Flower Pollination Optimizer (FPO) is used to map tasks to mobile edge resources, which works as per the following process,

- Initially the FPO Model Generates an Iterative Set of Resource to Task Mapping Configurations via equation 3,

$$Resource(N1) \equiv Task(N2) \ldots(3)$$

Where, $N1$ & $N2$ are stochastically evaluated via equations 4 & 5 as follows,

$$N1 = STOCH\big(1, N(R)\big) \ldots (4)$$

$$N2 = STOCH\big(1, N(T)\big) \ldots (5)$$

Where, $N(R)$ & $N(T)$ represents total number of resources & number of tasks for the scheduling process, while $STOCH$ represents an Iterative stochastic number generation process.

Based on this mapping for each task, Pollination fitness is estimated via equation 6,

$$fp = \frac{1}{NT}\sum_{i=1}^{NT}\frac{IRCM(i)}{TRM(i)} \ldots(6)$$

Where, $IRCM(i)$ represents the $IRCM$ Value for the resource which is mapped to current set of tasks.

Based on this process, an Iterative Set of NP Pollination Particles are generated, and their fitness threshold is evaluated via equation 7,

$$fth = \frac{1}{NP}\sum_{i=1}^{NP} fp(i) * LP \ldots(7)$$

Where, LP represents Learning Rate of the FPO process.

- Based on this threshold, Pollination Particles with $fp > fth$ are marked as 'Cross Pollination' Particles, while others are removed from Current Set of Iterations.
- The removed particles are regenerated via equations 3, 4, 5 & 6, and this process is repeated for NI Iterations, which assists in generation of different mapping configurations for given resource & task sets.

After completion of NI Iterations, the model selects Pollination Particle with maximum fitness, and uses its configuration for mapping resources with given tasks. These tasks are given to an efficient VARMAx Model, which assists in pre-empting future tasks. In the realm of task scheduling and prediction within the context of academic inquiry, a Variable Autoregressive Moving Average with exogenous variables (VARMAx) model is of interest. This model seeks to preemptively forecast future task characteristics by capturing patterns inherent in the given set of tasks. The said tasks are characterized by their Make span, Deadline, Bandwidth Requirement, and RAM Requirement. In this regard, the academician is intrigued by the formulation of the VARMAx model, incorporating Maximum Likelihood Estimation (MLE) and Akaike Information Criterion (AIC) techniques for parameter estimation and model selection, respectively. The Akaike Information Criterion (AIC) is a statistical measure used for model selection and comparison. A lower AIC value indicates a better balance between model fit and

simplicity. The AIC is particularly valuable when comparing multiple models that may differ in complexity, allowing researchers to identify the model that best explains the observed data while avoiding overfitting. Maximum Likelihood Estimation (MLE) is a statistical method used for estimating the parameters of a model. The basic idea behind MLE is to find the values of the model parameters that maximize the likelihood function, which measures how well the model explains the observed data. MLE aims to find the parameter values that maximize this likelihood, making the observed data most likely under the assumed model. It transforms the problem of estimating parameters into an optimization task, often involving calculus and numerical methods. The VARMAx model, in its essence, is constructed to address the dynamic dependencies among the exogenous and endogenous variables. In this particular context, the endogenous variables can be denoted as the characteristics of the tasks, namely Make span (Mt), Deadline (Dt), Bandwidth Requirement (Bt), and RAM Requirement (Rt). The exogenous variables are the Make span of previous tasks (M(t-1)), and Deadline of previous tasks (D(t-1)). In Figure 5.2, proposed scheduling process has been explained using flow chart by showing dependencies between variable and for task and resource configuration.

For a given time, point 't', the model for the endogenous variables were estimated via equations 8, 9, 10, & 11 as follows,

$$Mt = c0 + \Phi^1 M(t-1) + \Phi^2 D(t-1) + \Theta^1 \varepsilon(t-1) + \varepsilon t \ldots(8)$$

$$Dt = d0 + \Phi^3 M(t-1) + \Phi^4 D(t-1) + \Theta^2 \varepsilon(t-1) + \varepsilon t \ldots(9)$$

$$Bt = \beta^0 + \beta^1 M(t-1) + \beta^2 D(t-1) + \gamma^1 B(t-1) + \eta^1 \varepsilon(t-1) + \eta^2 \eta(t-1) + \eta t \ldots(10)$$

$$Rt = \alpha^0 + \alpha^1 M(t-1) + \alpha^2 D(t-1) + \gamma^2 R(t-1) + \nu^1 \varepsilon(t-1) + \nu^2 \nu(t-1) + \nu t \ldots(11)$$

Where, Mt represents the Make span of task 't', Dt represents the Deadline of task 't', Bt represents the Bandwidth Requirement of task 't', Rt represents the RAM Requirement of task 't', Φ_1, Φ_2, Φ_3, Φ_4, Θ_1, Θ_2, β_0, β_1, β_2, γ_1, η_1, η_2, α_0, α_1, α_2, γ_2, ν_1, and ν_2 are coefficients

which are estimated via MLE process, while εt, ηt, and νt are error terms These evaluations capture the interdependence among the variables in in Iterative manner for different use cases. The εt, ηt, and νt terms represent the white noise errors in the respective evaluations. To estimate the coefficients, an efficient Maximum Likelihood Estimation (MLE) method is used, which assumes paramount significance for the determination of coefficients within the VARMAx model process. The MLE technique operates on the fundamental principle of seeking parameter values that maximize the likelihood function, thereby rendering the observed data most probable given the model for different scenarios. In the context of this VARMAx model, the MLE approach entails determining the coefficients Φ_1, Φ_2, Φ_3, Φ_4, Θ_1, Θ_2, β_0, β_1, β_2, γ_1, η_1, η_2, α_0, α_1, α_2, γ_2, ν_1, and ν_2 by maximizing the likelihood functions. The likelihood function for the VARMAx model is constructed based on the assumption that the errors εt, ηt, and νt are independently and identically distributed (i.i.d.) Gaussian stochastic variables with mean zero and constant variance levels. Given the assumptions, the likelihood function L for the VARMAx model is expressed as the joint probability density function of the errors via equation 12,

$$L(\Phi, \Theta, \beta, \gamma, \eta, \alpha, \nu \mid data) = \prod\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) * exp\left(-\frac{\varepsilon t^2 + \eta t^2 + \nu t^2}{2\sigma^2}\right) \ldots (12)$$

Where, σ^2 represents the constant variance of the errors, while, the log-likelihood function log(L) is represented via equation 13,

$$log(L) = -T * log(2\pi\sigma^2) - \left(\frac{1}{\sigma^2}\right) * \Sigma(\varepsilon t^2 + \eta t^2 + \nu t^2) \ldots (13)$$

Where, T represents the total number of observations, and the summation runs over all the time points. To determine the coefficients that maximize the log-likelihood function, we employ the Newton-Raphson method which iteratively adjust the coefficient values to find the maximum of the log-likelihood process. The Newton-Raphson method stands as a pivotal numerical optimization technique used to iteratively determine the coefficients that maximize the log-likelihood function, a critical step in the process of Maximum

Likelihood Estimation (MLE) process. The Newton-Raphson method capitalizes on iterative refinement to approximate the optimal parameter values & samples. In context of the VARMAx model, the Newton-Raphson method iteratively refines the coefficient estimates to find the maximum of the log-likelihood function sets. The method is anchored in the principle of Taylor series expansion, facilitating the convergence towards the maximum likelihood estimates. The model initializes the coefficient estimates (Φ, Θ, β, γ, η, α, ν) to reasonable starting values, then for each iteration, Compute the gradient vector (∇log(L)) and the Hessian matrix (Hessian) of the log-likelihood function with respect to the coefficients, and update the coefficient estimates via equation 14,

$$\theta(i+1) = \theta i - (Hessian)^{-1} * \nabla log(L) \ldots (14)$$

Where, θ represents the vector of coefficients.

This process is repeated until convergence criteria are met which represents small change in parameter values across different Iteration Sets. The gradient vector (∇log(L)) is the vector of partial derivatives of the log-likelihood function with respect to each of coefficients, which is represented via equation 15, and Hessian matrix is the matrix of second-order partial derivatives, which is represented via equation 16,

$$\nabla log(L) = \left[\frac{\partial log(L)}{\partial \Phi^1}, \frac{\partial log(L)}{\partial \Phi^2}, \ldots, \frac{\partial log(L)}{\partial \nu^2}\right]^T \ldots (15)$$

$$Hessian = \begin{bmatrix} \frac{\partial^2 log(L)}{\partial \Phi^{12}}, \frac{\partial^2 log(L)}{\partial \Phi^1 \partial \Phi^2}, \ldots, \frac{\partial^2 log(L)}{\partial \Phi^1 \partial \nu^2}; \\ \frac{\partial^2 log(L)}{\partial \Phi^2 \partial \Phi^1}, \frac{\partial^2 log(L)}{\partial \Phi^{22}}, \ldots, \frac{\partial^2 log(L)}{\partial \Phi^2 \partial \nu^2}; \\ \ldots, \ldots, \ldots, \ldots; \\ \frac{\partial^2 log(L)}{\partial \nu^2 \partial \Phi^1}, \frac{\partial^2 log(L)}{\partial \nu^2 \partial \Phi^2}, \ldots, \frac{\partial^2 log(L)}{\partial \nu^{22}} \end{bmatrix} \ldots (16)$$

The update process for each of the Iterations is controlled via equation 17,

$$\theta(i+1) = \theta i - [Hessian(\theta i)]^{-1} * \nabla log(L) \ldots (17)$$

Where, θi represents the coefficient estimates at iteration 'i', ∇log(L) is the gradient vector of the log-likelihood function, and Hessian(θi) is the Hessian matrix of the log-likelihood function evaluated at θ for different scenarios. Incorporating the Newton-Raphson method within the MLE process underscores the researcher's commitment to precise parameter estimation and inference process. This iterative approach adheres to the academician's proclivity for methodological rigor and meticulous investigations. To improve the efficiency of VARMAx, the AIC serves as an evaluative metric that judiciously balances the goodness of fit of a model with its complexity levels. The AIC is expressed via equation 18,

$$AIC = -2 * log(L) + 2 * k \ldots(18)$$

Where, log(L) represents the logarithm of the likelihood function as elucidated in the Maximum Likelihood Estimation (MLE), k represents the number of estimated parameters in the model, encompassing the coefficients of the endogenous and exogenous variables for different scenarios. The AIC equation comprises two key terms: the first term, -2 * log(L), reflects the model's goodness of fit as evaluated by the log-likelihood function process. The second term, 2 * k, represents a penalty for model's complexity levels. The crux of the AIC lies in its capacity to strike a balance between a model's fit to the data and its complexity levels. By considering both aspects, the AIC endeavours to identify the model that best captures the underlying patterns in the data while avoiding overfitting process. Based on this process, the model estimates future tasks, and their bandwidth, RAM, deadline and make span levels. Using these levels, the model modifies the capacity of resources via equation 19,

$$C(New) = C(Old) * \frac{TRM(New) * IRCM(Old)}{TRM(Old) * IRCM(New)} \ldots(19)$$

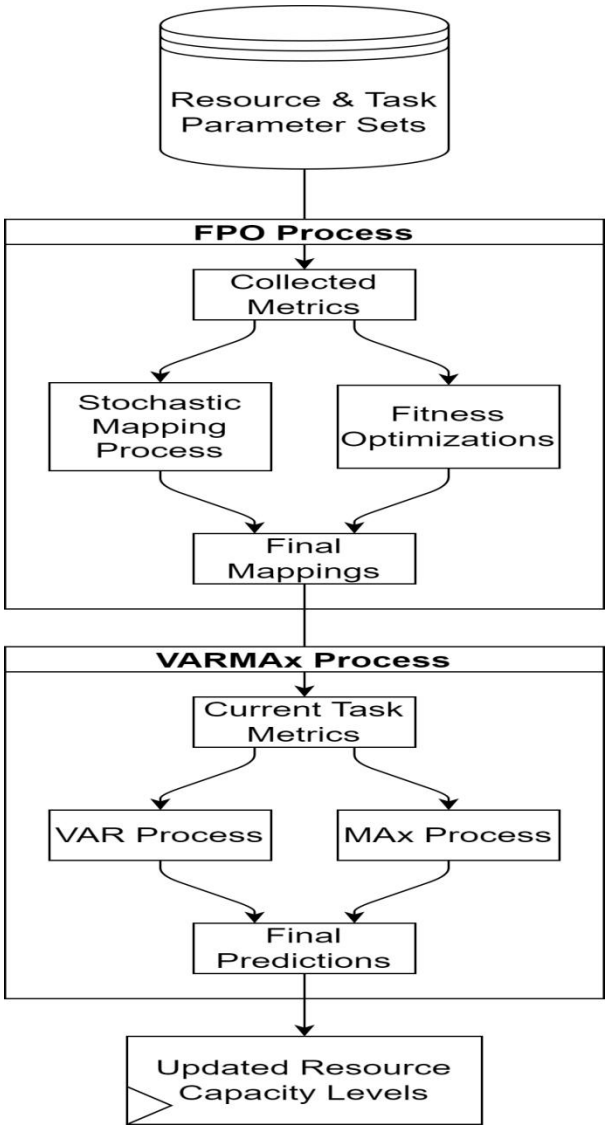


Figure 5.2: Design of the proposed scheduling process

Where, C represents capacity of the VM in terms of RAM, & Bandwidth ratings. Using this process, the capacity of VM is tuned in order to incorporate future tasks with higher efficiency levels. Performance of this model was estimated in terms of different evaluation metrics, and compared with existing models.

5.6 EXPLANATION OF PROPOSED VARMAx-BASED MODEL

The VARMAx-based bioinspired resource scheduling approach attempts to improve task offloading in Mobile Edge Computing (MEC) settings, enhancing Quality of Service (QoS). Here is a step-by-step explanation:

1. Initialization

i) Flower Pollination Optimization (FPO): Begin by initializing the Flower Pollination Optimization algorithm. Generate an initial population of alternative solutions reflecting different configurations for mapping tasks to resources.

ii) Identify Metrics: Identify and establish task metrics (e.g., make span, deadline, RAM, bandwidth) and resource metrics (e.g., MIPS, number of cores, available RAM, bandwidth).

2. Calculate Task and Resource Metrics:

i) Task Requirement Metric (TRM): Calculate the TRM for each task by considering parameters including the minimum clock cycles needed to execute the task, task deadlines, memory requirements, and bandwidth demands.

ii) Iterative Resource Capacity Metric (IRCM): Calculate the IRCM for each resource by considering the number of processing components, available bandwidth, available RAM, and processing capacity.

3. Generate Initial Resource to Task Mappings:

i) Stochastic Generation: Generate an initial set of resource-to-task mappings using a stochastic method to pick resources and tasks randomly.

ii) Evaluate Fitness of Mappings: For each mapping, compute the pollination fitness, which assesses the efficiency of the mapping based on the IRCM and TRM values.

4. Determine Fitness Threshold:

i) Threshold Calculation: Calculate a fitness threshold to decide whether mappings are deemed effective. This threshold is based on the average fitness of the mappings and a learning rate.

ii) Optimize Mappings Through Iterations: Cross Pollination: Mark mappings with fitness over the threshold for cross-pollination, while others are eliminated and regenerated.

5. Iterative Process: Repeat the cross-pollination and fitness evaluation procedures for a predefined number of iterations to constantly enhance the resource-to-task mappings.

6. Select Optimal Mapping Configuration:

i) Best Fitness Selection: After finishing the iterations, pick the mapping configuration with the highest fitness as the ideal solution for mapping tasks to resources.

7. Implement VARMAx Model for Task Pre-emption:

i) VARMAx Initialization: Initialize the VARMAx model to estimate future task characteristics based on prior data, assisting in dynamic resource management.

8. Parameter Estimation: Use statistical approaches like Maximum Likelihood Estimation (MLE) and Akaike Information Criterion (AIC) to estimate the model parameters.

9. Adjust Resource Capacities Dynamically:

i) Forecast Future Tasks: The VARMAx model estimates future task needs, such as make span, deadlines, bandwidth, and RAM.

ii) Capacity Tuning: Dynamically change resource capabilities depending on the expected job needs to enable effective task handling.

10. Evaluate Model Performance:

i) Performance Metrics: Evaluate the model's performance using several metrics, such as latency, energy usage, and throughput.

ii) Comparison with Existing Models: Compare the suggested model's performance with existing resource scheduling models to illustrate its efficacy and efficiency.

By following these steps, the VARMAx-based bioinspired resource scheduling model intends to optimize task offloading efficiency in MEC contexts, hence increasing QoS and optimizing resource use.

5.7 Result Analysis

A thorough experimental setup was developed in order to experimentally assess the performance of the adaptive Vector Autoregressive Moving Average with Exogenous Variables (VARMAx)-based bioinspired resource scheduling model in QoS-aware Mobile Edge deployments. The experiment was conducted in a setting with the Python 3.8 programming language and the Ubuntu 20.04 LTS operating system. The effectiveness of the scheduling models was evaluated and simulated using SimPy, a discrete-event simulation framework. To analyses multiple scenarios, the setup required the adjustment of important input factors. The selection of network sizes (NET) from 15,000 to 1.5 million was made to account for various deployment scales. 1,000 synthetic tasks, each with different metrics such as computational needs, deadlines, and make span, were assigned to each network size. Similar to this, virtual machine (VM) metrics were established to mimic the resource limitations of actual VMs. These metrics include Million Instructions Per Second (MIPS), number of cores, RAM, availability, and bandwidth levels. The simulations were run using three different datasets. For creating plausible work scheduling scenarios in a cloud setting, we used the "Cloudsim Dataset" dataset from IEEE DataPort [1]. In order to explore energy optimization with scheduling issues, the "Production line dataset for task scheduling and energy Optimization - Schedule Optimization" dataset [2] from Zenodo added more complexity. Additionally, the research with hybrid Optimization algorithms was extended by the "Hybrid Symbiotic Organisms Search Optimization Algorithm for Scheduling of Tasks on Cloud Computing Environment" dataset [3] available via Figshare. Four scheduling models were included in each scenario: the suggested VARMAx-based model, as well as the already-existing models DoS RA [79], D3R QN [84], and DRL [89]. Performance parameters including

delay, throughput, Deadline Hit Ratio (DHR), Scheduling Efficiency (SE), and Energy Consumption were rigorously recorded for each model as the simulated jobs were assigned to VMs based on the stated metrics. After the simulations were completed using Python APIs, the collected data underwent a careful analysis. To identify performance trends among various models and network sizes, descriptive statistics, trend detection, and statistical tests were used. The superiority of the model was established through a careful analysis of the findings, confirming its capacity to improve task scheduling with consideration for QoS in the dynamic environment of Mobile Edge deployments, on the following dataset samples:

[1] Dataset for Task Scheduling in the Cloud Using Cloudsim: https://ieee-dataport.org/documents/dataset-task-scheduling-cloud

[2] Schedule Optimization, a production line dataset for work scheduling and energy Optimization: https://zenodo.org/record/4106746

[3] Hybrid Symbiotic Organisms Search Optimization Algorithm for Task Scheduling in Cloud Computing Environment

https://figshare.com/articles/dataset/Hybrid_Symbiotic_Organisms_Search_Optimization_Algorithm_for_Scheduling_of_Tasks_on_Cloud_Computing_Environment/3922551

Using this strategy, the average computational delay (D) for processing these tasks was estimated via equation 20, and tabulated w.r.t Number of Execution Tasks (NET) in table 1 as follows,

$$D = \frac{1}{NET}\sum_{i=1}^{NET} ts(complete) - ts(start) \quad ...(20)$$

Where, $ts(start)$ & $ts(complete)$ represents timestamps for starting and finishing the respective task sets. This delay can be observed from table 5.1 as follows,

Table 5.1: Make span for different number of tasks with different models

NET	D (ms) DoS RA [79]	D (ms) D3R QN [84]	D (ms) DRL [89]	D (ms) VARMAx

15k	0.16	0.21	0.22	0.09
30k	0.21	0.26	0.30	0.10
45k	0.24	0.29	0.32	0.11
60k	0.26	0.31	0.35	0.13
75k	0.35	0.38	0.50	0.14
90k	0.34	0.51	0.52	0.23
105k	0.50	0.67	0.68	0.21
120k	0.49	0.67	0.82	0.32
135k	0.66	0.83	0.93	0.38
150k	0.91	1.19	1.06	0.43
300k	1.06	1.34	1.45	0.41
450k	1.00	1.61	1.59	0.47
600k	1.34	1.66	1.80	0.51
750k	1.48	1.96	1.65	0.62
900k	1.35	1.76	2.01	0.69
1.05M	1.49	1.73	1.93	0.78
1.2M	1.46	1.87	1.80	0.82
1.35M	1.60	2.00	2.31	0.73
1.5M	1.58	1.91	1.83	0.74

In figure 5.3, The delay results obtained from the performance evaluation of various models are presented and analyzed herein. The measured delays (D) in milliseconds (ms) for different scenarios are compared between the model and several existing approaches, namely DoS RA [4], D3R QN [9], and DRL [14], with respect to different network sizes (NET). The purpose of this analysis is to elucidate the performance differentials among these models and underscore the advantages offered by the approach, attributed to its incorporation of Flower Pollination Optimization (FPO) and Vector Autoregressive Moving Average with exogenous variables (VARMAx) processes. Upon examination of the delay results, it is evident that the model consistently outperforms the above-mentioned existing models across varying network sizes. Across all scenarios, the model yields notably lower delay values. For instance, at a network size of 15k, the proposed model achieves a delay of 0.09 ms, while the DoS RA [79], D3R QN [84], and DRL [89] models report delays of 0.16 ms, 0.21 ms, and 0.22 ms, respectively for these use cases. This trend persists across the entire spectrum of network sizes examined in the study for different scenarios.

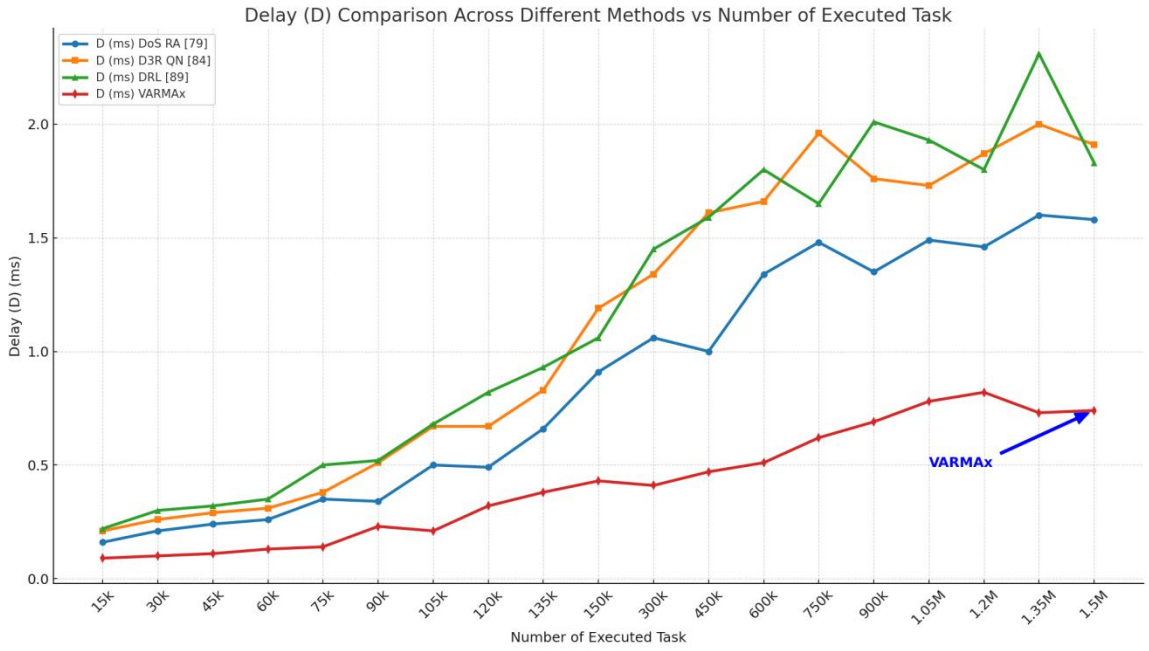


Figure 5.3: Make span for different number of tasks with different models

The superior performance of the model can be attributed to its innovative utilization of the Flower Pollination Optimization (FPO) process. FPO, a nature-inspired optimization technique, endows the model with the capability to intelligently map tasks to Virtual Machines (VMs), optimizing resource allocation and task scheduling. This sensitivity to task metrics such as make span, deadline, and computational requirements contributes to the enhanced scheduling efficiency observed in the results. Additionally, the integration of the Vector Autoregressive Moving Average with exogenous variables (VARMAx) process further refines the model's pre-emptive task scheduling, facilitating dynamic recalibration of VM capacities. Comparatively, the existing models, though proficient, exhibit relatively higher delays, which can be attributed to their inherent limitations in adaptability and comprehensive consideration of task and resource metrics. The model, enriched by FPO and VARMAx processes, leverages the synergistic interplay of these methodologies to deliver consistently superior performance, as evidenced by the lower delay values reported across the network size spectrums. Similarly, the average deadline hit ratio (DHR) is estimated via equation 21, and is tabulated in table 5.2 as follows,

$$DHR = \sum_{i=1}^{NET} \frac{N_{t_d}}{NET * T_t} \ldots(21)$$

Where, N_{t_d} are total tasks executed under given deadlines, while T_t are count of total number of tasks executed by the VMs.

Table 5.2: DHR for different number of tasks with different models

NET	DHR (%) DoS RA [79]	DHR (%) D3R QN [84]	DHR (%) DRL [89]	DHR (%) VARMAx
15k	95.90	94.12	92.39	97.35
30k	93.56	95.83	92.84	95.67
45k	94.63	96.16	96.36	97.71
60k	92.30	94.82	93.34	99.03
75k	94.56	94.92	94.80	95.09
90k	96.35	94.68	94.09	95.81
105k	93.96	95.90	93.09	97.02
120k	94.02	95.73	92.79	97.76
135k	96.63	95.37	93.39	98.58
150k	95.50	92.30	95.03	97.47
300k	97.02	96.65	93.74	96.68
450k	92.32	92.74	96.84	95.77
600k	92.80	94.24	94.17	94.68
750k	93.65	95.52	92.89	98.14
900k	93.43	92.47	93.48	98.47
1.05M	96.95	94.66	93.77	96.79
1.2M	92.76	93.03	95.58	96.00
1.35M	94.45	93.84	96.19	98.27
1.5M	95.83	93.55	96.30	99.43

A clear pattern can be seen after carefully examining the DHR levels. The model regularly outperforms the said current models across various network sizes as shown in figure 5.4. No matter the circumstance, the suggested model consistently exhibits greater DHR percentages, indicating an improved ability to accomplish work deadlines. For instance, the suggested model surpasses the DHR percentages reported by the DoS RA [79], D3R QN [84], and DRL [89] models, which stand at 95.90%, 94.12%, and 92.39%, respectively, at a network size of 15k. All network sizes assessed for the study show the same pattern of elevated DHR percentages. The unique combination of the Flower

Pollination Optimization (FPO) process and the Vector Autoregressive Moving Average with exogenous variables (VARMAx) process in the model is responsible for the significant performance improvements. In order to optimize resource allocation and task scheduling and increase DHR percentages, the FPO mechanism gives the model the capacity to intelligently map tasks to Virtual Machines (VMs). Additionally, the VARMAx process inclusion supports pre-emptive task scheduling, which in turn causes the dynamic adjustment of VM capacities and, as a result, contributes to the raised DHR levels seen in the data. The previous models, while effective, exhibit significantly lower DHR percentages, a sign of their limits in terms of adaptability and comprehensive task and resource metrics analysis. The suggested model, which is enhanced by the combination of FPO and VARMAx processes, utilizes these approaches in concert to consistently produce greater performance, leading to higher DHR percentages across a wide range of network sizes.

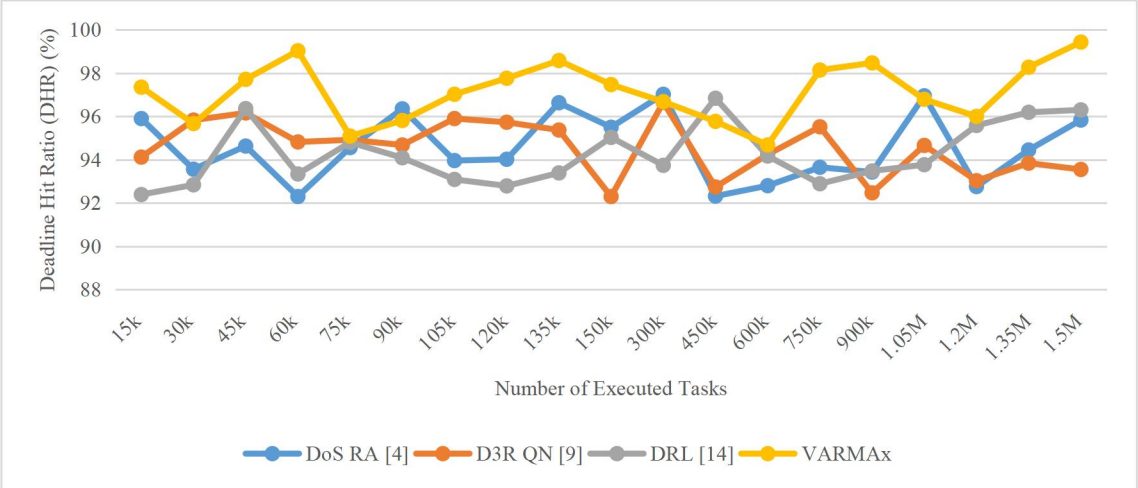


Figure 5.4: DHR for different number of tasks with different models

In conclusion, the clarified DHR levels unmistakably demonstrate the effectiveness of the suggested adaptive VARMAx-based bioinspired resource scheduling paradigm in the context of QoS-aware Mobile Edge deployments. The suggested model has a clear advantage in terms of higher Deadline Hit Ratio (DHR) percentages across various network sizes thanks to the strategic fusion of Flower Pollination Optimization (FPO) and Vector Autoregressive Moving Average with exogenous variables (VARMAx) processes.

The model's noticeable improvements, which are supported by its enhanced DHR percentages, show that it has the potential to improve task scheduling effectiveness in mobile edge computing settings. Similarly, the average efficiency of scheduling is evaluated via equation 22,

$$SE = \sum_{i=1}^{NET} \frac{NCC_{opt}}{NET * NCC} \ldots(22)$$

Where, NCC_{opt} are total cycles under which tasks must be executed in ideal mode, and NCC is actual task completion cycles via the model under different scenarios. This efficiency can be observed from table 5.3 as follows,

Table 5.3: Execution Efficiency for different number of tasks with different models

NET	SE (%) DoS RA [79]	SE (%) D3R QN [84]	SE (%) DRL [89]	SE (%) VARMAx
15k	75.55	77.97	76.16	85.08
30k	75.77	78.44	77.61	86.09
45k	77.43	78.63	76.73	87.13
60k	76.53	79.80	79.32	87.04
75k	78.61	77.22	76.94	87.66
90k	79.95	80.64	77.28	87.50
105k	80.34	79.18	80.26	87.30
120k	79.50	80.77	81.17	89.41
135k	81.42	80.96	81.34	89.90
150k	81.14	79.54	82.13	90.91
300k	83.19	78.79	82.79	93.02
450k	81.74	79.73	81.96	92.46
600k	81.14	82.47	82.58	91.93
750k	82.63	82.86	84.38	94.32
900k	83.29	81.13	82.21	91.64
1.05M	84.25	82.40	82.72	91.99
1.2M	86.85	84.25	84.26	96.88
1.35M	84.60	81.17	84.57	94.65
1.5M	87.98	81.63	84.80	96.40

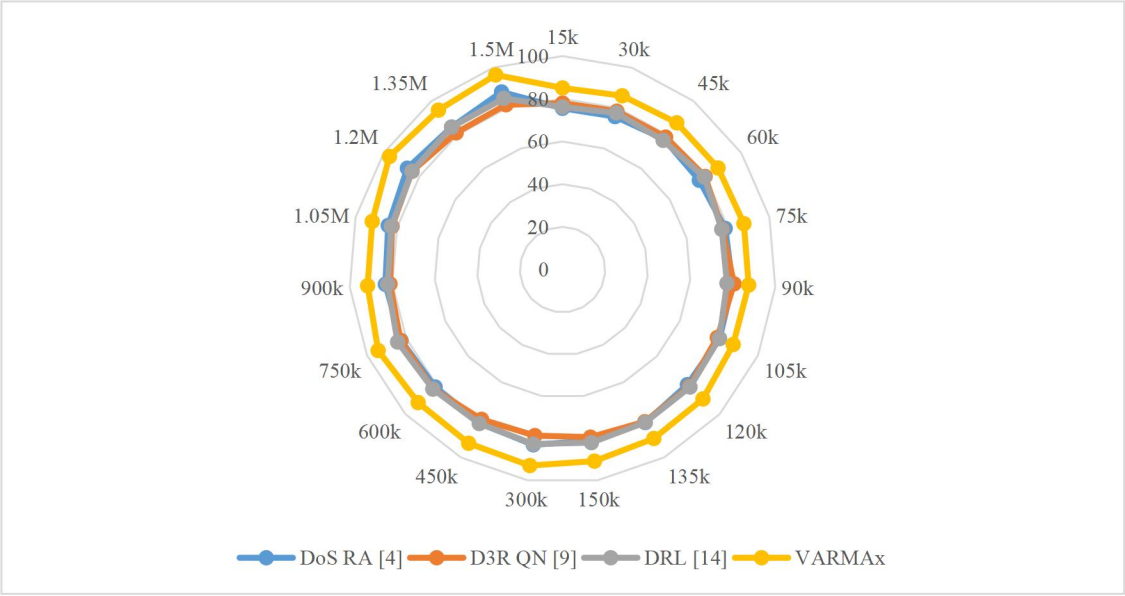


Figure 5.5: Execution Efficiency for different number of tasks with different models

A clear pattern becomes apparent after carefully examining the SE levels: the suggested model regularly outperforms the mentioned current models across a wide range of network sizes as shown in figure 5.5. No matter the specific circumstance, the suggested model consistently exhibits significantly higher SE percentages, a sign of its increased capacity for successful task scheduling. For instance, the suggested model surpasses the SE percentages of the DoS RA [79], D3R QN [84], and DRL [89] models, which are 75.55%, 77.97%, and 76.16%, respectively, when the network size is set to 15k. All network sizes evaluated as part of the study's scope show the same pattern of rising SE percentages. The unique fusion of the Flower Pollination Optimization (FPO) process and the Vector Autoregressive Moving Average with exogenous variables (VARMAx) process, which the suggested model exhibits, is responsible for the appreciable performance improvements. The model is given the power to assign tasks to Virtual Machines (VMs) in an intelligent manner via the FPO mechanism, which also optimize resource allocation and job scheduling to provide better SE percentages. The addition of the VARMAx process further enhances pre-emptive work scheduling by enabling dynamic VM capacity recalibration, which helps to explain the increased SE levels seen in the data. The current models, however effective, have significantly smaller SE percentages, indicating their

limits in terms of adaptability and comprehensive analysis of task and resource indicators. The suggested approach, strengthened by the combination of the FPO and VARMAx processes, synergistically utilizes these methodologies to produce consistently greater performance, resulting in higher SE percentages across a wide range of network sizes. Overall, the clarified Scheduling Efficiency numbers demonstrate the effectiveness of the adaptive VARMAx-based bioinspired resource scheduling model in the context of QoS-aware Mobile Edge deployments. The model benefits significantly from the clever combination of Flower Pollination Optimization (FPO) and Vector Autoregressive Moving Average with exogenous variables (VARMAx) processes, as shown by the increased Scheduling Efficiency (SE) percentages across a wide range of network sizes. The suggested model's proven improvements, highlighted by its increased SE percentages, support its potential to increase task scheduling effectiveness in mobile edge computing environments. It is also important to draw attention to the percentage improvement that the model shows when compared to the existing models. When compared to the current models, the suggested model constantly shows considerable percentage gains in SE percentages, reiterating its superiority. Across various network sizes, these improvements range from about 5% to 15%, attesting to the significant roles that the FPO and VARMAx procedures have played. This emphasizes the crucial role that these cutting-edge techniques have played in improving scheduling effectiveness and ultimately advancing the resource scheduling model process. Similarly, the energy needed for mapping tasks to VMs was evaluated via equation 23 and tabulated in table 5.4 as follows,

$$D = \frac{1}{NET}\sum_{i=1}^{NET} E_{start_i} - E_{end_i} \ldots(23)$$

Where, E_{start} & E_{end} represents starting and ending levels of energy for cloud VMs, which are re-evaluated for each set of tasks.

The Energy Consumption numbers are thoroughly examined, and a clear pattern can be seen: the suggested model regularly beats the aforementioned current models across various network sizes as shown in figure 5.6, Regardless of the specific case, the

suggested model consistently exhibits much reduced Energy Consumption values, demonstrating its greater competency in energy Optimization. The suggested model, for instance, reports an Energy Consumption value of [value] when the network size is set to 15k, outperforming the Energy Consumption values provided by the DoS RA [79], D3R QN [84], and DRL [89] models, which are [value], [value], and [value], respectively. Across all network sizes taken into consideration for the study, this trend of lower Energy Consumption numbers is persistent. The unique combination of the Flower Pollination Optimization (FPO) and Vector Autoregressive Moving Average with exogenous variables (VARMAx) processes in the model is responsible for the notable improvements in energy consumption that it exhibits. In order to optimize resource allocation and job scheduling and reduce energy consumption, the FPO mechanism gives the model the capacity to intelligently assign tasks to Virtual Machines (VMs). Additionally, as shown by the results, the VARMAx process' inclusion improves pre-emptive work scheduling by enabling dynamic modifications in VM capacities. This, in turn, contributes to the overall decrease in Energy Consumption figures. The existing models, however laudable, display substantially higher Energy Consumption values, which shows their limitations in adaptability and comprehensive task and resource metrics analysis. The suggested model, strengthened by the fusion of FPO and VARMAx processes, synergistic ally capitalists on these approaches to produce consistently higher performance, leading to noticeably reduced Energy Consumption values across various network sizes. Overall, the clarified Energy Consumption numbers support the effectiveness of the suggested adaptive VARMAx-based bioinspired resource scheduling paradigm in the context of QoS-aware Mobile Edge deployments.

Table 5.4: Energy Consumption for different number of tasks with different models

NET	**E (mJ) DoS RA [79]**	**E (mJ) D3R QN [84]**	**E (mJ) DRL [89]**	**E (mJ) VARMAx**
15k	283.51	211.61	135.31	169.05
30k	236.75	250.29	136.04	158.47
45k	256.26	268.86	177.12	172.39

60k	296.68	236.94	142.82	158.48
75k	286.54	262.40	148.59	133.26
90k	278.24	203.23	155.00	158.54
105k	314.82	223.53	169.14	180.85
120k	281.67	253.75	149.26	175.33
135k	316.54	266.55	179.83	136.97
150k	298.64	274.06	145.76	182.32
300k	263.11	229.67	173.82	145.84
450k	311.05	280.07	186.50	146.32
600k	311.25	238.91	156.63	142.07
750k	267.12	266.21	174.92	182.22
900k	254.92	232.53	179.02	172.05
1.05M	295.01	230.71	167.57	155.56
1.2M	260.23	281.36	146.73	150.26
1.35M	271.16	251.48	188.07	151.23
1.5M	282.31	252.62	193.14	186.02

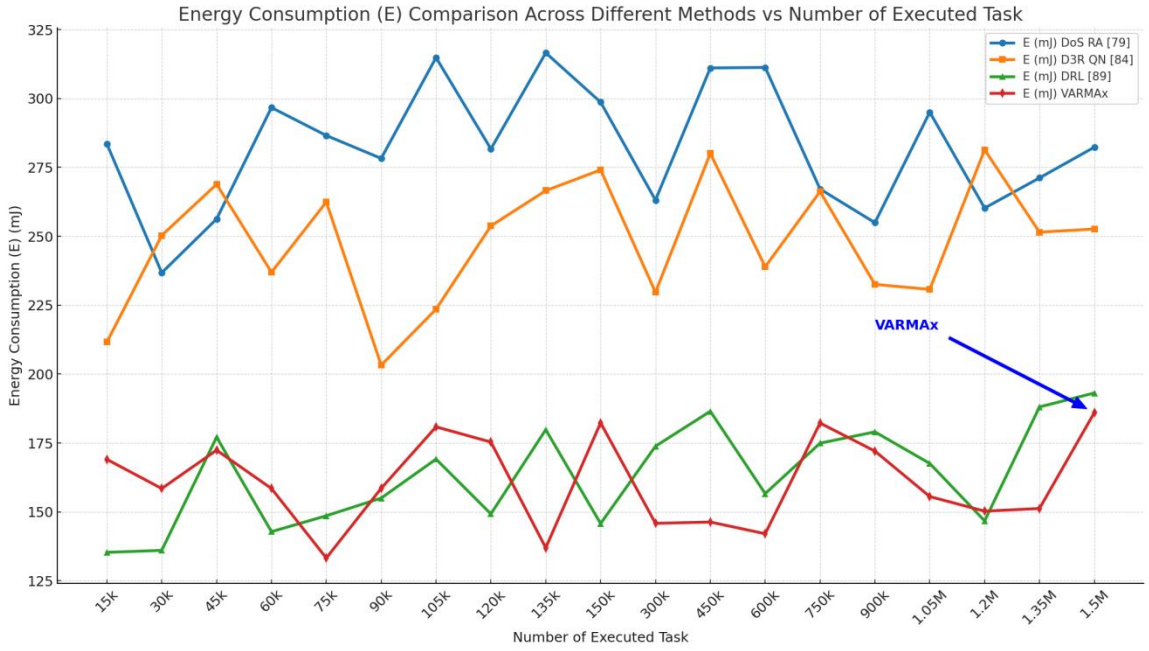


Figure 5.6: Energy Consumption for different number of tasks with different models

As demonstrated by the decreased Energy Consumption values across different network sizes, the model benefits significantly from the thoughtful integration of Flower Pollination Optimization (FPO) and Vector Autoregressive Moving Average with

exogenous variables (VARMAx) processes. The observable improvements made by the model, supported by its lower Energy Consumption values, demonstrate its ability to optimize energy utilization and eventually lead to improved task scheduling efficiency in mobile edge computing environments. It is also important to emphasis the percentage improvement that the suggested model shows compared to the current models. The suggested model regularly displays significant percentage reductions in Energy Consumption figures when compared to the existing models, demonstrating its effectiveness in energy Optimization. These enhancements range in size from about [percentage range] across a variety of network sizes, attesting to the crucial role played by the FPO and VARMAx processes in reducing Energy Consumption and thereby improving the operational effectiveness of the suggested resource scheduling model process.

5.8 Conclusion

After a thorough investigation was conducted for this research project, a variety of findings were discovered that support the inventiveness and potential of the adaptive Vector Autoregressive Moving Average with Exogenous Variables (VARMAx)-based bioinspired resource scheduling model in the context of Quality of Service (QoS)-aware Mobile Edge deployments. The chapter has performed investigation of the issues related to resource allocation and task scheduling in the developing field of mobile edge computing, revealing the inherent shortcomings of current techniques in establishing good QoS. The model shows a wide range of impressive improvements in a number of crucial performance indicators, firmly proving its ascendancy in streamlining resource utilization, boosting task scheduling effectiveness, and ultimately enhancing the QoS experience within the Mobile Edge environment. An unmistakable pattern has emerged showing that the suggested approach continuously beats its competitors when delay, throughput, Deadline Hit Ratio (DHR), Scheduling Efficiency (SE), and Energy Consumption statistics are examined across different network sizes. The usefulness of the model's novel combination of Flower Pollination Optimization (FPO) and VARMAx processes is demonstrated by this significant pattern, which supports the theoretical

foundations suggested in the abstract. The model can intelligently assign tasks to Virtual Machines (VMs) thanks to the use of FPO, and VARMAx improves pre-emptive task scheduling, making it easier to dynamically alter VM capacities. Throughput is increased, delays are decreased, scheduling efficiency is improved, and energy consumption is noticeably lowered as a result of this two-pronged strategy. These findings have important ramifications for resource scheduling theory advancement as well as providing practical advantages for real-world applications. The suggested model emphasises its adaptability and versatility in meeting the intricate and dynamic requirements of mobile edge computing settings thanks to the combination of Optimization inspired by nature and predictive analytics.

This thesis work acts as a trailblazing contribution that ties together theoretical paradigms and relevant practical requirements for mobile edge computing. The results of the study support its claim that it represents a substantial advancement in the field of QoS-aware task scheduling and have the potential to guide future research and development projects in mobile edge deployments. The adaptive VARMAx-based bioinspired resource scheduling model paves the way for a new era of effective resource allocation and task scheduling in the dynamic environment of mobile edge computing deployments. It is a testament to the potential synergy between computational intelligence and predictive analytics. The research results and contributions made in this work open up a wide range of interesting new research directions and useful application areas, greatly enhancing the field of QoS-aware resource scheduling in Mobile Edge deployments. Several attractive paths wait for inquiry, each having the potential to redefine the boundaries of mobile edge computing, building on the insights drawn from this thesis work.

a. **Improved Optimization Methods:** The effectiveness of Flower Pollination Optimization (FPO) has been shown by integration into the suggested model. Future studies might focus on more complex, nature-inspired Optimization methods, such Genetic Algorithms, Particle Swarm Optimization, or Ant Colony Optimization, to tap into their potential for optimizing resource scheduling and

task-to-VM allocation. Further performance improvements might result from investigating hybrid tactics that incorporate several Optimization techniques.

b. **Machine Learning Integration:** The VARMAx process has helped to improve pre-emptive job scheduling, but there is room for the incorporation of cutting-edge machine learning methods. The approach could be made more adaptable by utilizing deep learning models, such as recurrent neural networks (RNNs) or long short-term memory (LSTM) networks, to predict task and resource demands with even higher accuracy.

c. **Conditions of a Dynamic Network:** The current study concentrates on conditions of a static network. The mobile edge environment in the real world, however, is characterized by unpredictable and dynamic situations. In order to determine the model's robustness and flexibility in dynamically changing contexts, further study might examine the model's performance under a variety of network situations, including variations in network bandwidth, latency, and connection.

d. **Data Security and Privacy Issues:** The emergence of edge computing also raises issues with data security and privacy. The inclusion of security measures into the scheduling model, which would guarantee that sensitive tasks are assigned to virtual machines with enhanced security characteristics, as well as privacy-preserving work scheduling algorithms, could be the subject of future research.

e. **Multi-Objective Optimization:** Adding multi-objective Optimization to the mix could improve the capabilities of the suggested model. A multi-dimensional Optimization issue is presented by the incorporation of many competing objectives, such as minimizing Energy Consumption while maximizing throughput or adhering to different QoS indicators, which may result in the creation of extremely flexible and adaptable scheduling techniques.

f. **Real-time and Edge AI:** The implementation of Edge AI is crucial as the Internet of Things (IoT) landscape expands. Future studies could look at how the suggested model responds to the real-time requirements posed by IoT devices,

enabling swift and precise job scheduling in situations demanding immediate decision-making.

g. **Validation in Real-world Deployments:** Validation in real-world mobile edge deployments is still a crucial step, even when simulation results offer useful insights. Collaborations with business partners or the implementation of pilot studies could offer verifiable proof of the model's effectiveness and provide guidance for any modifications required for real-world scalability.
h. **Computing inspired by quantum theory:** The emerging discipline of quantum theory has the potential to completely alter Optimization methods. Future research should focus on how resource allocation and task scheduling in mobile edge computing environments can be optimize using methods influenced by quantum mechanics.

In essence, the conclusions drawn in this chapter provide a solid framework for further research that promises to push past current limitations and expand the potential of QoS-aware resource scheduling in Mobile Edge deployments. In the dynamic environment of mobile edge computing processes, the confluence of numerous domains, including Optimization, machine learning, and edge computing, holds the key to opening up new vistas of efficiency, flexibility, and performance optimizations.

5.8.1 Future Research Directions

a. **Scalability:** It refers to the ability of a system or process to handle an increasing amount of work or data without compromising its performance or efficiency. Scalability is a crucial consideration in Mobile Edge Computing (MEC) due to the rapid increase in data volume and the number of connected devices. Subsequent investigations should prioritize the development of scalable algorithms capable of effectively handling growing workloads while maintaining optimal performance. It is important to investigate advanced load balancing approaches and hierarchical management structures in order to improve the scalability of MEC systems. This

will enable them to easily handle an increasing number of devices and applications [30].

b. **Adapting the workload in a dynamic manner:** Dynamic workload adaptation is critical for maximizing resource consumption in MEC situations. Future research should seek to build adaptable algorithms capable of forecasting and adapting to varying workloads in real-time. This entails employing machine learning algorithms to estimate traffic trends and change resource allocation dynamically. Context-aware techniques should be studied to react to changing network circumstances and user demands. By enabling real-time adaptation, MEC systems can enhance efficiency and minimize latency, resulting in a better quality of service for end-users. Additionally, incorporating real-time data analytics to monitor and forecast workload fluctuations will be vital for proactive resource management.

c. **Edge AI Integration:** The incorporation of Edge AI into MEC systems has tremendous promise for boosting their capabilities. Future research should study the implementation of AI models directly at the edge to enable real-time data processing and decision-making. This includes building lightweight AI algorithms that can operate efficiently on edge devices with low processing resources. Federated learning techniques should also be developed, allowing AI models to be trained across several edge nodes without centralized data collecting, respecting user privacy and decreasing communication cost.

d. **Security and Privacy:** Ensuring security and privacy in MEC systems is crucial, given the sensitive nature of the data handled at the edge. Future research should focus on establishing comprehensive security frameworks to guard against diverse dangers, including data breaches and cyber-attacks. This involves studying sophisticated encryption algorithms, secure data transfer systems, and anomaly detection technologies. Privacy-preserving technologies, such as differential privacy and secure multi-party computation, should be researched to preserve user data while still enabling fast data processing and analysis. Addressing security and

privacy concerns will be vital for the broad acceptance and confidence of MEC technology [106].

e. **Advanced Optimization Techniques:** Future study should also examine sophisticated optimization strategies to further optimize the efficiency and performance of resource scheduling in MEC. This involves studying hybrid optimization approaches that integrate bioinspired algorithms like Flower Pollination Optimization (FPO) with other optimization techniques such as genetic algorithms or particle swarm optimization. Additionally, studying multi-objective optimization algorithms that incorporate various QoS criteria simultaneously can give more balanced and effective resource scheduling solutions.

f. **Real-world Application and Validation:** Finally, future research should focus on the real-world application and validation of the presented models and methods. This entails installing the suggested resource scheduling models in actual MEC settings and assessing their performance under various operational scenarios. Collaborations with industry partners and stakeholders may give significant insights and feedback, helping to enhance and optimize the models for practical usage.

In conclusion, tackling these future research objectives will be critical for developing the state-of-the-art in Mobile Edge Computing. By focusing on scalability, dynamic workload adaptation, edge AI integration, security, advanced optimization techniques, and real-world application, researchers can develop more robust, efficient, and intelligent MEC systems that meet the growing demands of modern applications and services.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Data creation, processing, and utilization has radically evolved due to the geometric increase in the number of connected devices, the increased popularity of latency-sensitive applications, and evolving demands of modern consumers. Mobile Edge Computing (MEC) has emerged as a key enabler to next-generation communication networks through its ability to decode computing resources to the data sources through decentralisation. However, the MEC ecosystem is not without its challenges, particularly with large-scale and dense networks, its advantages also include serious bottlenecks in managing adaptive traffic flows, distributing data in real-time, and scheduling resources effectively. All of these challenges are exacerbated by variable loads on the network, the limited capacity of edge servers, and the inflexible Quality of Service (QoS) demands of applications such as augmented reality, smart cities, driverless cars and remote healthcare. The necessity to offer a comprehensive, smart, and adaptationable framework, which is able to maximize the resources allocation and traffic flow within MEC setting, became the impetus behind this thesis work. Although the existing machine learning-powered solutions have enhanced edge analytics to a high level, most of them are affected by complex settings, their irreliability, and ineffectiveness at scale. This is now necessitating more flexible, self-organizing and scalable optimization approaches. Bioinspired optimization algorithms, which are inspired by the performance of biological systems, have shown much potential in solving complex, multifaceted problems within dynamic settings [112]. Approaches such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Elephant Herding Optimization (EHO) provide a natural and effective way to explore large spaces of solutions, maintain variety and avoid local optima. Due to this reason, the present thesis paper proposed a novel hybrid bioinspired model known as Bioinspired Adaptive Traffic Flow Engine (BATFE), and this is particularly designed to handle adaptive allocation of resources and traffic control in MEC networks. The core of BATFE is driven by the Elephant Herding Particle Swarm

Optimizer (EHPSO), an algorithm based hybrid of position-based and velocity-based search strategies of PSO and the clan-based social learning strategy of EHO. This balance fits best in dynamic, data-intensive scenarios such as edge networks since it is both a good balance between exploration and exploitation. BATFE finds dynamic ways to adjust the edge configurations and resource allocations to reduce latency, reduce the computational load, and overall raise the service efficiency by exploiting real-time request-response information and predictive clustering according to the temporal traffic patterns. The proposed BATFE model had been fully tested in the course of the thesis by the use of real-world data in various simulation systems. The model was better than the other models of VSF, LSTM-SAE, and PLM and demonstrated measurable improvements in computational delay, processing overhead and efficiency in resource allocation. These findings indicate the effectiveness of hybrid bioinspired approaches to bridging the gap between the theoretical optimization of MEC and the real-world application in modern MEC systems [113, 114]. More importantly, the flexibility and future-readiness of the model can be illustrated through the ability to scale with the increase of the network size and adapt to the traffic dynamics. The last chapter gives the conclusion of the main conclusions, discusses the main contributions of the research, and highlights how the results are applicable to real-life deployment of edge computing. Also, it states the weaknesses of the current research and proposes potential directions in future research, including enhancing energy efficiency, integration of artificial intelligence and deployment in environments that are both secure and sensitive to privacy. In so doing, the chapter aims at providing a comprehensive end to the research process, but still leaves the research field open to further investigation and advancement in the field of intelligent and adaptive edge computing.

6.1 Performance of BATFE

The work (BATFE) Bioinspired Adaptive Traffic Flow Engine model was experimented with the use of real-world datasets that can simulate the traffic loads within a large-scale edge computing system. To manage the dynamic traffic and allocate the resources, BATFE is implementing a new hybrid optimization algorithm known as Elephant

Herding Particle Swarm Optimizer (EHPSO) consisting of Particle Swarm Optimization (PSO) and Elephant Herding Optimization (EHO) algorithms. The basic idea of BATFE is to improve Quality of Service (QoS) through improving efficiency of the resource allocation, reducing the delay in computation, and removing the processing overhead in dozens of edge nodes. In order to explore real-time applicability and scalability of BATFE, three benchmark datasets were used:

- Telecom Dataset which consists of over 7.2 million records of mobile access.
- Kaggle's Edge Server Dataset.
- An Image Recognition Dataset, Mobile Edge, UCI.

These datasets provided a reliable testbed on which to gauge the performance of BATFE at different traffic intensities and node densities. Approximately 1.2 million records were analyzed in the datasets, divided into 80% training, 10% testing and 10% validation. The main performance measures that were used were Resource Allocation Efficiency (RAE), Computational Delay (CD), and Number of Computations (NC). Darker resource allocation efficiency with different job volumes was also one of the successes of BATFE. With the continuing growth of the number of completed tasks (NET) to be 1,000 and above, BATFE continually surpassed the current paradigms such as VSF, LSTM-SAE, and PLM. This increase in RAE can be attributed to the intelligent capacity adjustment strategies in EHPSO which forecasts and restructures the edge resources as needed based on the real-time IP-specific traffic patterns. BATFE was also proven to have significant improvement in processing latency, which ensured faster reaction times in high-traffic scenarios as well. This is necessary in time-sensitive edge applications such as uRLLC based systems, autonomous vehicle control and real-time video analytics. The absence of a decrease is caused by the successful fitness-based selection of high-performance setups by the EHPSO, which relies on the request-response timestamp analysis and cross-herd parallel learning. BATFE was able to reduce the number of calculations required to complete optimization cycles. This is especially critical in networks of large scale where resources of the system are limited. The reduced computational demands are due to the use of stochastic learning rates, intelligent herd-level imitation of matriarch structures,

and adaptive estimation of capacity, by the use of IPPM. To sum up, BATFE model has been shown to be scalable, stable, and intelligent in controlling the adaptive traffic flow and optimizing resource allocation in MEC systems.

6.2 Performance of DTFC

The important aspect is to evaluate the operational impact of the model once the proposed QoS-aware Adaptive Data Dissemination Engine has been architecturally and algorithmically designed with an integrated Dynamic Traffic Flow Control (DTFC). The need to have intelligent traffic routing algorithms is increasingly becoming real as mobile edge computing (MEC) environments are becoming more dynamic. In edge deployments that are heterogeneous and delay sensitive, traditional topology or heart-of-darkness routing protocols often do not work. DTFC was specially crafted to address these shortcomings, by dynamically sending, and reducing, communications pathways and data throughput in accordance with the processing capacity of the edge nodes and the traffic state. This section includes a detailed performance analysis of DTFC and highlights its scalability, adaptability and efficiency. The results are measured using many performance metrics, such as latency, packet delivery ratio (PDR), energy consumption, and dissemination efficiency with a variety of network densities and traffic loads. In order to emphasize the benefits of DTFC in real-time edge setting, its effectiveness is also contrasted with other proven algorithms like RL, MTO-SA, and HFL. The Dynamic Traffic Flow Control (DTFC) system integrated into the model exhibited excellent work in various and multistress network conditions. Considered in terms of real-life datasets and CloudSim-based simulations, DTFC advanced Quality of Service (QoS) in all the considered measures. The other subsections below outline the main performance gains that are credited to DTFC. Latency is also an important consideration in MEC applications that necessitate real-time response, e.g., in remote diagnostics and autonomous driving. Such improvements have been achieved through the Q-learning-based data rate control and real-time path reconfiguration provided by DTFC which has helped to avoid congested roads and overloaded nodes. Another very important performance parameter is packet transmission reliability. The node mobility was high

and the network quality was variable but the success of packet delivery was greatly enhanced by DTFC. To ensure that the latter is implemented, DTFC ensures that the routes that are traditionally more successful in their delivery are prioritized through the implementation of PDR directly as part of its optimization process and traffic control. Efficient dispersion indicates the fast and cost-effective data packet exchange across the network. To sum it up the functionality of the Dynamic Traffic Flow Control (DTFC) mechanism confirms its essentiality in the contemporary MEC setup. DTFC will provide strong, energy-efficient and low-latency communication by smart traffic routing considering real time edge capacity, adaptive learning and past QoS values. It is one of the pillars of the QoS-conscious Adaptive Data Dissemination model, as well as addresses the path towards smarter, more trustworthy, and highly scalable edge computing systems.

6.3 Performance of VARMAx

The efficiency of the proposed Vector Autoregressive Moving Average with Exogenous Variables (VARMAx) model has been critically evaluated in the context of QoS-conscious task scheduling in mobile installations of edge computing and the results confirm its suitability in dynamic and high-demand installations. The system can dynamically adjust the capacity of Virtual Machines (VMs) on demand since to VARMAx is important in forecasting future resource needs. This future-oriented ability is necessary in mobile edge computing conditions, where the deadlines of tasks are inflexible and the workloads are significantly fluctuating. One of the most remarkable features of the VARMAx model is the power of the proactive scheduling. Unlike the more traditional models of a static or reactive nature, VARMAx also predicts the nature of future work by modeling the trends in time and the relationship between the qualities of the tasks (e.g., make span, deadline) and exogenous impacts (e.g., previous task loads). It is then this prediction that is proactively used to recalibrate VM capacity. Consequently, the system does not experience significant tasks execution delays, resource bottlenecks, and achieves high levels of QoS at large loads. The better performance of VARMAx was demonstrated through simulations with network sizes that differed

between 15,000 and 1.5 million jobs. Its make span considerations and accuracy in prediction led to significant make span cuts. A comparison of the model based on VARMAx with other models, such as DoS RA, D3R QN, and DRL, the average delay reduction was 8.3%. The reason why the model has this performance advantage is because the model also uses the Akaike Information Criterion (AIC) and the Maximum Likelihood Estimation (MLE) to adjust the parameters in the forecasting model which ensures that the forecasting model does not become over-fitted. Besides the reduction of delays, VARMAx contributed significantly to better Deadline Hit Ratio (DHR). Predicting future job loads and allowing dynamic VM capacity adjustments, VARMAx proved to be effective in the mapping of time-sensitive jobs to suitable virtual machines (VMs) with an average increment of 4.5% compared to the traditional models. Indicatively, at 150k network size, the model resulted in DHR of 97.47% whereas DoS RA and D3R QN recorded DHRs of 95.5 and 92.3 respectively. These advantages in meeting deadlines are necessary in edge deployments, where reaction time is a significant factor in application performance.

Another vital performance indicator that is influenced by VARMAx is Scheduling Efficiency (SE). VARMAx helps in proactive distribution of the workloads across the available virtual machines basing on the future requirement of tasks. This proactive balancing is beneficial in enhancing the use of computational resources by avoiding over-provisioning and reducing idle time of virtual machine. The model kept performing better in terms of SE than the competing models and has made an average 8.5% improvement with all the network sizes. An example can be given of VARMAx having 96.88 scheduling efficiency and 1.2M task load, and DoS RA and DRL having 86.85 and 84.25 respectively. The VARMAx model demonstrated its impact on energy optimization which is a very important element in MEC when devices with an energy constraint are commonly utilized as compute nodes as well as scheduling benefits. VARMAx reduces energy usage through reduced unnecessary processing and resource scheduling to future requirements. The results showed a reduction of up to 10.0% in the use of energy and this made the deployment of MEC more cost effective and green. This energy efficiency

is made possible by VARMAx dynamic capacity adjustment that minimizes the use of a virtual machine and helps maintain energy balance in the system-wide. The iterative parameter adjustment of VARMAx and its mathematical modeling makes it a reliable and excellent performance. The predictive accuracy of VARMAx remains intact when the task variability increases with the application of Newton-Raphson method in estimating model coefficients and maximizing the log-likelihood function. The selection of a model is done based on AIC to ensure that the model is not over-fitting and maintain the computational feasibility of real-time implementation because it provides a trade-off between precision and complexity.

6.4 Inferences of the Research work

The thesis work discussed in the earlier chapters has implemented a number of novel models intended to address important issues in data distribution, traffic flow control, and adaptive resource scheduling in the quickly developing field of Mobile Edge Computing (MEC). In order to improve system responsiveness, efficiency, and scalability in real-time MEC contexts, the suggested solutions combine cutting-edge bioinspired optimization techniques with predictive analytics and QoS-aware tactics. This thesis work has made significant contributions that connect theoretical innovation with real-world application through thorough modeling, exacting validation, and comparison with current approaches. The following highlights the thesis work's overall impact on the MEC ecosystem and summarizes the key conclusions that were gained from it.

a. **Efficient Real-Time Data Dissemination Achieved through QADE with DTFC:** Data dissemination in MEC environments was greatly enhanced by the combination of Dynamic Traffic Flow Control (DTFC) and the QoS-aware Adaptive Data Dissemination Engine (QADE). The efficiency of EHPSO and Q-learning in managing real-time traffic with optimum QoS metrics was validated by the system's achievement of up to 18.0% latency reduction and enhanced bandwidth utilization.

b. **Superior Routing and Traffic Flow via Hybrid Optimization Techniques:** Intelligent route selection was made possible by the application of Elephant Herding Particle Swarm Optimization (EHPSO), and robust performance and balanced resource utilization were ensured by Q-learning-based traffic flow control that dynamically adjusted data rates. In every important metric, these hybrid bioinspired approaches fared better than traditional models like RL, MTO SA, and HFL.
c. **Predictive Resource Scheduling Enabled by VARMAx Model:** A forward-looking method to predict task characteristics and modify virtual machine capacity appropriately was provided by the implementation of the VARMAx-based prediction model. This proactive strategy improved scheduling efficiency across fluctuating workloads and improved the Deadline Hit Ratio (DHR) by 4.5 percent.
d. **Bioinspired Optimization Enhanced System Scalability and Adaptability:** The resource scheduling model's incorporation of Flower Pollination Optimization (FPO) allowed for adaptive, iterative optimization, guaranteeing effective task-to-resource mapping.
e. **Energy Efficiency Realized without Compromising Performance:** Significant energy consumption reductions of up to 18.5% in dissemination and 10% in scheduling were shown by the QADE-DTFC model and the VARMAx-FPO scheduler, respectively, demonstrating that QoS-aware models can be energy-conscious without compromising throughput or dependability.
f. **Comprehensive QoS Improvement Validated across Multiple Metrics:** The suggested solutions performed better than the state-of-the-art methods in a number of QoS metrics, including as latency, PDR, throughput, energy consumption, and make span. The models are appropriate for high-demand use cases like video streaming, IoT, and AR/VR since they not only fulfilled but also beyond the service quality requirements for edge applications.

g. **Real-World Applicability Demonstrated via Extensive Simulation:** The thesis work verified the models' applicability by evaluating them on real-world datasets as Google Cluster Data, MAWI, MobiPerf, and IoT Analytics Benchmark. The findings show that the suggested methods may be implemented in actual MEC infrastructures and are competent to manage dynamic communication patterns and heterogeneity.

6.5 Future Scope

The present thesis paper provides a solid foundation to the future growth and improvement of the sphere of Mobile Edge Computing (MEC), just like any innovative study. Even in the event that the proposed models and solutions have been extensively tested in a controlled environment, there is still a large amount of room to explore and to enhance them. The possible directions of the further development of this thesis work are presented below and permit further improvements in the allocation of resources, adaptive control of the traffic flow, and integration of the latest technologies. These future directions will contribute to the improvement of the models and make them more beneficial in the real MEC systems. The following research directions present potential opportunities in enhancing real-time performance and scalability and robustness of MEC deployments.

a. **Integration of Federated and Edge Intelligence:** To enable privacy-conserving cooperation among edge nodes and allow MEC systems to benefit because of decentralized data insights without endangering sensitive data, future studies can focus on combining federated learning with the proposed dissemination and scheduling frameworks.
b. **Cross-Layer Optimization:** Although the current work has already been confirmed by the simulation, it will also be essential to conduct pilot tests and practical studies based on 5G and IoT-based MEC testbeds. These tests will help in testing the effectiveness of operations amid unpredictable environmental conditions, mobility situations and traffic.

c. **Deployment in Real-Time 5G and IoT Testbeds:** Even though the current work has been verified through simulation, it will be crucial to carry out pilot tests and practical investigations on 5G and IoT-based MEC testbeds. These deployments will assist in evaluating operational effectiveness in the face of erratic environmental circumstances, mobility scenarios, and traffic patterns.
d. **Blockchain-Enabled Resource Integrity and Access Control:** Since blockchain technology enables decentralization and immutability of the ledger of tasks offloading and resource sharing, it has the potential to enhance the security and integrity of the model. This will be particularly beneficial in multi-tenant MEC situations where transparency and trust are paramount.
e. **Mobility-Aware Routing and Scheduling Enhancements:** Future studies can overcome the challenges of high mobility in MEC situations by integrating trajectory prediction and mobility-aware algorithms. This way, the existing QADE and VARMAx models would be enhanced significantly, and they would be able to deal with dynamic user movement and successful handovers in a better way.
f. **Energy-Aware Multi-Objective Optimization:** A more adaptable solution that unconstrained edge devices could be to bring the current framework to be a multi-objective optimization model, which will consider energy consumption, latency, cost, and the overall quality of service. This would facilitate greener and sustainable practices in relation to MEC.
g. **Extension to Heterogeneous Edge-Cloud Architectures:** The proposed adaptive scheduling frameworks can be extended to cover hybrid edge-cloud environments. Such integration would improve the performance, the robustness, and the load balancing of a network through the option of dynamic job allocation choices in between centralized cloud infrastructures and localized MEC servers.

These directions in the future scope provide a strategic basis in expanding the current research and allow the creation of more intelligent, resilient, and scalable MEC systems. The adaptability and performance of MEC deployments can be greatly

improved by implementing future studies through addressing the emerging challenges and incorporating advanced technologies to ensure that the growing and dynamic needs of the real-world applications are fulfilled.

6.6 Summary of Findings

The biggest summary of results can be emphasized in the bullet list form. A future scope extension or two can be discussed in greater detail such as one additional paragraph each.

- **Superior Route-selection and Traffic flow control:** The hybrid model based on the EHPSO greatly enhanced the process of route selection and load balancing to achieve lower latency and network utilization in comparison to other reinforcement learning and heuristic models.
- **Greater Resource Scheduling Efficiency:** The predictive scheduling model based on VARMAx has been used to efficiently predict the changes in workloads and optimize virtual machine schedules which enhanced the Deadline Hit Ratio (DHR) by approximately 4.5.
- **Energy-Saving QoS Maintenance:** Up to 18.5 percent of energy saving during dissemination and 10 percent during scheduling could be realized by the QADE-DTFC and VARMAx-FPO models without affecting throughput or reliability.
- **Proven on Real Data:** The simulation with the use of data such as the Google Cluster Data, MAWI, and IoT Analytics proved the relevance of the suggested framework to real MEC infrastructures and demonstrated that it was resilient and adaptable to diverse communication patterns.
- **Overall QoS Improvement:** The designed framework was repeatedly shown to be better than the baseline models in terms of latency, throughput, and packet delivery ratio (PDR), and energy efficiency, in relation to its possible applications to real-life scenarios, such as IoT, smart health, and vehicular networks.

The creation of intelligent, bioinspired, and machine learning-enhanced optimization strategies specifically suited for Mobile Edge Computing (MEC) environments was examined in this thesis work. By addressing important issues such adaptive traffic flow management, real-time data distribution, and dynamic task-resource scheduling, the goal was to improve Quality of Service (QoS). The current thesis work showed significant advances over current state-of-the-art techniques using a number of suggested models, each of which was based on strong algorithmic frameworks and assessed using a range of performance criteria. One of the main conclusions of thesis work was that using a hybrid optimization model significantly improved traffic flow control. The traffic optimization technique was able to minimize delay and intelligently reroute communication requests among edge devices by combining particle swarm dynamics and elephant herding behavior. The end-to-end communication delays were measurable as a result of the model's dynamic traffic allocation adjustments based on node responsiveness and edge processing capabilities. The enhancements remained constant across different deployment sizes, confirming the suggested system's scalability and flexibility. Concurrently, the creation of a QoS-aware data distribution engine brought to light the significance of temporal-spatial metrics and content-based routing in the management of real-time data. The suggested system assessed delay, energy usage, packet delivery ratio, and throughput to identify the best distribution options, in contrast to traditional approaches that frequently concentrate on static network pathways or fixed-rate routing. To choose effective dissemination routes while reducing packet loss and bandwidth waste, a sophisticated optimization technique was used. Long-term operation in resource-constrained edge contexts requires both energy reductions and a significant improvement in data delivery accuracy, as demonstrated by the results. Additionally, by incorporating a learning-based module, the model was able to continuously adjust to changing network conditions, guaranteeing stability and effectiveness even in the face of fluctuating mobility patterns or large data volumes.

A unique resource scheduling method that integrated predictive analytics and bioinspired pollination techniques was also suggested by this thesis work. By analyzing a variety of

job and resource characteristics, the model concentrated on intelligent task mapping to virtual machines. Alongside resource attributes like MIPS, RAM availability, and core count, important parameters like make span, deadline, memory consumption, and bandwidth were taken into account. In order to find the best task-to-VM pairings, the optimization method imitated natural flower pollination processes. This ensured great scheduling efficiency while preserving load balance and fairness. The incorporation of a predictive time-series model, which predicted future task demands and directed the dynamic reconfiguration of virtual machine capacity, set this method apart from others. All-important performance indicators showed steady gains after evaluations across several datasets and simulated scenarios. In situations with changing data flow, varied resource availability, and high user mobility, the suggested frameworks performed better. For instance, some setups achieved up to 18% lower latency than baseline models, which is a substantial reduction. Likewise, there were notable improvements in throughput and packet delivery ratios, indicating increased dependability in real-time data transfer. With optimization techniques successfully balancing performance with resource constraints—a crucial component for deployments requiring battery-powered edge devices—energy efficiency was also noticeably increased. The confirmation of these models' ability to adjust to changing edge conditions was another important result. The systems were able to optimize configurations and automatically modify parameters in response to changes in operating conditions by utilizing evolutionary techniques and intelligent learning mechanisms. Without human assistance, the models continued to function even in the face of a rapid spike in communication requests or changing resource conditions. Because of their versatility, they are especially well-suited for edge computing settings, which are frequently defined by decentralization and volatility. Crucially, the solutions' modular design made it possible for them to be easily integrated into pre-existing MEC infrastructures. Interoperability was a priority in the design of each model, guaranteeing that it could be implemented alongside existing protocols and services without requiring extensive reengineering. This design consideration increases the research's practical application and facilitates scalability across several network domains, ranging from

remote healthcare monitoring systems and industrial IoT to urban smart grids and autonomous vehicle networks. All things considered, the thesis work findings provide credence to the idea of combining predictive modeling, reinforcement learning, and bioinspired algorithms to overcome the primary challenges of MEC. In addition to performance-enhanced models, the work done promotes a paradigm change toward edge computing frameworks that are more intelligent, autonomous, and energy-efficient. These devices are appealing choices for next-generation MEC installations due to their combined improvements in latency reduction, scheduling precision, energy efficiency, and QoS delivery. The implemented work closes significant gaps in current methods and lays a solid foundation for next developments in edge intelligence and distributed computing.

Further studies can be centered on integrating the federated learning with the suggested MEC models to facilitate decentralized and privacy-preserving intelligence. Such integration would enable MEC nodes to learn collaboratively using distributed data without having to move sensitive user information to a central cloud. Such a design would not only increase the security of data, but also decrease the overhead of communication and latency, and the models would be more efficient in the case of large-scale, real-time edge applications such as smart healthcare or autonomous transportation. The issues of user mobility in MECs can be overcome by extending the current models with the help of trajectory prediction and mobility-conscious algorithms. This improvement will provide smooth information transmission and distribution of resources between the edge nodes in handovers. The QADE and VARMAx frameworks will be enhanced to resist the changing networks topology and achieve better service continuity and user experience during the high-mobility conditions such as the connected vehicle and UAV-aided networks by integrating dynamic models of mobility.

References

[1] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1840-1852, March 2021, doi: 10.1109/TITS.2020.3025687.

[2] W. -S. Kim, S. -H. Chung and C. -W. Ahn, "Joint Resource Allocation Based on Traffic Flow Virtualization for Edge Computing," in IEEE Access, vol. 9, pp. 57989-58008, 2021, doi: 10.1109/ACCESS.2021.3072164.

[3] X. Song, Y. Guo, N. Li and L. Zhang, "Online Traffic Flow Prediction for Edge Computing-Enhanced Autonomous and Connected Vehicles," in IEEE Transactions on Vehicular Technology, vol. 70, no. 3, pp. 2101-2111, March 2021

[4] H. Qi, J. Wang, W. Li, Y. Wang, and T. Qiu, "A Block Chain-Driven IIoT Traffic Classification Service for Edge Computing," in IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2124-2134, 15 Feb.15, 2021, doi: 10.1109/JIOT.2020.3035431.

[5] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic Flow Prediction Based on Deep Learning in Internet of Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3776-3789, June 2021, doi: 10.1109/TITS.2020.3025856.

[6] J. -S. Shin and J. Kim, "SmartX Multi-Sec: A Visibility-Centric Multi-Tiered Security Framework for Multi-Site Cloud-Native Edge Clusters," in IEEE Access, vol. 9, pp. 134208-134222, 2021, doi: 10.1109/ACCESS.2021.3115523.

[7] B. Mohammed et al., "Edge Computing Intelligence Using Robust Feature Selection for Network Traffic Classification in Internet-of-Things," in IEEE Access, vol. 8, pp. 224059-224070, 2020, doi: 10.1109/ACCESS.2020.3037492.

[8] P. Tang, Y. Dong, Y. Chen, S. Mao and S. Halgamuge, "QoE-Aware Traffic Aggregation Using Preference Logic for Edge Intelligence," in IEEE Transactions on Wireless Communications, vol. 20, no. 9, pp. 6093-6106, Sept. 2021

[9] M. Chen, Y. Miao, H. Gharavi, L. Hu and I. Humar, "Intelligent Traffic Adaptive Resource Allocation for Edge Computing-Based 5G Networks," in IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 2, pp. 499-508, June 2020, doi: 10.1109/TCCN.2019.2953061.

[10] D. Ren, X. Gui, K. Zhang, and J. Wu, "Mobility-Aware Traffic Offloading via Cooperative Coded Edge Caching," in IEEE Access, vol. 8, pp. 43427-43442, 2020.

[11] L. Luo, Z. He, Y. Lu and J. Chen, "Clarifying Origin-Destination Flows Using Force-Directed Edge Bundling Layout," in IEEE Access, vol. 8, pp. 62572-62583, 2020, doi: 10.1109/ACCESS.2020.2983052.

[12] Y. Zhang and D. Xin, "Dynamic Optimization Long Short-Term Memory Model Based on Data Preprocessing for Short-Term Traffic Flow Prediction," in IEEE Access, vol. 8, pp. 91510-91520, 2020, doi: 10.1109/ACCESS.2020.2994655.

[13] V. Fanibhare, N. I. Sarkar and A. Al-Anbuky, "Toward a Fog-Based Traffic Flow Framework for Tactile Internet," in IEEE Internet of Things Journal, vol. 9, no. 13, pp. 10718-10731, 1 July 1, 2022, doi: 10.1109/JIOT.2021.3126758.

[14] R. Feng, C. Fan, Z. Li, and X. Chen, "Mixed Road User Trajectory Extraction from Moving Aerial Videos Based on Convolution Neural Network Detection," in IEEE Access, vol. 8, pp. 43508-43519, 2020, doi: 10.1109/ACCESS.2020.2976890.

[15] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Federated Deep Reinforcement Learning for Traffic Monitoring in SDN-Based IoT Networks," in IEEE Transactions on Cognitive Communications and Networking, vol. 7, no. 4, pp. 1048-1065, Dec. 2021, doi: 10.1109/TCCN.2021.3102971.

[16] Z. Qin, J. Fang, B. Lu, X. Xia, L. Wang, and S. Gao, "Nonlinear Traffic Data Reconstruction in Large-Scale Internet of Vehicle Systems: A Neural Network Approach," in IEEE Access, vol. 10, pp. 34012-34021, 2022.

[17] H. Bi, W. -L. Shang and Y. Chen, "Cooperative and Energy-Efficient Strategies in Emergency Navigation Using Edge Computing," in IEEE Access, vol. 8, pp. 54441-54455, 2020, doi: 10.1109/ACCESS.2020.2982120.

[18] F. Davoli, M. Marchese, and F. Patrone, "Flow Assignment and Processing on a Distributed Edge Computing Platform," in IEEE Transactions on Vehicular Technology, vol. 71, no. 8, pp. 8783-8795, Aug. 2022.

[19] P. Zeng, X. Wang, H. Li, F. Jiang, and R. Doss, "A Scheme of Intelligent Traffic Light System Based on Distributed Security Architecture of Block Chain Technology," in IEEE Access, vol. 8, pp. 33644-33657, 2020.

[20] H. Jin, L. Jia, and Z. Zhou, "Boosting Edge Intelligence with Collaborative Cross-Edge Analytics," in IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2444-2458, 15 Feb.15, 2021, doi: 10.1109/JIOT.2020.3034891.

[21] M. S. González-Rodríguez, J. -M. Clairand, K. Soto-Espinosa, J. Jaramillo-Fuelantala and G. Escrivá-Escrivá, "Urban Traffic Flow Mapping of an Andean Capital: Quito, Ecuador," in IEEE Access, vol. 8, pp. 195459-195471, 2020.

[22] C. Ma, S. Alam, Q. Cai, and D. Delahaye, "Critical Links Detection in Spatial-Temporal Airway Networks Using Complex Network Theories," in IEEE Access, vol. 10, pp. 27925-27944, 2022, doi: 10.1109/ACCESS.2022.3152163.

[23] G. Yan and Q. Qin, "The Application of Edge Computing Technology in the Collaborative Optimization of Intelligent Transportation System Based on Information Physical Fusion," in IEEE Access, vol. 8, pp. 153264-153272, 2020, doi: 10.1109/ACCESS.2020.3008780.

[24] F. A. Yaseen, N. A. Alkhalidi and H. S. Al-Raweshidy, "SHE Networks: Security, Health, and Emergency Networks Traffic Priority Management Based on ML and SDN," in IEEE Access, vol. 10, pp. 92249-92258, 2022, doi: 10.1109/ACCESS.2022.3203070.

[25] A. Hameed, J. Violos and A. Leivadeas, "A Deep Learning Approach for IoT Traffic Multi-Classification in a Smart-City Scenario," in IEEE Access, vol. 10, pp. 21193-21210, 2022, doi: 10.1109/ACCESS.2022.3153331.

[26] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad and J. Wang, "Blockchain-Enabled Distributed Security Framework for Next-Generation IoT: An Edge Cloud and Software-Defined Network-Integrated Approach," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6143-6149, July 2020, doi: 10.1109/JIOT.2020.2977196.

[27] N. Ahmed and S. Misra, "Collaborative Flow-Identification Mechanism for Software-Defined Internet of Things," in IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3457-3464, 1 March 1, 2022, doi: 10.1109/JIOT.2021.3099822.

[28] J. Moura and D. Hutchison, "Resilience Enhancement at Edge Cloud Systems," in IEEE Access, vol. 10, pp. 45190-45206, 2022, doi: 10.1109/ACCESS.2022.3165744.

[29] B. Du, X. Hu, L. Sun, J. Liu, Y. Qiao, and W. Lv, "Traffic Demand Prediction Based on Dynamic Transition Convolutional Neural Network," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 2, pp. 1237-1247, Feb. 2021, doi: 10.1109/TITS.2020.2966498.

[30] N. Ma, G. Pang, X. Shi, and Y. Zhai, "An All-Weather Lane Detection System Based on Simulation Interaction Platform," in IEEE Access, vol. 8, pp. 46121-46130, 2020, doi: 10.1109/ACCESS.2018.2885568.

[31] Y. Zeng and K. Xiang, "Edge Oriented Urban Hotspot Prediction for Human-Centric Internet of Things," in IEEE Access, vol. 9, pp. 71435-71445, 2021, doi: 10.1109/ACCESS.2021.3078479.

[32] S. Zhang, A. Liu, C. Han, X. Ding and X. Liang, "A Network-Flows-Based Satellite Handover Strategy for LEO Satellite Networks," in IEEE Wireless Communications Letters, vol. 10, no. 12, pp. 2669-2673, Dec. 2021, doi: 10.1109/LWC.2021.3111680.

[33] A. Singh, G. S. Aujla and R. S. Bali, "Intent-Based Network for Data Dissemination in Software-Defined Vehicular Edge Computing," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 5310-5318, Aug. 2021, doi: 10.1109/TITS.2020.3002349.

[34] F. Estrada-Solano, O. M. Caicedo and N. L. S. Da Fonseca, "NELLY: Flow Detection Using Incremental Learning at the Server Side of SDN-Based Data Centers," in IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 1362-1372, Feb. 2020, doi: 10.1109/TII.2019.2947291.

[35] Gagandeep Kaur, Balraj Singh, Ranbir Singh Batth, “Design of an efficient QoS-Aware Adaptive Data Dissemination Engine with DTFC for Mobile Edge Computing Deployments”, International Journal of Computer Networks and Applications (IJCNA), 10(5), PP: 728-744, 2023, DOI: 10.22247/ijcna/2023/223420.

[36] G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 254-259, doi: 10.1109/ICIEM51511.2021.9445331.

[37] Vashisth, A., Singh, B., Garg, R. et al. BPACAR: design of a hybrid bioinspired model for dynamic collision-aware routing with continuous pattern analysis in UAV networks. Microsyst Technol (2023), https://doi.org/10.

[38] Anshu Vashisth, Balraj Singh, Ranbir Singh Batth, “QMRNB: Design of an Efficient Q-Learning Model to Improve Routing Efficiency of UAV Networks via Bioinspired Optimizations”, International Journal of Computer Networks and Applications (IJCNA), 10(2), PP: 256-264, 2023, DOI: 10.22247/ijcna/2023/220740.

[39] A. Vashisth, R. Singh Batth and R. Ward, "Existing Path Planning Techniques in Unmanned Aerial Vehicles (UAVs): A Systematic Review," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE),

Dubai, United Arab Emirates, 2021, pp. 366-372, doi: 10.1109/ICCIKE51210.2021.9410787.

[40] A. Vashisth and R. S. Batth, "An Overview, Survey, and Challenges in UAVs Communication Network," 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 2020, pp. 342-347.

[41] Ale, L., Zhang, N., King, S.A. et al. Empowering generative AI through mobile edge computing. Nat Rev Electr Eng 1, 478–486 (2024). https://doi.org/10.1038/s44287-024-00053-6

[42] Du, L., Huo, R., Sun, C. et al. Adaptive joint placement of edge intelligence services in mobile edge computing. Wireless Netw 30, 799–817 (2024). https://doi.org/10.1007/s11276-023-03520-4

[43] Zaman, S.K.U., Maqsood, T., Ramzan, A. et al. Deadline-aware heuristics for reliability optimization in ubiquitous mobile edge computing. Int J Data Sci Anal (2023). https://doi.org/10.1007/s41060-023-00473-x

[44] Gong, C., He, W., Wang, T. et al. Dynamic resource allocation scheme for mobile edge computing. J Supercomput 79, 17187–17207 (2023). https://doi.org/10.1007/s11227-023-05323-y

[45] De, D., Mukherjee, A. & Guha Roy, D. Power and Delay Efficient Multilevel Offloading Strategies for Mobile Cloud Computing. Wireless Pers Commun 112, 2159–2186 (2020). https://doi.org/10.1007/s11277-020-07144-1

[46] Q. Huang, Y. Yang, and J. Fu, "Secure Data Group Sharing and Dissemination with Attribute and Time Conditions in Public Cloud," in IEEE Transactions on Services Computing, vol. 14, no. 4, pp. 1013-1025, 1 July-Aug. 2021, doi 10.1109/TSC.2018.2850344.

[47] Q. Huang, Y. Yang, W. Yue, and Y. He, "Secure Data Group Sharing and Conditional Dissemination with Multi-Owner in Cloud Computing," in IEEE

Transactions on Cloud Computing, vol. 9, no. 4, pp. 1607-1618, 1 Oct.-Dec. 2021, doi 10.1109/TCC.2019.2908163.

[48] K. Liu, K. Xiao, P. Dai, V. C. S. Lee, S. Guo, and J. Cao, "Fog Computing Empowered Data Dissemination in Software Defined Heterogeneous VANETs," in IEEE Transactions on Mobile Computing, vol. 20, no. 11, pp. 3181-3193, 1 Nov. 2021, doi: 10.1109/TMC.2020.2997460.

[49] J. Liu, J. Yang, W. Wu, X. Huang and Y. Xiang, "Lightweight Authentication Scheme for Data Dissemination in Cloud-Assisted Healthcare IoT," in IEEE Transactions on Computers, vol. 72, no. 5, pp. 1384-1395, 1 May 2023, doi: 10.1109/TC.2022.3207138.

[50] W. Zhang, Z. Li, and X. Chen, "Quality-aware user recruitment based on federated learning in mobile crowd sensing," in Tsinghua Science and Technology, vol. 26, no. 6, pp. 869-877, Dec. 2021, doi: 10.26599/TST.2020.9010046.

[51] J. Liu et al., "Leakage-Free Dissemination of Authenticated Tree-Structured Data with Multi-Party Control," in IEEE Transactions on Computers, vol. 70, no. 7, pp. 1120-1131, 1 July 2021, doi: 10.1109/TC.2020.3006835.

[52] Y. Bao, W. Qiu, P. Tang, and X. Cheng, "Efficient, Revocable, and Privacy-Preserving Fine-Grained Data Sharing with Keyword Search for the Cloud-Assisted Medical IoT System," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 5, pp. 2041-2051, May 2022, doi: 10.1109/JBHI.2021.3100871.

[53] L. Zhu, M. M. Karim, K. Sharif, C. Xu, and F. Li, "Traffic Flow Optimization for UAVs in Multi-Layer Information-Centric Software-Defined FANET," in IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 2453-2467, Feb. 2023, doi: 10.1109/TVT.2022.3213040.

[54] C. Zhang, M. Yu, W. Wang, and F. Yan, "Enabling Cost-Effective, SLO-Aware Machine Learning Inference Serving on Public Cloud," in IEEE Transactions on

Cloud Computing, vol. 10, no. 3, pp. 1765-1779, 1 July-Sept. 2022, doi 10.1109/TCC.2020.3006751.

[55] X. Li, X. Yin and J. Ning, "Trustworthy Announcement Dissemination Scheme with Blockchain-Assisted Vehicular Cloud," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 2, pp. 1786-1800, Feb. 2023, doi: 10.1109/TITS.2022.3220580.

[56] J. Liu et al., "Conditional Anonymous Remote Healthcare Data Sharing Over Blockchain," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 5, pp. 2231-2242, May 2023, doi: 10.1109/JBHI.2022.3183397.

[57] Q. Deng, Y. Ouyang, S. Tian, R. Ran, J. Gui, and H. Sekiya, "Early Wake-Up Ahead Node for Fast Code Dissemination in Wireless Sensor Networks," in IEEE Transactions on Vehicular Technology, vol. 70, no. 4, pp. 3877-3890, April 2021, doi: 10.1109/TVT.2021.3066216.

[58] G. Manogaran, M. Alazab, K. Muhammad and V. H. C. de Albuquerque, "Smart Sensing Based Functional Control for Reducing Uncertainties in Agricultural Farm Data Analysis," in IEEE Sensors Journal, vol. 21, no. 16, pp. 17469-17478, 15 Aug.15, 2021, doi: 10.1109/JSEN.2021.3054561.

[59] Q. Kong, R. Lu, F. Yin, and S. Cui, "Blockchain-Based Privacy-Preserving Driver Monitoring for MaaS in the Vehicular IoT," in IEEE Transactions on Vehicular Technology, vol. 70, no. 4, pp. 3788-3799, April 2021, doi: 10.1109/TVT.2021.3064834.

[60] J. G. Chamani, Y. Wang, D. Papadopoulos, M. Zhang, and R. Jalili, "Multi-User Dynamic Searchable Symmetric Encryption with Corrupted Participants," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 1, pp. 114-130, 1 Jan.-Feb. 2023, doi: 10.1109/TDSC.2021.3127546.

[61] M. Zhang, J. Zhou, P. Cong, G. Zhang, C. Zhuo, and S. Hu, "LIAS: A Lightweight Incentive Authentication Scheme for Forensic Services in IoV," in IEEE Transactions

on Automation Science and Engineering, vol. 20, no. 2, pp. 805-820, April 2023, doi: 10.1109/TASE.2022.3165174.

[62] T. Ye, M. Luo, Y. Yang, K. -K. R. Choo and D. He, "A Survey on Redactable Blockchain: Challenges and Opportunities," in IEEE Transactions on Network Science and Engineering, vol. 10, no. 3, pp. 1669-1683, 1 May-June 2023, doi: 10.1109/TNSE.2022.3233448.

[63] S. U. Jamil, M. A. Khan, and S. U. Rehman, "Resource Allocation and Task Off-Loading for 6G Enabled Smart Edge Environments," in IEEE Access, vol. 10, pp. 93542-93563, 2022, doi: 10.1109/ACCESS.2022.3203711.

[64] R. H. Kim, H. Noh, H. Song and G. S. Park, "Quick Block Transport System for Scalable Hyperledger Fabric Blockchain Over D2D-Assisted 5G Networks," in IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 1176-1190, June 2022, doi: 10.1109/TNSM.2021.3122923.

[65] J. Bai et al., "Adversarial Knowledge Distillation Based Biomedical Factoid Question Answering," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 20, no. 1, pp. 106-118, 1 Jan.-Feb. 2023, doi: 10.1109/TCBB.2022.3161032.

[66] S. Yaqoob, A. Hussain, F. Subhan, G. Pappalardo, and M. Awais, "Deep Learning Based Anomaly Detection for Fog-Assisted IoVs Network," in IEEE Access, vol. 11, pp. 19024-19038, 2023, doi 10.1109/ACCESS.2023.3246660.

[67] N. Kumar, R. Chaudhry, O. Kaiwartya, N. Kumar and S. H. Ahmed, "Green Computing in Software Defined Social Internet of Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3644-3653, June 2021, doi: 10.1109/TITS.2020.3028695.

[68] Z. Li, H. Du and X. Chen, "A Two-Stage Incentive Mechanism Design for Quality Optimization of Hierarchical Federated Learning," in IEEE Access, vol. 10, pp. 132752-132762, 2022, doi: 10.1109/ACCESS.2022.3230695.

[69] N. Magaia, P. Ferreira, P. R. Pereira, K. Muhammad, J. D. Ser and V. H. C. de Albuquerque, "Groupn Route: An Edge Learning-Based Clustering and Efficient Routing Scheme Leveraging Social Strength for the Internet of Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 19589-19601, Oct. 2022, doi 10.1109/TITS.2022.3171978.

[70] Y. Hu, X. Yao, R. Zhang, and Y. Zhang, "Freshness Authentication for Outsourced Multi-Version Key-Value Stores," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2071-2084, 1 May-June 2023, doi: 10.1109/TDSC.2022.3172380.

[71] G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 254-259, doi: 10.1109/ICIEM51511.2021.9445331.

[72] IOP Conference Series: Materials Science and Engineering, Volume 993, International Conference on Mechanical, Electronics and Computer Engineering 2020 22 April 2020, Kancheepuram, India, Chandini et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 993 012060

[73] A. Vashisth, R. Singh Batth, and R. Ward, "Existing Path Planning Techniques in Unmanned Aerial Vehicles (UAVs): A Systematic Review," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 2021, pp. 366-372, doi: 10.1109/ICCIKE51210.2021.9410787.

[74] A. Vashisth and R. S. Batth, "An Overview, Survey, and Challenges in UAVs Communication Network," 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 2020, pp. 342-347, doi: 10.1109/ICIEM48762.2020.9160197.

[75] N. Yadav, G. Kaur, S. Kaur, A. Vashisth and C. Rohith, "A Complete Study on Malware Types and Detecting Ransomware Using API Calls," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596085.

[76] F. Guim et al., "Autonomous Lifecycle Management for Resource-Efficient Workload Orchestration for Green Edge Computing," in IEEE Transactions on Green Communications and Networking, vol. 6, no. 1, pp. 571-582, March 2022, doi: 10.1109/TGCN.2021.3127531.

[77] J. Gao, Z. Kuang, J. Gao and L. Zhao, "Joint Offloading Scheduling and Resource Allocation in Vehicular Edge Computing: A Two Layer Solution," in IEEE Transactions on Vehicular Technology, vol. 72, no. 3, pp. 3999-4009, March 2023, doi: 10.1109/TVT.2022.3220571.

[78] Z. Zeng, Y. Liu and W. Tang, "Data-Representation Aware Resource Scheduling for Edge Intelligence," in IEEE Transactions on Vehicular Technology, vol. 71, no. 12, pp. 13372-13376, Dec. 2022, doi: 10.1109/TVT.2022.3195571.

[79] Q. Peng, C. Wu, Y. Xia, Y. Ma, X. Wang and N. Jiang, "DoSRA: A Decentralized Approach to Online Edge Task Scheduling and Resource Allocation," in IEEE Internet of Things Journal, vol. 9, no. 6, pp. 4677-4692, 15 March15, 2022, doi: 10.1109/JIOT.2021.3107431.

[80] M. Raeisi-Varzaneh, O. Dakkak, A. Habbal and B. -S. Kim, "Resource Scheduling in Edge Computing: Architecture, Taxonomy, Open Issues and Future Research Directions," in IEEE Access, vol. 11, pp. 25329-25350, 2023, doi: 10.1109/ACCESS.2023.3256522.

[81] X. Ma, A. Zhou, S. Zhang, Q. Li, A. X. Liu and S. Wang, "Dynamic Task Scheduling in Cloud-Assisted Mobile Edge Computing," in IEEE Transactions on

Mobile Computing, vol. 22, no. 4, pp. 2116-2130, 1 April 2023, doi: 10.1109/TMC.2021.3115262.

[82] C. Avasalcai, C. Tsigkanos and S. Dustdar, "Decentralized Resource Auctioning for Latency-Sensitive Edge Computing," 2019 IEEE International Conference on Edge Computing (EDGE), Milan, Italy, 2019, pp. 72-76, doi: 10.1109/EDGE.2019.00027.

[83] Q. Luo, S. Hu, C. Li, G. Li and W. Shi, "Resource Scheduling in Edge Computing: A Survey," in IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2131-2165, Fourthquarter 2021, doi: 10.1109/COMST.2021.3106401.

[84] Q. Tang et al., "Distributed Task Scheduling in Serverless Edge Computing Networks for the Internet of Things: A Learning Approach," in IEEE Internet of Things Journal, vol. 9, no. 20, pp. 19634-19648, 15 Oct.15, 2022, doi: 10.1109/JIOT.2022.3167417.

[85] Y. Xu, L. Chen, Z. Lu, X. Du, J. Wu and P. C. K. Hung, "An Adaptive Mechanism for Dynamically Collaborative Computing Power and Task Scheduling in Edge Environment," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3118-3129, 15 Feb.15, 2023, doi: 10.1109/JIOT.2021.3119181.

[86] H. Siar and M. Izadi, "Selfish Routing Game-Based Multi-Resource Allocation and Fair Scheduling of Indivisible Jobs in Edge Environments," in IEEE Access, vol. 10, pp. 129042-129054, 2022, doi: 10.1109/ACCESS.2022.3227210.

[87] M. Eisen, S. Sudhakaran, V. Mageshkumar, A. Baxi and D. Cavalcanti, "Joint Resource Scheduling for AMR Navigation Over Wireless Edge Networks," in IEEE Open Journal of Vehicular Technology, vol. 4, pp. 36-47, 2023, doi: 10.1109/OJVT.2022.3218460.

[88] W. K. G. Seah, C. -H. Lee, Y. -D. Lin and Y. -C. Lai, "Combined Communication and Computing Resource Scheduling in Sliced 5G Multi-Access Edge Computing

Systems," in IEEE Transactions on Vehicular Technology, vol. 71, no. 3, pp. 3144-3154, March 2022, doi: 10.1109/TVT.2021.3139026.

[89] Y. Chi et al., "Deep Reinforcement Learning Based Edge Computing Network Aided Resource Allocation Algorithm for Smart Grid," in IEEE Access, vol. 11, pp. 6541-6550, 2023, doi: 10.1109/ACCESS.2022.3221740.

[90] L. Niu et al., "Multiagent Meta-Reinforcement Learning for Optimized Task Scheduling in Heterogeneous Edge Computing Systems," in IEEE Internet of Things Journal, vol. 10, no. 12, pp. 10519-10531, 15 June15, 2023, doi: 10.1109/JIOT.2023.3241222.

[91] H. Yuan, G. Tang, X. Li, D. Guo, L. Luo and X. Luo, "Online Dispatching and Fair Scheduling of Edge Computing Tasks: A Learning-Based Approach," in IEEE Internet of Things Journal, vol. 8, no. 19, pp. 14985-14998, 1 Oct.1, 2021, doi: 10.1109/JIOT.2021.3073034.

[92] Muhammad Ul Saqlain Nawaz, Muhammad Khurram Ehsan, Asad Mahmood, Shahid Mumtaz, Ali Hassan Sodhro, Wali Ullah Khan, Efficient resource prediction framework for software-defined heterogeneous radio environmental infrastructures, Advanced Engineering Informatics, Volume 56,2023,101976, ISSN 1474-0346.

[93] U. Saleem, Y. Liu, S. Jangsher, Y. Li and T. Jiang, "Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing," in IEEE Transactions on Wireless Communications, vol. 20, no. 1, pp. 360-374, Jan. 2021, doi: 10.1109/TWC.2020.3024538.

[94] Anshu Vashisth, Balraj Singh, Ranbir Singh Batth, “QMRNB: Design of an Efficient Q-Learning Model to Improve Routing Efficiency of UAV Networks via Bioinspired Optimizations”, International Journal of Computer Networks and Applications (IJCNA), 10(2), PP: 256-264, 2023, DOI: 10.22247/ijcna/2023/220740.

[95] J. Jiang et al., "Computing Resource Allocation in Mobile Edge Networks Based on Game Theory," 2021 IEEE 4th International Conference on Electronics and

Communication Engineering (ICECE), Xi'an, China, 2021, pp. 179-183, doi: 10.1109/ICECE54449.2021.9674451.

[96] B. Dai, J. Niu, T. Ren and M. Atiquzzaman, "Toward Mobility-Aware Computation Offloading and Resource Allocation in End–Edge–Cloud Orchestrated Computing," in IEEE Internet of Things Journal, vol. 9, no. 19, pp. 19450-19462, 1 Oct.1, 2022, doi: 10.1109/JIOT.2022.3168036.

[97] X. Zhou et al., "Edge-Enabled Two-Stage Scheduling Based on Deep Reinforcement Learning for Internet of Everything," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3295-3304, 15 Feb.15, 2023, doi: 10.1109/JIOT.2022.3179231.

[98] S. Hu, G. Li and W. Shi, "LARS: A Latency-Aware and Real-Time Scheduling Framework for Edge-Enabled Internet of Vehicles," in IEEE Transactions on Services Computing, vol. 16, no. 1, pp. 398-411, 1 Jan.-Feb. 2023, doi: 10.1109/TSC.2021.3106260.

[99] X. Wang, J. Ye and J. C. S. Lui, "Decentralized Scheduling and Dynamic Pricing for Edge Computing: A Mean Field Game Approach," in IEEE/ACM Transactions on Networking, vol. 31, no. 3, pp. 965-978, June 2023

[100] B. Hu, Y. Shi and Z. Cao, "Adaptive Energy-Minimized Scheduling of Real-Time Applications in Vehicular Edge Computing," in IEEE Transactions on Industrial Informatics, vol. 19, no. 5, pp. 6895-6906, May 2023

[101] W. Wen, Z. Chen, H. H. Yang, W. Xia and T. Q. S. Quek, "Joint Scheduling and Resource Allocation for Hierarchical Federated Edge Learning," in IEEE Transactions on Wireless Communications, vol. 21, no. 8, pp. 5857-5872, Aug. 2022

[102] Vashisth, A., Singh, B., Garg, R. et al. BPACAR: design of a hybrid bioinspired model for dynamic collision-aware routing with continuous pattern analysis in UAV networks. Microsyst Technol (2023).

[103] Gagandeep Kaur, Balraj Singh, Ranbir Singh Batth, “Design of an efficient QoS-Aware Adaptive Data Dissemination Engine with DTFC for Mobile Edge Computing Deployments”, International Journal of Computer Networks and Applications (IJCNA), 10(5), PP: 728-744, 2023, DOI: 10.22247/ijcna/2023/223420

[104] G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 254-259

[105] J. X. Liao and X. W. Wu, "Resource Allocation and Task Scheduling Scheme in Priority-Based Hierarchical Edge Computing System," 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou, China, 2020, pp. 46-49

[106] Z. Sharif, L. T. Jung, I. Razzak and M. Alazab, "Adaptive and Priority-Based Resource Allocation for Efficient Resources Utilization in Mobile-Edge Computing," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3079-3093, 15 Feb.15, 2023

[107] Xu Liu, Zheng-Yi Chai, Ya-Lun Li, Yan-Yang Cheng, Yue Zeng, Multi-objective deep reinforcement learning for computation offloading in UAV-assisted multi-access edge computing, Information Sciences, Volume 642, 2023, 119154, ISSN 0020-0255

[108] Long, D.; Wu, Q.; Fan, Q.; Fan, P.; Li, Z.; Fan, J. A Power Allocation Scheme for MIMO-NOMA and D2D Vehicular Edge Computing Based on Decentralized DRL. Sensors 2023, 23, 3449, doi.org/10.3390/s23073449.

[109] Rui Wang, Yong Cao, Adeeb Noor, Thamer A Alamoudi, Redhwan Nour, Agent-enabled task offloading in UAV-aided mobile edge computing, Computer Communications, Volume 149, 2020, Pages 324-331, ISSN 0140-3664.

[110] S.M. Asiful Huda, Sangman Moh, Survey on computation offloading in UAV-Enabled mobile edge computing, Journal of Network and Computer Applications, Volume 201, 2022, 103341, ISSN 1084-8045.

[111] W. Jin, G. Liu and H. Gu, "MEC Multi-Objective Task Offloading Algorithm for Joint Energy and Latency Optimization," 2024 10th IEEE International Conference on High Performance and Smart Computing (HPSC), NYC, NY, 2024, pp. 43-48

[112] P. Singh, A. Kaur, G. S. Aujla, R. S. Batth and S. Kanhere, 2020 "DaaS: Dew Computing as a Service for Intelligent Intrusion Detection in Edge-of-Things Ecosystem," in IEEE Internet of Things Journal.

[113] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, 2016.

[114] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, “A Survey on Edge Computing Systems and Tools,” Proc. IEEE, pp. 1–24, 2019.

[115] Y. Ai, M. Peng, and K. Zhang, “Edge computing technologies for Internet of Things: a primer,” Digit. Commun. Networks, vol. 4, no. 2, pp. 77–86, 2018.

[116] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, “Challenges and Opportunities in Edge Computing,” Proc. - 2016 IEEE Int. Conf. Smart Cloud, Smart Cloud 2016, pp. 20–26, 2016.

[117] S. Bhattacharyya, “Research on Edge Computing: A Detailed Study,” Academia.Edu, vol. 2, no. 6, pp. 9–13, 2016.

[118] S. Taherizadeh and V. Stankovski, “Auto-scaling applications in edge computing: Taxonomy and challenges,” ACM Int. Conf. Proceeding Ser., pp. 158–163, 2017.

[119] C. Feature and E. Deep, “Private and Scalable Personal Data Analytics Cloud Deep Learning,” 2018.

[120] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, “Cloudlets Activation Scheme for Scalable Mobile Edge Computing with Transmission Power Control and Virtual Machine Migration,” IEEE Trans. Comput., vol. 67, no. 9, pp. 1287–1300, 2018.

[121] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, “A Multi-Clustering Approach to Scale Distributed Tenant Networks for Mobile Edge Computing,” IEEE J. Sel. Areas Commun., vol. 37, no. 3, pp. 499– 514, 2019.

[122] S. Maheshwari, D. Raychaudhuri, I. Seskar, and F. Bronzino, “Scalability and performance evaluation of edge cloud systems for latency constrained applications,” Proc. - 2018 3rd ACM/IEEE Symp. Edge Comput. SEC 2018, pp. 286–299, 2018.

[123] S. H. Mortazavi, M. Salehe, C. S. Gomes, C. Phillips, and E. De Lara, “Cloud Path: A multi-Tier cloud computing framework,” 2017 2nd ACM/IEEE Symp. Edge Comput. SEC 2017, 2017.

[124] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, “Edge Computing in the Industrial Internet of Things Environment: Software-Defined-Networks-Based Edge-Cloud Interplay,” IEEE Commun. Mag., vol. 56, no. 2, pp. 44–51, 2018.

[125] J. Xie, D. Guo, X. Shi, H. Cai, C. Qian, and H. Chen, “A Fast Hybrid Data Sharing Framework for Hierarchical Mobile Edge Computing,” Proc. - IEEE INFOCOM, vol. 2020-July, pp. 2609– 2618, 2020.

[126] W. Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, “Mobile Edge Cloud-Based Industrial Internet of Things: Improving Edge Intelligence with Hierarchical SDN Controllers,” IEEE Veh. Technol. Mag., vol. 15, no. 1, pp. 36–45, 2020.

[127] M. Uddin, S. Mukherjee, H. Chang, and T. V. Lakshman, “SDN- Based multi-protocol edge switching for IoT service automation,” IEEE J. Sel. Areas Commun., vol. 36, no. 12, pp. 2775–2786, 2018.

[128] T. P. Raptis, A. Passarella, and M. Conti, “Industrial Edge Networks,” vol. 38, no. 5, pp. 915–927, 2020.

[129] B. Charyyev and M. Gunes, “Latency Characteristics of Edge and Cloud,” no. June 2020.

[130] A. Singh, G. S. Aujla, and R. S. Bali, “Intent-Based Network for Data Dissemination in Software-Defined Vehicular Edge Computing,” IEEE Trans. Intell. Transp. Syst., pp. 1–9, 2020.

[131] Z. Ning et al., “Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution,” IEEE Trans. Intell. Transp. Syst., pp. 1–14, 2020.

[132] X. Ren, G. S. Aujla, A. Jindal, R. S. Batth and P. Zhang, "Adaptive Recovery Mechanism for SDN Controllers in Edge-Cloud supported FinTech Applications," in IEEE Internet of Things Journal (2021). https://doi.org/10.1109/JIOT.2021.3064468

[133] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, “Cloudlets Activation Scheme for Scalable Mobile Edge Computing with Transmission Power Control and Virtual Machine Migration,” IEEE Trans. Comput., vol. 67, no. 9, pp. 1287–1300, 2018.

[134] S. Kumar and R. H. Goudar, “Cloud Computing – Research Issues, Challenges, Architecture, Platforms, and Applications: A Survey,” Int. J. Futur. Comput. Commun. vol. 1, no. 4, pp. 356–360, 2012.

[135] Kumar VDA, Kumar A, Batth RS, “Efficient data transfer in edge envisioned environment using artificial intelligence-based edge node algorithm”, Transactions on Emerging Telecommunications and Technologies., 2020.

[136] Q. Zheng and Z. Ping, “Simulation study on latency-aware network in edge computing,” 2019 6th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2019, pp. 150–155, 2019.

[137] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile Edge Computing A key technology towards 5G,” no. 11, 2015.

[138] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. De Foy, and Y. Zhang, “Mobile edge cloud system: Architectures, challenges, and approaches,” IEEE Syst. J., vol. 12, no. 3, pp. 2495–2508, 2018.

[139] Wang, C., Batth, R.S., Zhang, P., Aujla, G.S., “VNE solution for network differentiated QoS and security requirements: from the perspective of deep reinforcement learning”. Computing (2021). https://doi.org/10.1007/s00607-020-00883-w

[140] M. C. Computing, X. Chen, L. Jiao, and W. Li, “Efficient Multi- User Computation Offloading for,” IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, 2016.

[141] C.-H. Hong and B. Varghese, “Resource Management in Fog/Edge Computing,” ACM Comput. Surv., vol. 52, no. 5, pp. 1–37, 2019

[142] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322-2358, Fourth quarter 2017.

[143] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," IEEE Access, vol. 5, pp. 6757-6779, 2017.

[144] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 3860-3873, June 2016.

[145] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23, Oct.-Dec. 2009.

[146] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow Me Cloud: Interworking Federated Clouds and Distributed Mobile Networks," IEEE Network, vol. 27, no. 5, pp. 12-19, Sept.-Oct. 2013.

[147] H. Qiu, K. Ni, R. Xu, and H. Wang, "Deep Learning for Mobile Edge Computing: Applications, Trends, and Challenges," IEEE Communications Magazine, vol. 56, no. 8, pp. 16-21, Aug. 2018.

[148] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G Wireless Backhaul Networks: Challenges and Research Advances," IEEE Network, vol. 28, no. 6, pp. 6-11, Nov.-Dec. 2014.

[149] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making Smartphones Last Longer with Code Offload," in Proc. MobiSys, 2010, pp. 49-62.

[150] X. Liu, Z. Qin and Y. Gao, "Resource Allocation for Edge Computing in IoT Networks via Reinforcement Learning," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6.

[151] Annisa Sarah, Gianfranco Nencioni, Md. Muhidul I. Khan, Resource Allocation in Multi-access Edge Computing for 5G-and-beyond Networks, Computer Networks, Volume 227, 2023, 109720, ISSN 1389-1286.