Efficient Resource Management Framework for Critical Healthcare Applications in Integrated Edge-Fog-Cloud Environments using Blockchain based Federated Learning Methods

A Thesis

Submitted in partial fulfilment for the Degree of

Doctor of Philosophy under the

Faculty of Engineering

by

SHINU M. R.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

BENGALURU CAMPUS – 560035 (INDIA)

July 2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

BENGALURU CAMPUS -560 035

DECLARATION

I, **SHINU M. R.**, **Reg. No: BL.EN.R4CSE18002**, hereby declare that this thesis entitled "**Efficient Resource Management Framework for Critical Healthcare Applications in Integrated Edge-Fog-Cloud Environments using Blockchain based Federated Learning Methods**" is the record of the original work done by me under the guidance of **Dr. Supriya M.**, Associate Professor, Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru Campus and **Prof. Rajkumar Buyya**, Professor and Director of the Cloud Computing and Distributed Systems Lab, The University of Melbourne, Australia. To the best of my knowledge, this work has not formed the basis for the award of any degree/diploma/associateship/fellowship or a similar award to any candidate in any University.

Place: Bengaluru **Signature of the Student**
Date: 17-07-2024

COUNTERSIGNED

Dr. Supriya M.
Thesis Advisor,
Associate Professor,
Department of Computer Science and Engineering,
Amrita Vishwa Vidyapeetham,
Bengaluru Campus, India.

Prof. Rajkumar Buyya
Thesis Co-Advisor,
Professor & Director,
CLOUDS Lab,
The University of Melbourne,
Australia.

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

BENGALURU CAMPUS -560 035

BONAFIDE CERTIFICATE

This is to certify that the thesis entitled "**Efficient Resource Management Framework for Critical Healthcare Applications in Integrated Edge-Fog-Cloud Environments using Blockchain based Federated Learning Methods**" submitted by **SHINU M. R., BL.EN.R4CSE18002** for the award of the **Degree of Doctor of Philosophy** under the **Faculty of Engineering** is a bonafide record of the work carried out by her under our guidance and supervision at Department of Computer Science & Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru Campus.

Dr. Supriya M.
Thesis Advisor,
Associate Professor,
Department of Computer Science and Engineering,
Amrita Vishwa Vidyapeetham,
Bengaluru Campus, India.

Prof. Rajkumar Buyya
Thesis Co-Advisor,
Professor & Director,
CLOUDS Lab,
The University of Melbourne,
Australia.

Contents

Acknowledgements

The fulfillment experienced upon successfully completing any task would remain incomplete without acknowledging the individuals who made it achievable. Their unwavering support and guidance served as a continuous source of inspiration throughout the journey of this thesis work.

I humbly offer my heartfelt respects at the divine lotus feet of "AMMA", Mata Amritanandamayi Devi, whose blessings graciously accompanied me throughout the entirety of this research endeavor.

I owe my gratitude to Prof. Manoj P., (Director, Amrita School of Engineering, Bengaluru), Dr. Bharat Jayaraman (Dean, Amrita School of Computing, Bengaluru), Dr. Sriram Devanathan (Principal, Amrita School of Engineering, Bengaluru) and Dr. Gopalakrishnan E. A., (Principal, Amrita School of Computing, Bengaluru) for their support and inspiration. I would like to use this opportunity to express my sincere gratitude to Amrita Vishwa Vidyapeetham for providing the facilities and necessary infrastructure to conduct the research.

It is with great pleasure that I express my gratitude and indebtedness to my thesis advisor, Dr. Supriya M., Associate Professor in the Department of Computer Science and Engineering at Amrita School of Computing, Bengaluru. Her invaluable guidance, encouragement, moral support, and affection have been instrumental throughout the thesis work. Her profound knowledge in the research area has served as a guiding light, enabling me to navigate through challenges and achieve milestones in this endeavor.

I extend my heartfelt appreciation to Prof. Rajkumar Buyya, Professor & Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory The University of Melbourne, Australia., my thesis co-advisor, for his exceptional support and guidance throughout the research process. His expertise and encouragement have been invaluable in facilitating the progress and success of this endeavor.

I am grateful for the valuable suggestions and insightful discussions provided by my doctoral committee members, Dr. Deepa K. (Convener), Dr. Prakash G. (Subject Expert), and Dr. Nagaraja K. V. (Faculty from Minor Area), during the review presentations. Their expertise and guidance have greatly contributed to the enhancement of my work.

I extend my deepest gratitude to Amrita Vishwa Vidyapeetham for providing the infrastructure to facilitate the study. I thank my friends and well-wishers at Amrita School of Computing, Bengaluru, for their unwavering encouragement and support.

I am immensely grateful to my husband for his unwavering support, encouragement, and understanding throughout this journey. To my parents, I owe a debt of gratitude for their love, guidance, and sacrifices that have shaped my academic path. And to my children, your boundless love and patience have been a constant source of motivation and joy.

(SHINU M. R.)

List of Figures

List of Tables

Abbreviations

AE	Auto-Encoder
AHP	Analytical Hierarchical Process
ANN	Artificial Neural Network
ANF	Algebraic Normal Form
ACO	Ant Colony Optimization
BC	Blockchain
BCFL	Blockchain Federated Learning
CAL	Current Average Delay
CBD	Central Business District
CFL	Centralized FL
DC	Data Center
DF	Decentralized FL
DP	Differential Privacy
DSS	Decision Support Systems
DT	Decision Tree
DVFS	Dynamic Voltage and Frequency Scaling
EAMC	Exigency Alert Mobile Cloud
ECG	Electrocardiogram
EC	Edge Computing
ED	Edge Device
EMS	Emergency Medical Services
EXT	The Execution Time
FCFS	First Come First Serve
FLaaS	FL as a Service
FL	Federated Learning
FPA	Flower Pollination Algorithm
GA	Genetic Algorithm
HFL	Hierarchical Federated Learning
HIPAA	Health Insurance Portability and Accountability Act
IoHT	Internet of Healthcare Things
IoT	Internet of Things
LBS	Load Balancing Scheme
LR	Linear Regression
LTE	Long-Term Evolution
LSTM	Long Short-Term Memory

MCI	Microcomputing Instances
MEC	Mobile Edge Computing
MGA	Modified Genetic Algorithm
MFPA	Modified Flower Pollination Algorithm
MD-QoE	Multidimensional QoE
NSGA	Non-Dominated Sorting Genetic Algorithm
NCDs	Non-Communicable Diseases
NFC	Near-Field Communications
NIMH	Nature-Inspired Metaheuristic
NIST	National Institute of Standards and Technology
ORP	Optimal Resource Provisioning
PF/EXP	Proportional Fairness and Exponential Proportional Fairness
PSO	Particle Swarm Optimization
QoS	Quality of Service
RF	Random Forest
RSU	Road Side Unit
SI	Swarm Intelligence
SDM	Smart Decision Making
SGD	Stochastic Gradient Descent
SPEA	Strength Pareto Evolutionary Algorithm
SLA	Service Level Agreement
SVM	Support Vector Machine
WHO	World Health Organization

Abstract

A critical system in real-time generally manages any environment by receiving the input, processing it, and producing output data while meeting specific time constraints. In such systems, deadlines must always be satisfied irrespective of the load of the system. As the number of real-time applications that need smart connections among each other increases, the Internet of Thing (IoT) challenges will also increase. The new evolving technology that may help in time-critical real-time IoT-based systems is the use of Edge/Fog. To make use of the advantages of edge/fog computing for real-time critical applications, we propose an integrated system of edge, fog, and cloud computing environments for the healthcare sector. By eliminating irrelevant data at the Edge/Fog nodes, the proposed architecture saves time as well as minimizes the amount of data that must be transferred to the cloud.

Resource management is one of the primary challenges in a real-time environment since the demand for resources grows dynamically and needs rapid provisioning and processing. Resource management techniques in edge/fog are challenging because the modules of analytic applications are moved to each edge device to minimize the delay and avoid congestion. This research focuses on developing efficient Edge-Fog-Cloud-based resource management for medical applications to avoid delay and improve the efficiency of smart healthcare systems. To maximize the utilization of these resources and improve the efficiency of applications, efficient scheduling and resource allocation are required. To avoid over-provisioning and under-provisioning, dynamic resource provisioning, an effective method of preparing resources based on changes in the workload of IoT applications, is required. Due to the vast solution space, scheduling in fog/edge computing is classified as an NP-hard problem, which means it takes a long time to discover an optimal solution. No algorithms can handle these issues in polynomial time and yield optimal results. Finding a sub-optimal solution in a short period is preferred in such scenarios. To address such issues, meta-heuristic-based strategies have been experimented in our proposed approach to generate near-optimal solutions in a reasonable amount of time. By reducing the time to diagnose the critical state of patients, the proposed model guarantees prompt medical assistance.

Massive data collection in modern systems has paved the way for data-driven machine learning, a promising technique for creating reliable and robust statistical models. Combining the data into centralized storage to develop a reliable learning model involve few concerns related to privacy, ownership, and strict rules. Due to the heterogeneous and dynamic nature of critical medical IoT applications in such Edge/Fog scenarios, the privacy of patients has become a crucial problem. The main challenges that exist in the current smart healthcare applications are security and privacy. Security and privacy challenges can be rectified using federated learning and blockchain technologies. It is self-evident that the samples in the typical machine learning centralized server paradigm have vastly different probability distributions of data supplied by each user. As a result, the standard model fails to personalize. Federated Learning is model training on diverse, dispersed networks while maintaining privacy. There has been an increase in attention to federated learning since its introduction in 2016, with a broad range of applications, challenges, and concerns associated with this unique paradigm, motivating us to perform the research. We propose the integration of Federated Learning for distributed Edge-Fog-Cloud architecture in the IoT smart healthcare sector. It deploys the Federated Learning model at the Edge, Fog, and Cloud layers for performance comparison. The parameters considered for performance evaluation are energy consumption, network usage, cost, execution time, and latency.

Blockchain technology is triggering a very sensation and prompting across various industries. Given the massive progress in blockchain technology, assessing the feasibility of existing blockchain technologies for use in various new and unsolved fields is necessary. Although several studies have focused on various applications of blockchain technology and federated learning, a framework for IoT resource management using these technologies in integrated edge, fog, and cloud computing environments is yet to be explored. The research goal is to develop a general resource provisioning framework that provide efficient resource management for heterogeneous, unpredictable dynamic resource demands in computing paradigms like Edge, Fog, Cloud, Mobile Cloud, and Mobile Edge computing environments using blockchain-based federated learning technologies.

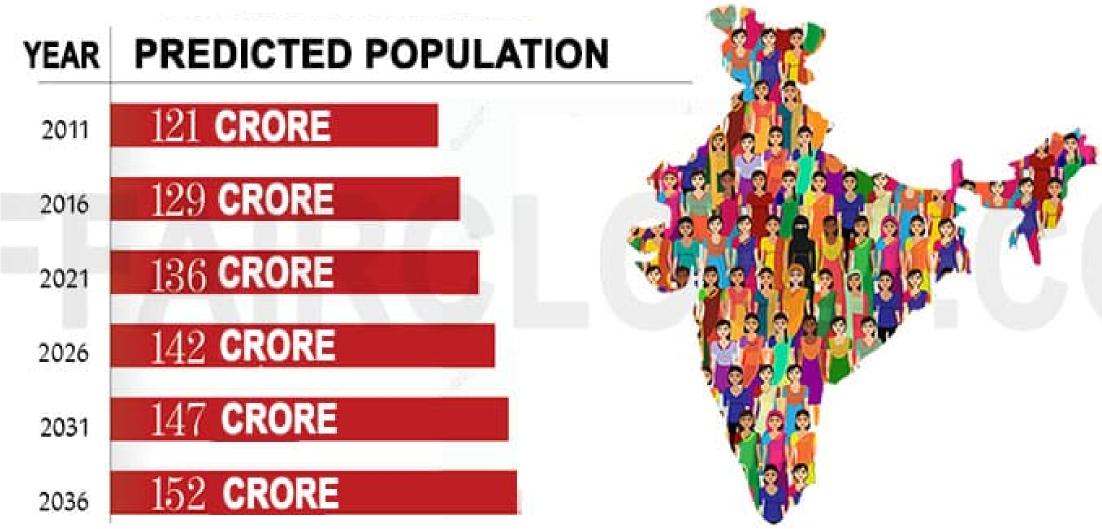
Chapter 1

Introduction

1.1 Overview

The United Nations analysis indicates a continued growth in the Indian population for the next few decades. According to the UN's World Population Prospects, India is projected to reach a substantial population of 1.66 billion people by the year 2050 [1]. This forecast highlights the persistent demographic expansion in the country and emphasizes the importance of addressing the associated challenges and opportunities, particularly in areas such as infrastructure, healthcare, and education, to ensure sustainable development in the coming years. India has one of the largest population globally, surpassed only by China. As of the latest available data, India's population is over 1.3 billion and is expected to grow more as can be seen in Figure 1.1 [2]. Managing and catering to the needs of such a vast population is a complex task that requires strategic planning and effective policies.

The demographic makeup of India's population reveals a distinctive distribution, with 31% representing individuals aged 0-14 years, 62% falling within the crucial working age between 15-60 years, and 7% constituting the elderly population aged 60 years and above as presented in Figure 1.2. This breakdown underscores the significant presence of children and adolescents, emphasizing the critical need for substantial investments in healthcare, education, and social infrastructure to support their development. The majority in the 15-60 age group highlights a robust workforce, crucial for economic productivity, while the 7% aged 60 and above signals the presence of an aging population, necessitating a heightened focus on healthcare and robust support systems. This demographic snapshot provides indispensable insights for policymakers, guiding them to prioritize healthcare strategies to address the unique needs of each age group, thereby fostering the overall well-being of society [2].

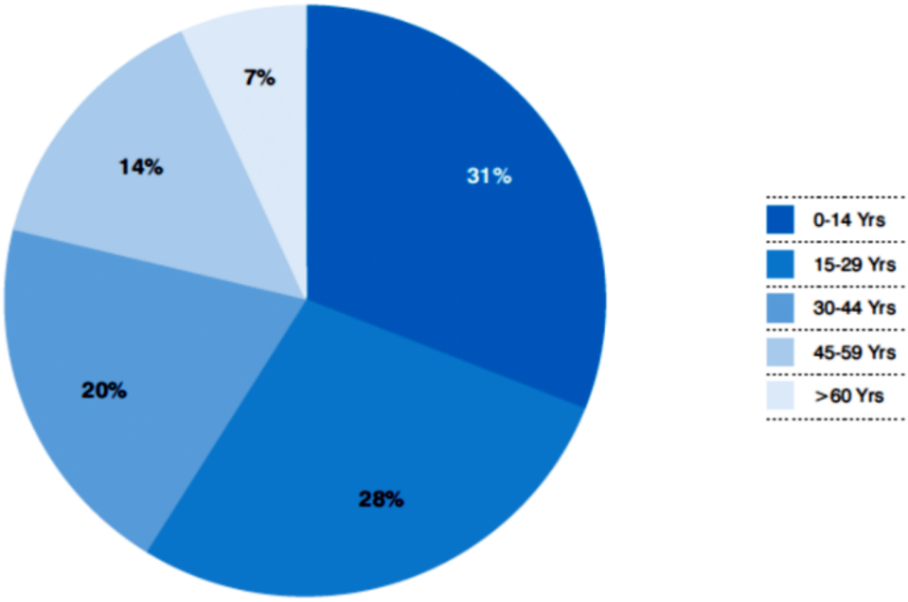


Source: www.affairscloud.com
Figure 1.1: Indian Population

Upon analyzing the health reports of India, it is evident that a substantial portion of the population faces various health challenges. Specifically, 26 percent of individuals suffer from cardiovascular conditions, 2 percent from diabetes, 13 percent from respiratory issues, 7 percent from cancer, 28 percent from communicable and external diseases, 12 percent from Non-Communicable Diseases (NCDs), and another 12 percent from injuries as depicted in Figure 1.3. These findings underscore the diverse health concerns within the population and emphasize the need for targeted healthcare interventions and public health strategies to address the prevalent health conditions effectively [3].

India has made significant progress in the healthcare industry over the last few decades, resulting in increased life expectancy and reduced infant mortality. These achievements signify substantial improvements in the overall health and well-being of the population [5]. The advancements in healthcare services, interventions, and awareness have contributed to longer life spans and a decrease in infant mortality rates, showcasing the positive impact of healthcare initiatives in the country. Continued efforts in this sector remain crucial for sustaining and further enhancing these positive health outcomes for the people of India.

However, when it comes to emergency medical care, India faces challenges. The availability and accessibility of urgent medical services may not be as robust as desired. This gap in emergency healthcare infrastructure points to an area that needs attention and improvement. Strengthening emergency medical care systems



Source: https://en.wikipedia.org/wiki/Demographics of India
Figure 1.2: Distinctive Distribution of Indian Population

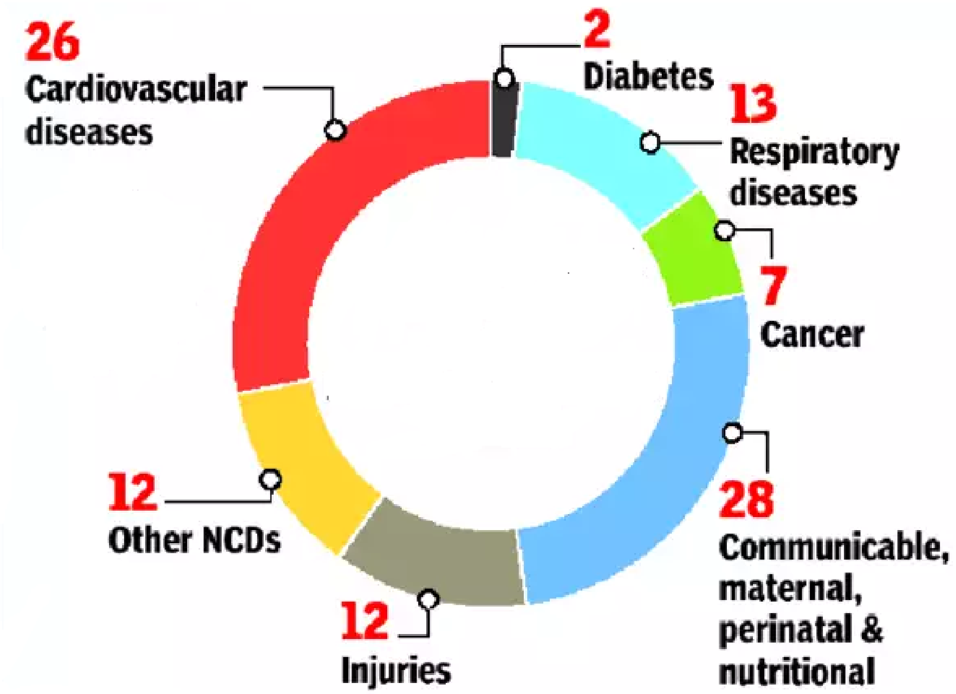
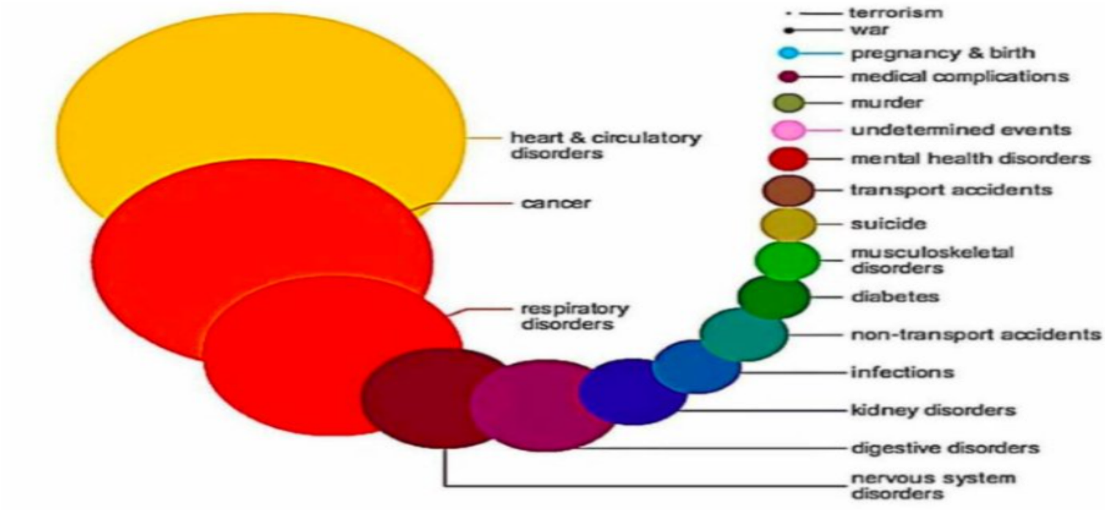


Figure 1.3: Health report of Indian population [4]



Source: www.phfi.org/india-health-of-the-nations-states
Figure 1.4: Leading causes of death in India

can play a vital role in ensuring timely and effective response to critical health situations, contributing to an overall enhancement of the healthcare landscape in India [6].

Indeed, the limitations in emergency medical care were notably evident during the challenging times of the COVID-19 pandemic. The surge in cases exposed gaps in the healthcare system, particularly in terms of emergency response and critical care infrastructure. The overwhelming demand for medical services, including oxygen support and ICU facilities, highlighted the need for reinforcing and expanding emergency healthcare capabilities. The pandemic experience emphasizes the importance of proactive measures and investments in emergency medical care to enhance the country's readiness in responding effectively to unforeseen health crises. Addressing these challenges will be crucial for building a resilient healthcare system in India.

The lack of a robust emergency response system has proven to be a critical factor that has cost millions of lives in India. During times of crisis, the absence of a well-coordinated and timely emergency response infrastructure has contributed to challenges in delivering life-saving medical care. This gap in the system has resulted in tragic consequences, underscoring the urgent need for significant improvements in India's emergency medical services. Strengthening and expanding emergency response systems can play a vital role in preventing unnecessary loss of life during health emergencies and ensuring a more resilient healthcare setup for the future.

It is a widely acknowledged fact that Emergency Medical Services (EMS) often face challenges in providing care during the critical "golden hour" of emergencies.

The golden hour is the crucial first 60 minutes after a traumatic injury or a medical emergency when swift and effective medical intervention significantly improves the chances of a positive outcome. In many cases, limitations in infrastructure, accessibility, and response times may hinder EMS from delivering timely care during this critical period. Recognizing and addressing these challenges is essential for enhancing emergency medical services and ensuring that life-saving interventions can be administered promptly, especially during the crucial initial moments of an emergency [7].

Among the EMS, Heart and cardiovascular diseases stand as the foremost global causes of death, taking the lives of approximately 17.9 million people each year, as reported by the World Health Organization (WHO) [8]. A survey conducted an analysis of the factors contributing to mortality in India, the results of which are illustrated in Figure 1.4. This staggering statistic underscores the significant impact of cardiovascular diseases on public health. According to the Indian Heart Healthcare Association, a noteworthy statistic reveals that 50 percent of heart attacks in India occur in individuals below the age of 50. This alarming trend indicates a concerning prevalence of heart-related issues among a relatively younger demographic. Understanding and addressing the factors contributing to heart attacks in this age group are imperative for devising effective preventive measures and healthcare strategies. Lifestyle factors, stress, and genetic predispositions may play significant roles in this trend, highlighting the importance of promoting heart-healthy habits and early screenings to mitigate the risk of heart attacks among the younger population in India.

Indeed, the Electrocardiogram (ECG) is a crucial diagnostic tool used to detect various heart problems. It records the electrical activity of the heart over a period of time, producing a visual representation of the heart's rhythm and function. Abnormalities in the ECG waveform can indicate conditions such as arrhythmia, heart attack, and other cardiac related issues. ECGs are commonly employed in clinical settings for both routine check-ups and emergency situations, providing valuable information that aids healthcare professionals in assessing and diagnosing heart-related problems. This non-invasive and widely used technique plays a pivotal role in the early detection and monitoring of heart conditions, contributing to effective medical interventions and improved patient outcomes.

ECG is a standardized and widely accepted medical test, ensuring consistency and reliability in its application across healthcare settings. Additionally, with advancements in technology, ECG sensors can now be seamlessly integrated into wearable devices, facilitating unobtrusive and convenient health monitoring in real-world settings. This integration aligns with the growing trend towards personalized and remote healthcare, providing individuals with continuous cardiac

Figure 1.5: IoT healthcare [11]

monitoring for improved overall well-being [9, 10].

These wearable devices are integral components of the Internet of Things (IoT) healthcare landscape, contributing to the collection and transmission of real-time health data as depicted in Figure 1.5. By seamlessly integrating with IoT platforms, wearable devices enable continuous monitoring of various health parameters, including ECG, and facilitate the transmission of this data to healthcare providers or centralized systems. This interconnected approach allows for remote patient monitoring, timely intervention, and personalized healthcare strategies. Wearable devices in IoT healthcare empower individuals to actively engage in their well-being by providing real-time insights into their health conditions, promoting preventive measures, and facilitating more efficient and patient-centric healthcare delivery.

The integration of the IoT in healthcare brings forth numerous advantages, revolutionizing the traditional healthcare paradigm. By enabling remote patient monitoring, IoT facilitates real-time tracking of vital signs, medication adherence, and health metrics, beneficial for managing chronic conditions. The data-driven nature of IoT enhances diagnostic accuracy and healthcare decision-making, while automated workflows and smart devices contribute to improved efficiency in healthcare processes [12]. Wearable devices and IoT applications empower patients to actively engage in their well-being, promoting informed decision-making and adherence to treatment plans. Additionally, IoT supports preventive healthcare by identifying potential issues, contributing to cost savings through reduced hospital readmissions and efficient resource utilization. Interoperability among different healthcare systems, along with the facilitation of telehealth services, ensure

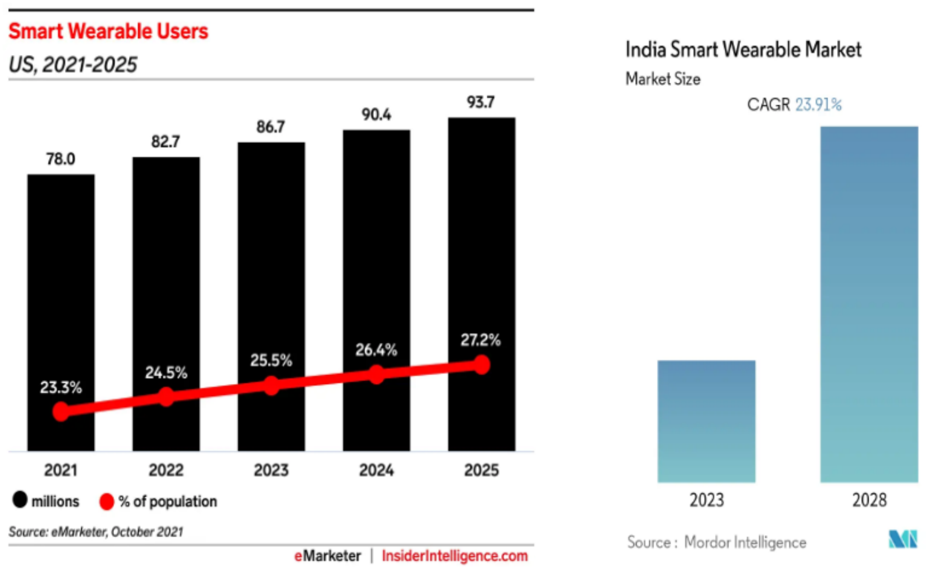


Figure 1.6: Wearable device market growth [13]

seamless connectivity and accessibility to healthcare. Ultimately, IoT in healthcare provides personalized treatment plans, quick responses to emergencies, and a transformative approach to healthcare delivery. Addressing challenges related to data security and privacy is crucial to fully unlock the vast potential of IoT in healthcare.

As can be seen in Figure 1.6, the global wearable device market has been experiencing significant growth, driven by increasing consumer interest in health and fitness tracking, smartwatches, and other wearable technologies. The adoption of wearables has become widespread, with various companies offering a range of devices with advanced features. In India, the wearable device market has also seen substantial growth, propelled by factors such as rising health awareness, increasing disposable income, and the popularity of fitness tracking. Major players in the global wearable market, including companies like Apple, Samsung, Fitbit, and others, have a presence in the Indian market. The market encompasses a variety of devices, including smartwatches, fitness trackers, and health monitoring wearables.

In recent instances, smart wearable devices have demonstrated their potential to save lives through early detection and rapid response capabilities. Wearables equipped with heart rate monitoring features have alerted users to irregular heartbeats, prompting timely medical intervention and potentially preventing serious cardiac events. Additionally, fall detection features beneficial for the elderly, have enabled swift alerts to emergency contacts or services in the event of a fall. These real-time monitoring capabilities extend to chronic health conditions such as diabetes and epilepsy, where wearables offer continuous tracking and immediate alerts, empowering individuals to manage their health effectively. As the adoption


ET THE ECONOMIC TIMES Feb 20, 2023, 04:09:20 PM IST

Apple Watch saves owner's life from fatal internal bleeding after nap, here's what happened

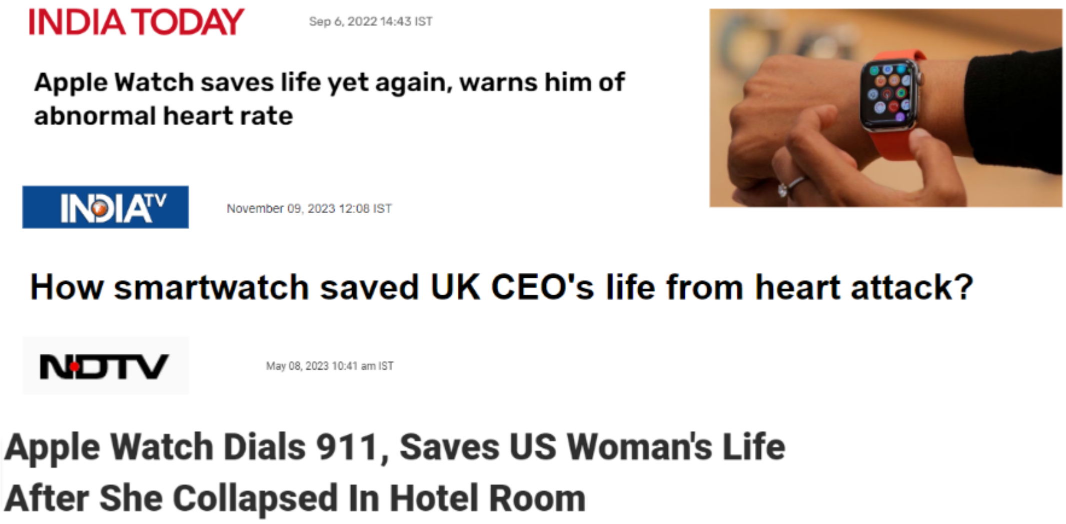
INDIA TODAY Mar 11, 2023 16:51 IST

Apple Watch saves life of a 36-year-old user suffering from heart condition

Source: www.indiatoday.in/technology/news
Figure 1.7: News Articles Highlighting How Wearable Devices Have Contributed to Saving Lives in Recent Years [16]

of smart wearables continue to grow, their role in facilitating early detection, providing timely alerts, and aiding in the management of various health conditions highlight their potential to contribute significantly to healthcare and potentially save lives [14]. The news articles in Figures 1.7 and 1.8 depicts the above facts. The study conducted by doctors from AIIMS Delhi highlights a concerning trend where a small proportion of patients experiencing cardiac and stroke emergencies have been provided with timely support as they manage to reach healthcare facilities early. Such early access to medical intervention is crucial in cases of cardiac and stroke emergencies, as prompt treatment significantly improves outcomes. The findings underscore the need for increased awareness, education, and improved emergency response systems to ensure that individuals recognize the symptoms of such emergencies and seek timely medical assistance. This study emphasizes the importance of public health initiatives to enhance community awareness about the signs of cardiac and stroke events, ultimately leading to more effective and timely interventions to save lives, which can be achieved with the help of smart decision-making applications [15].

The preceding paragraphs highlight the essential need for an intelligent decision-making module in the detection of heart abnormalities. Presently, these decisions are happening in the cloud as the IoT devices store the data in the cloud data centers. The decision making in the cloud has many challenges, among which latency plays a major role [18]. Integrating fog and edge computing in conjunction with cloud services presents a formidable solution to mitigate delays encountered



Source: www.timesofindia.indiatimes.com

Figure 1.8: News Articles Highlighting How Wearable Devices Have Contributed to Saving Lives in Recent Years [17]

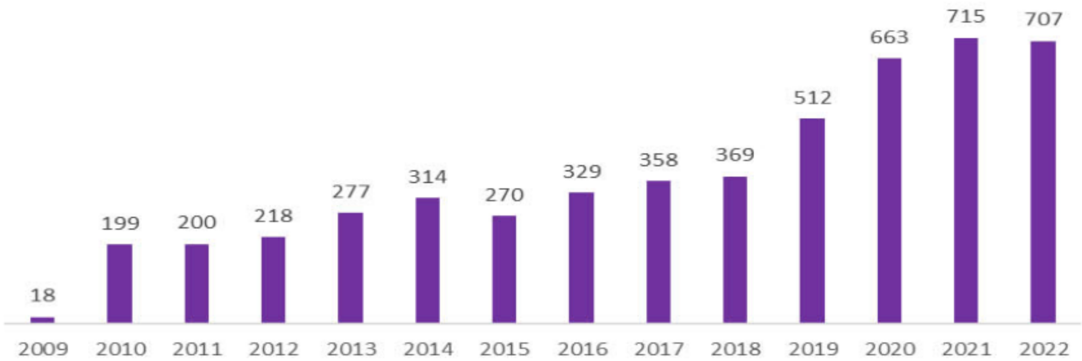
by smart decision-making modules. By distributing computational tasks across the network layers, these paradigms capitalize on proximity to data sources, local processing capabilities, and bandwidth optimization. Edge devices, positioned in close proximity to data sources, engage in a preliminary analysis, reducing the need for transmitting vast amounts of data to the cloud. Meanwhile, fog computing, situated between edge devices and the cloud, facilitates decentralized processing, alleviating the computational burden on the cloud. This optimized architecture enhances response times and allows for real-time decision-making at the source [19]. The synergy of edge, fog and cloud computing thus creates a more responsive and efficient infrastructure, crucial for applications such as healthcare automation where minimizing latency is paramount.

The wearables and IoT devices generate a large amount of data continuously, and securing that is a big challenge. In the context of handling medical data, a thorough examination of issues associated with the security and privacy of healthcare information has been conducted. Medical data breaches worldwide pose significant risks to patient confidentiality as sensitive health information becomes vulnerable to unauthorized access. The impact extends beyond individual privacy concerns, potentially compromising healthcare systems' integrity and eroding trust in the safeguarding of critical medical data on a global scale. The impact of healthcare data breaches in India is a critical concern with far-reaching implications for patient privacy and the overall integrity of medical information systems [20]. A notable instance is the ransomware attack on AIIMS Delhi, exemplifying the vul-

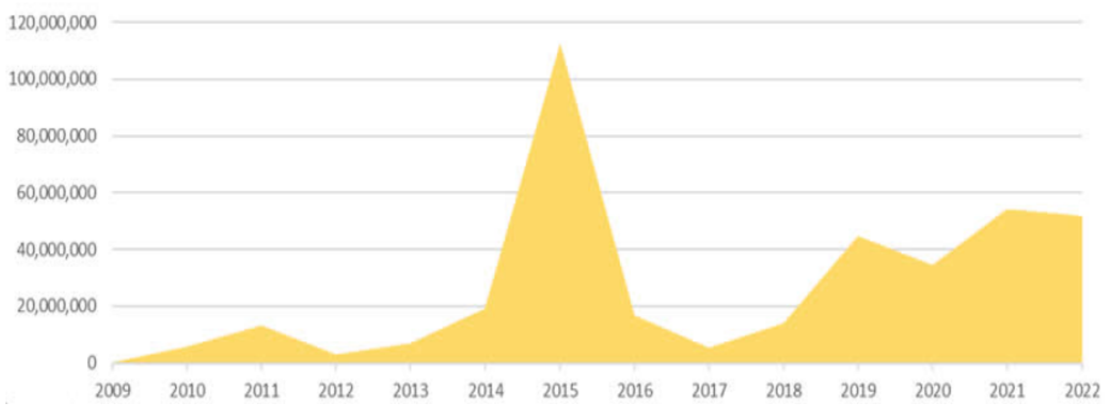
Source: www.hindustantimes.com/cities/delhi-news
Figure 1.9: AIIMS Data Breach: Examining the Latest News Articles on Security Incidents [21]

nerability of healthcare institutions to cyber threats. The news article pertaining to this can be seen in Figure 1.9. Such breaches not only compromise sensitive patient data but also pose severe risks to public health. Figures 1.10 and 1.11 presents the growth of such data breaches in the recent past, which motivates us to develop a mechanism to secure the data that is stored in the cloud.

The global challenges of medical data breaches can be effectively addressed through the integration of blockchain-based federated learning [23]. This innovative approach ensures enhanced security by leveraging the decentralized and cryptographic features of blockchain, creating an immutable and transparent ledger for medical data. Federated learning (FL) complements this by enabling collabo-



Source: www.hipaajournal.com/healthcare-data-breach-statistics
Figure 1.10: Healthcare Data Breaches Over the Last Years



Source: www.hipaajournal.com/healthcare-data-breach-statistics
Figure 1.11: Healthcare Data Breaches and Their Human Impact Over the Last Years [22]

rative model training without exposing raw data, preserving patient privacy [24]. The combination of these technologies establishes a robust framework with features such as an immutable audit trail, patient empowerment, and decentralized validation, fostering trust and resilience in healthcare data management systems worldwide. This paradigm shift not only mitigates the risk of unauthorized access but also paves the way for a more secure and privacy-centric healthcare ecosystem on a global scale. Figure 1.12 serves to visually depict the earlier discussed information.

Based on the foregoing survey, we have formulated our research problem as follows: Efficient Resource Management Framework for Critical Healthcare Applications in Integrated Edge-Fog-Cloud Environments using Blockchain based Federated Learning Methods

In connection with the formulated research problem the subsequent paragraphs discuss the technical details associated with it, along with the background needed to the solution.

The concept of virtualization constitutes a fundamental step in the realm of cloud computing, playing a critical role in resource optimization and management. Virtualization involves the abstraction of physical hardware resources, such as computing power, storage, and networking components, to create virtual instances that can be dynamically provisioned and allocated to users as needed. Users can access these virtualized resources through a pay-as-you-go model, gaining the ability to scale their computing needs based on demand without the constraints of physical hardware limitations. Beyond virtualization, the cloud model offer benefits such as ease of access, availability of resources on demand, efficiency, substantial storage capacity, and processing capabilities [25, 26]. Nonetheless, despite these advantages, the cloud model remains vulnerable to attacks, partly

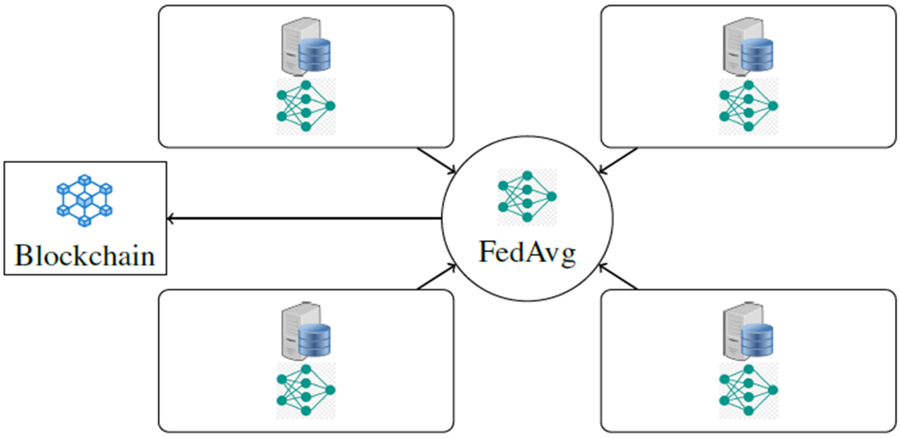


Figure 1.12: Basic Architecture diagram - Blockchain based Federated learning

due to the delays arising from the geographical gap between end devices and computational resources. Additionally, the impracticality of streaming massive volumes of data between nodes and the cloud presents significant challenges [27]. These drawbacks result in cloud computing being less effective for real-time critical applications. To address this challenge, the emergence of edge/fog computing becomes a pertinent consideration. Edge/Fog computing extends the cloud computing paradigm to encompass every layer of the networking infrastructure, from end devices to the cloud. This approach involves redistributing resources, including computation and storage, across the network layers, thereby utilizing cloud services to support interconnected embedded devices [28].

The global edge/fog computing market size was valued at USD 11.99 billion in 2022 and is projected to reach USD 139.58 billion by 2030 [29]. With the integration of IoT, 5G networks, and advanced networking technologies, the trajectory of edge/fog computing's growth becomes evident [6]. By 2025, the number of IoT-connected devices worldwide is expected to reach 30.9 billion, necessitates for robust computing capabilities for effective data processing and storage [30,31]. This surge in computational demand underscores the need for a proficient edge/fog architecture to ensure swift data processing. The edge/fog architecture involves configuring both hardware and software components to establish a functional IoT network. This process entails identifying crucial application requirements suitable for computing, followed by mapping these specifications onto a partitioned network of well-designed edge/fog nodes. Structurally, edge/fog networks exhibit a hierarchical topology, creating a distributed computing system [32]. IoT, characterized by interconnected smart embedded devices, particularly aligns with the principles of edge/fog computing [4]. Given that IoT devices generate substantial data volumes at high frequencies, efficient data processing is pivotal for time-critical applications. Edge/Fog computing serves as an intermediary connecting IoT and the cloud, offering specialized processing capabilities tailored to specific needs. The benefits include latency reduction, reduced data traffic, optimized bandwidth usage, efficient utilization of networking devices, and geographic distribution of devices [33]. The integration of IoT and edge/fog computing enhances the overall efficiency of systems while enhancing the performance of critical IoT applications [25].

Edge/Fog computing finds applications in diverse domains, including healthcare, military domain, atomic reactors, and augmented reality. To enable such applications, efficient resource management becomes imperative. Resource management is a pivotal function in both cloud and edge/fog systems, as suboptimal management can negatively impact performance, costs, and overall system functionality. Real-time environments pose unique challenges due to dynamic resource demands that necessitate rapid provisioning and processing [34]. Factors like heterogeneity, on-demand access, pay-as-you-use models, and quality of service (QoS) considerations contribute to the complexity of resource management. It encompasses various components, such as resource provisioning, which involves selecting, deploying, and managing runtime software and hardware resources to optimize application performance based on service level agreements [35]. Despite its importance, resource provisioning faces challenges like task fragmentation, which might not align with user expectations [13]. Another crucial aspect of resource management is resource allocation, which facilitates on-demand allocation of resources by end devices, occupying a central position in the middle layer of the computing architecture. Resource management within edge/fog computing is intricate due to the movement of analytic application modules to edge devices for minimizing delays and congestion. Resource provisioning entails the allocation and distribution of resources, including computational power, storage, network bandwidth, and others, to meet the requirements of various applications or services. In contrast, resource scheduling involves determining when and how to allocate resources to tasks or jobs based on their priority and demands. Combining resource provisioning methods with resource scheduling policies facilitates efficient resource allocation to applications, especially in distributed scenarios, thus enhancing resource utilization and application efficiency. Dynamic resource provisioning, which involves adjusting resources based on changes in IoT application workloads, is particularly essential to avoid over-provisioning or under-provisioning of resources [36].

In the realm of IoT critical healthcare applications, the significance of key parameters such as energy consumption, cost, latency, network usage, and execution time is paramount. These parameters collectively dictate the operational efficiency, economic viability, and patient-centric efficacy of healthcare systems. Energy consumption directly impacts the longevity of battery-powered medical devices, ensuring uninterrupted monitoring and treatment. Cost considerations are crucial for resource allocation and scalability, enabling cost-effective healthcare delivery. Low latency is essential for real-time data transmission and swift decision-making, particularly in critical situations. Efficient network usage minimizes data congestion and ensures seamless communication, essential for reliable patient data exchange. Lastly, optimized execution time guarantees the timely processing of medical information, influencing diagnosis accuracy and treatment outcomes. Therefore, a holistic approach that balances these parameters is indispensable for creating IoT-based healthcare solutions that are dependable, cost-efficient, and patient-focused [37].

Initially, the proposed research problem uses multi objective optimization to effectively provision the resources. This methodology addresses the intricate interplay of key parameters, including energy consumption, cost, latency, network usage, and execution time, which collectively define the performance landscape of such systems. By integrating these diverse objectives, multiobjective optimization techniques provide a systematic framework to attain resource provisioning strategies that effectively balance the trade-offs among these parameters. Through the adept utilization of advanced algorithms and computational intelligence, these methods empower decision-makers to derive optimal solutions that ensure minimal energy usage, cost-effectiveness, low latency, efficient network utilization, and swift execution times. This not only enhances the overall quality of edge/fog computing but also contributes to shaping a sustainable and well-performing technical ecosystem. The proposed weighted sum method for edge/fog applications provide a practical and flexible approach for optimizing energy, cost, network use, execution time, and latency. Its simplicity allows for straightforward implementation and trade-off analysis between objectives, making it valuable for decision-makers. Adjusting the weights assigned to each objective offer flexibility and allow for real-time adaptation, which is crucial in dynamic edge/fog environments. Additionally, the method's efficiency and ease of integration make it suitable for resource-constrained settings, while its interpretability aids in understanding the implications of changing optimization criteria. Overall, the weighted sum method is an effective tool for achieving a balanced solution that meets the diverse requirements of edge/fog computing [38, 39].

Given the extensive solution space, scheduling in edge/fog/cloud computing is classified as an NP-hard problem, implying that finding an optimal solution takes considerable time. Although algorithms cannot provide optimal results within polynomial time, finding a suboptimal solution swiftly is preferred in such scenarios. To address these challenges, metaheuristic-based strategies have been explored to generate near-optimal solutions within a reasonable timeframe [40]. Metaheuristic scheduling algorithms, such as the flower pollination algorithm (FPA) and genetic algorithm (GA), draw inspiration from natural processes to enhance scheduling efficiency. The genetic algorithm, a prominent population-based algorithm known for its ease of use and effectiveness across diverse problems, employ chromosomes consisting of genes representing potential solutions. The initial population, selected at random, serves as the algorithm's starting point. A fitness function evaluates the suitability of a chromosome within its environment. Single-point crossover and mutation methods create a new population iteratively, continuing until a sufficient number of offspring is generated. While heuristic algorithms use the objective function to select the best solution, genetic algorithms rely on the fitness function to determine the optimal solution [41]. The flower pollination algorithm, designed to mimic the pollination process of flowering plants, addresses multiobjective optimization problems through global and local pollination processes. By considering the flower and its pollen gametes, this algorithm provides a robust approach to optimization. The benefits of the flower pollination algorithm include a simpler flower analogy and lightweight computing reliant on a single control factor [42]. Our research advocates for the effective provisioning of resources in edge/fog and cloud computing by employing modified genetic algorithm (MGA) and the modified flower pollination algorithm (MFPA).

IoT applications use various technologies to connect, manage, and operate IoT smart devices. Microservices, a service-oriented architecture, have attracted much interest nowadays. It is an emerging technology based on the microservices concept to enable services with the smallest granularities that perfectly complement the distributed nature of IoT devices [43]. Each microservice is responsible for a single sub-task or service, requiring fewer compute resources and lowering communication overhead. Based on the resource availability and workload of fog nodes, microservices can scale up and down dynamically due to loosely coupled modules. Compared to an existing monolithic design, integrating distributed microservices into the application process provide advantages such as independent deployment, scalability, and fault isolation [44].

As a user relocates from one place to another, the proximity to a fog or edge service may change; hence, user mobility restricts such benefits in practice [45]. IoT device mobility can impact fog computing systems when they repeatedly change access points. The mobility of end IoT devices causes migration of the requested application services from one computing node to another to maintain the desired

QoS. The deployment of local, small-scale data processing and storage at the network's edge using edge computing makes computations closer to the source data, thus ensuring the QoS requirements. In addition to meeting the demands of latency and bandwidth on the network, it offers intelligent services at the edge to fulfill the vital needs of IoT applications in real time [46, 47].

As described, resource augmentation plays a pivotal role in enhancing applications that rely on finite fog resources, with a specific emphasis on bolstering storage and computing capabilities. This augmentation becomes particularly critical for applications demanding efficient data processing and analysis. Notably, adherence to stringent Quality of Service benchmark necessitates innovative approaches, leading to the delegation of operational tasks from IoT nodes to neighboring nodes as proposed by Zhou *et al.* [48]. This strategy ensures optimal QoS while highlighting the need for a clustering methodology that supports resource augmentation within fog environments. Its distributed architecture not only adapts to changing resource demands but also promotes efficient resource utilization. The integration of clustering policies empowers individual nodes to strategically probe and subsequently register suitable cluster members, effectively aligning nodes with appropriate clusters. This meticulous process not only optimizes cluster formation but also enhances overall resource distribution, reflecting the significance of resource augmentation and dynamic collaboration in advancing the efficiency of fog-based IoT applications.

The integration of microservices, mobility, and clustering benefits in intelligent IoT edge healthcare applications stems from the need to develop agile, scalable, and responsive healthcare solutions that cater to the challenges posed by IoT and edge computing. The microservices architecture enables modular service development, ensuring efficient resource utilization and seamless updates. Mobility facilitates real-time data collection, remote patient monitoring, and telemedicine through mobile devices, expanding healthcare accessibility and improving remote diagnostics. Clustering enhances reliability, scalability, and workload distribution in edge environments, ensuring uninterrupted services and swift responses. This integrated approach addresses concerns related to latency, bandwidth, and privacy, culminating in dynamic, patient-centric healthcare services that harness the potential of IoT to offer personalized, real-time care directly at the edge.

This research primarily focuses on the implementation of edge/fog computing in time-critical applications, notably healthcare. The healthcare sector demands special attention due to its requirement for quality service delivery. For instance, emergency services like ambulances necessitate swift and secure data communication to maximize their efficacy. Even a minor communication delay can lead to substantial setbacks. While traditional healthcare relies on cloud computing for cost reduction, enhanced efficiency, and effective management of extensive data, it grapples with latency issues stemming from centralized processing. In contrast, edge/fog computing permits computations and minimal decisions to occur at edge/fog nodes, thereby optimizing efficiency. The integration of IoT and healthcare through edge/fog computing alleviates the burden on healthcare systems, offering individuals greater control over their well-being [14]. Consequently, IoT-integrated edge/fog computing, with its ability to incorporate processing into the network infrastructure, emerges as a viable solution to address healthcare requirements. In summary, processing vast volumes of IoT data using cloud computing seems to be time-consuming and inefficient due to factors like data variety, velocity, and latency; edge/fog computing excels in swiftly provisioning and processing data for real time applications due to its proximity to end devices [15]. In recent years, the convergence of the IoT and medical applications has brought about transformative changes in the healthcare landscape. This convergence allows for the seamless integration of connected medical devices and real-time patient monitoring.

Smart decision making holds exceptional significance within critical healthcare applications, particularly in the context of anomaly detection. In the intricate domain of healthcare, where timely and accurate responses are imperative, the ability to make intelligent decisions play a pivotal role in ensuring patient safety and well-being. Anomaly detection, which involves identifying unusual or potentially harmful patterns in medical data, necessitates swift and precise actions to mitigate risks. By employing advanced algorithms and data analytics, smart decision making systems can promptly recognize deviations from expected norms, enabling healthcare providers to respond proactively. This proactive approach not only aids in preventing adverse events but also enhances diagnostic accuracy and treatment efficacy. Moreover, the integration of smart decision making frameworks in anomaly detection minimizes false positives, thereby reducing unnecessary interventions and healthcare costs. Consequently, the adoption of intelligent decision making mechanisms in critical healthcare applications not only safeguards patient health but also optimizes resource utilization, augments clinical outcomes, and reinforces the overall reliability of healthcare systems.

Among various medical IoT applications, the detection of anomalies in ECG holds paramount significance in identifying and diagnosing cardiovascular irregularities, potentially preventing life-threatening conditions. However, the success of ECG anomaly detection hinges on efficient processing and analysis of extensive and sensitive medical data generated by distributed IoT devices. Traditional approaches to ECG anomaly detection typically involve centralizing data in cloud infrastructures for processing and analysis. While this centralized approach offers convenience, it raises considerable concerns about data privacy, security, and latency, especially in time critical medical applications. Moreover, adhering to stringent data regulations like the Health Insurance Portability and Accountability Act (HIPAA) presents significant challenges for healthcare providers and institutions. The integration of IoT devices, such as wearable ECG sensors, has enabled real time data collection and monitoring of patients' vital signs. However, the efficient processing and analysis of vast and sensitive medical data generated by these distributed IoT devices pose considerable challenges. These challenges include data privacy and security concerns, network bandwidth limitations, data heterogeneity, latency issues, and regulatory compliance. Given these complexities, federated learning has emerged as a promising solution to address these challenges by allowing multiple IoT devices to collaboratively train a global machine learning model while preserving the privacy and security of raw data in a decentralized manner.

Federated learning operates under the principle of decentralization, which is crucial in the context of medical IoT applications. Rather than sending sensitive medical data to a central server for analysis, federated learning enables training machine learning models directly on the devices where the data is generated. This approach offer enhanced data privacy and security since the raw data remains localized and is not exposed to a central entity. This aspect is particularly relevant in the medical field, where patient data confidentiality is of utmost importance. Federated learning thus mitigates concerns about data breaches and unauthorized access while enabling the development of accurate and valuable models [49].

However, federated learning is not without its challenges. One notable limitation is the communication overhead involved in coordinating the training process across multiple devices. This overhead can lead to increased latency and network congestion, potentially impacting the efficiency of the learning process. Additionally, federated learning faces vulnerabilities, such as Byzantine attacks, in situations where a trusted central authority is absent. These attacks involve malicious devices providing false or manipulated data to disrupt the learning process. Mitigating such attacks while maintaining the collaborative nature of federated learning is a complex task that requires careful consideration.

In conclusion, federated learning presents a promising approach to address the challenges posed by the integration of IoT devices in the medical field. Its ability to enable collaborative model training while preserving data privacy and security holds great potential. However, challenges related to communication overhead and security vulnerabilities must be carefully managed. The integration of blockchain technology further enhances the privacy, security, and efficiency of federated learning, paving the way for a robust and reliable framework for medical IoT applications. Blockchain offers a decentralized and tamper-proof ledger, ensuring transparent and immutable records of data transactions and model updates. By integrating blockchain with federated learning, a secure and auditable environment can be established for the collaborative training of machine learning models. Blockchain's decentralized nature and cryptographic principles provide additional protection against unauthorized access and tampering. This combination enhances the overall security and trustworthiness of the federated learning process, making it even more suitable for medical IoT applications.

Blockchain technology has emerged as a transformative force, revolutionizing data management by providing a decentralized and tamper-proof ledger. This innovative approach ensures that data records are transparent and auditable, allowing for secure and trustworthy transactions. When integrated with federated learning, blockchain's attributes synergize to create a powerful platform for training machine learning models in medical IoT applications. The immutability inherent to blockchain guarantees the integrity of data, preventing unauthorized alterations and ensuring a reliable historical record of transactions. This quality is especially critical in the context of medical data, where accuracy and accountability are paramount. The transparency offered by blockchain fosters a level of openness that enhances trust among stakeholders, assuring them that data has not been tampered or compromised. By incorporating these attributes, blockchain bolsters the security and reliability of federated learning, enabling the development of accurate and robust machine learning models [50].

The fusion of blockchain and federated learning creates a distributed and trustless environment that aligns seamlessly with the goals of medical IoT applications. With blockchain as the underlying foundation, the collaborative training of machine learning models become more secure and resilient. The decentralized nature of blockchain eliminates the need for a single central authority, distributing control and responsibility across a network of participants. This characteristic resonates well with the distributed nature of IoT devices, ensuring that no single point of failure exists. The environment's trustless nature arises from the blockchain's cryptographic principles, where transactions are validated through consensus mechanisms rather than relying on a central entity. This enhances security and reduces the risks associated with potential data breaches or unauthorized access. As a result, blockchain's integration empowers federated learning in medical IoT applications to operate in a highly secure, collaborative, and accountable manner, ultimately contributing to improved patient care and healthcare outcomes [51].

The convergence of blockchain technology and federated learning yields a synergistic approach that creates a distributed and trustless ecosystem exceptionally well-suited for the objectives of medical IoT applications. The foundational layer of blockchain introduces a paradigm shift in how collaborative machine learning model training is orchestrated, infusing heightened security and resilience into the process. By design, blockchain's decentralized architecture obviates the requirement for a centralized authority, effectively dispersing control and accountability among a network of participants. This inherent decentralization aligns harmoniously with the intricate web of interconnected IoT devices, effectively nullifying the vulnerabilities associated with a singular point of failure.

Transactions within this framework are validated through consensus mechanisms, eradicating the dependence on a central entity for verification. This cryptographic validation augments security and mitigates the possibility of data breaches and unauthorized access. As a result, bringing together the features of blockchain enhances the working environment of federated learning in medical IoT situations. This combination provides strong security, exceptional collaboration, and unquestionable accountability. This combined approach takes on a transformative role, with the potential to significantly change patient care and healthcare results. It does so by guaranteeing the confidentiality, integrity, and effective collaboration of medical data and insights.

1.2 Motivation of Research

Developing IoT smart healthcare applications for ECG anomaly detection through edge/fog and cloud computing while incorporating the mobility of end nodes is driven by several compelling factors. The need for timely and accurate ECG anomaly detection is critical for early diagnosis and intervention in cardiac conditions [52]. By harnessing IoT, the application can continuously monitor patients' ECG data remotely, enabling rapid detection of abnormalities and reducing the risk of life-threatening events. Integrating edge/fog and cloud computing further enhances real time processing at the network's edge, minimizing latency and ensuring immediate responses, which is crucial for time-sensitive medical scenarios.

The incorporation of mobility in end nodes amplify the potential impact of the application [53]. Patients can be equipped with wearable devices that collect ECG data as they go about their daily lives, allowing continuous monitoring without confining them to a specific location. This freedom of movement facilitates more accurate assessments of cardiac abnormalities under various real world conditions and activities. Additionally, mobile devices enable seamless transmission of data to edge nodes for analysis, ensuring that healthcare providers receive up-to-date information regardless of the patient's physical location.

The convergence of edge/fog computing offers distinct advantages for ECG anomaly detection. Edge/Fog computing distributes computing tasks between edge devices and centralized servers, optimizing data processing and reducing the load on the central infrastructure [54]. This approach not only enhances real time processing but also minimizes data transmission to the cloud, addressing bandwidth constraints and privacy concerns. By fusing mobility with this architecture, patients can experience continuous cardiac monitoring while benefiting from localized, rapid analytics, ultimately leading to improved patient outcomes, enhanced diagnostic accuracy, and a more accessible and flexible healthcare solution.

The integration of blockchain based federated learning addresses data privacy concerns and empowers collaborative healthcare. In the context of ECG data, privacy is paramount. Blockchain's decentralized and secure nature ensures that patient data remains encrypted and is accessible only to authorized entities. The federated learning approach enables model training across distributed edge devices without the need to centralize sensitive data. This collaborative learning benefits from a diverse range of data while maintaining privacy, which is particularly crucial for healthcare applications involving personal medical data [55].

The mobility of end nodes add a dimension of versatility and patient-centric care. Incorporating wearable devices or mobile sensors allow individuals to carry on with their routines while under continuous ECG monitoring [56]. Mobility captures a broader spectrum of cardiac activities, facilitating more accurate diagnosis and personalized treatment plans. These mobile devices can also contribute to the federated learning process, enhancing the model's accuracy with real world data from diverse locations and contexts.

Furthermore, the synergy of edge/fog computing amplifies efficiency and responsiveness. Edge/Fog computing's proximity to data sources reduces latency and enhances real time analysis, while edge computing optimizes resource usage. This combination ensures that critical ECG data is processed swiftly, allowing for instant feedback and potentially life-saving interventions. The federated learning methodology enriches this ecosystem by enabling iterative model updates across distributed nodes, ensuring continuous improvement without compromising data security.

In conclusion, the motivation behind this advanced IoT smart healthcare application is the need to revolutionize cardiac care through a comprehensive approach. By leveraging edge/fog computing, blockchain based federated learning, and the mobility of end nodes, this application addresses the intricacies of cardiac health monitoring, privacy concerns, collaborative learning, and real time responsiveness. Ultimately, the fusion of these technologies stand to enhance patient outcomes, empower healthcare providers, and pave the way for a new era of patient-centric, data-secure, and technologically-driven healthcare solutions.

1.3 Scope of Research

The scope for developing an IoT smart healthcare application for ECG anomaly detection using edge/fog and cloud computing, along with blockchain based federated learning methods and incorporating mobility of end nodes, is substantial and offer transformative possibilities in the field of healthcare. Such an application holds the potential to revolutionize remote patient monitoring. By enabling continuous ECG monitoring through wearable devices or mobile sensors, patients can experience personalized and proactive cardiac care, leading to early anomaly detection and intervention. This real time monitoring not only improves patient outcomes but also reduces the burden on healthcare facilities by mitigating the need for frequent in-person visits.

The integration of edge/fog computing ensures data processing at the network's edge, minimizing latency and facilitating immediate responses. This is crucial for cardiac care, where time-sensitive anomalies demand quick reactions. The combination of mobility and edge computing enables seamless data collection, analysis, and transmission, allowing healthcare providers to make informed decisions in real time. Additionally, the decentralized and secure nature of blockchain based federated learning ensures patient data privacy while harnessing the collective intelligence of distributed edge devices for model improvement.

The application's scope extends to enhancing medical research and innovation. The federated learning approach fosters collaboration among institutions, enabling the aggregation of diverse ECG datasets without compromising individual data privacy. This accumulated knowledge can lead to the development of more accurate and robust anomaly detection models. Moreover, the mobility of end nodes contribute to real world simulations to the models, making them adaptable to different patient lifestyles and environmental factors.

Furthermore, the scope encompasses addressing healthcare disparities and accessibility challenges. The application's architecture empowers remote and underserved populations with the ability to access high-quality cardiac care regardless of geographical location. This inclusivity can lead to earlier detection and management of cardiac conditions, ultimately reducing healthcare inequalities.

In conclusion, the scope for developing an IoT smart healthcare application for ECG anomaly detection is expansive and multidimensional. By combining edge/fog computing, blockchain based federated learning, and mobility of end nodes, this application has the potential to redefine patient monitoring, enhance medical research, and bridge healthcare gaps. It represents a powerful convergence of technological advancements and healthcare needs, ushering in a new era of personalized, responsive, and privacy-centric cardiac care.

1.4 Research objectives and contributions

1.4.1 Thesis Goals

- Propose an advanced resource provisioning solution utilizing IoT microservices combined with mobility management for healthcare applications.
- Implement a multiobjective optimization framework using the weighted sum method to fine-tune critical parameters of the application.
- Employ modified metaheuristic scheduling techniques to enhance resource provisioning efficiency in fog and edge devices.
- Design an early warning system for ECG anomalies leveraging a Smart Decision Making module utilizing blockchain-based federated learning methods.
- Create a robust and efficient healthcare application framework that can effectively manage resources while ensuring high performance and reliability.

1.4.2 Contributions

- Introduces an innovative resource provisioning solution that integrates IoT microservices with mobility management, addressing the dynamic needs of healthcare applications.
- Develops a multiobjective optimization approach, utilizing the weighted sum method to optimize key application parameters, thus enhancing overall system performance.
- Employs modified metaheuristic scheduling techniques, which improve the efficiency of resource allocation in fog and edge computing environments.
- Designs and implements an early warning system for ECG anomalies, which enhances patient monitoring and timely intervention with enhanced privacy and security.

1.4.3 Innovative Aspects

- The use of Blockchain-based Federated Learning introduces a privacy-preserving method that safeguards end-user data, a critical aspect in healthcare.
- Identification and analysis of the most suitable placement policy for deploying the Blockchain-based Federated Learning module within edge, fog, and cloud layers demonstrate a pioneering approach to enhancing data security and processing efficiency.

- The effectiveness of the proposed solution is validated through extensive simulations under real workloads, measuring parameters such as energy consumption, network usage, cost, execution time, and latency.
- Demonstrates the practical utility and advantages through simulations executed within a controlled experimental framework.

1.4.4 Research questions addressed in the Thesis

- How can IoT microservices be effectively utilized for resource provisioning and mobility management in healthcare applications?
- What are the key parameters to be optimized in healthcare applications, and how can multiobjective optimization using the weighted sum method improve these parameters?
- How can modified metaheuristic scheduling techniques enhance resource provisioning efficiency in fog and edge devices for healthcare applications?
- What design considerations are essential for developing an early warning system for ECG anomalies using a Smart Decision Making module?
- How can Blockchain-based Federated learning be integrated into critical healthcare applications to ensure privacy-preserving methods for end-user data protection?
- What is the most suitable placement policy for deploying the Blockchain-based Federated learning module within the Edge, Fog, and Cloud layers of the architecture?
- How does the proposed resource provisioning solution perform under real workloads in terms of energy consumption, network use, cost, execution time, and latency?
- What challenges arise when implementing IoT microservices with mobility management in healthcare applications, and how can they be mitigated?
- How can the effectiveness of the proposed resource provisioning solution be validated through simulation experiments, and what metrics should be used?
- What impact does the integration of Blockchain-based Federated learning have on the overall performance and security of healthcare applications in a distributed environment?

1.5 Organization of the Thesis

This thesis is organized into seven chapters.

Chapter 1 discusses the introduction and the motivation of the research. In Chapter 2 of this thesis, a comprehensive examination of the literature on edge and fog computing for IoT applications is presented. The survey is organized into six distinct sections, covering a wide spectrum of topics. The initial sections deeply examine the expansion of IoT applications beyond conventional domains and their transformative role in healthcare. The subsequent sections critically evaluate resource management strategies in edge and fog environments, explore the integration of microservices and mobility support in IoT scenarios, and analyze the simulation tools tailored for assessing the behavior and scalability of IoT edge and fog setups. The summarized literature survey is presented, which fostered identifying research gaps and helped define the objectives of this research work.

In Chapter 3, a method called Multiobjective Optimization for IoT applications is introduced. The chapter explains how the proposed system works with its different parts. It also talks about the datasets used in the experiments. The results of the experiments are carefully studied, and important observations are shared. This exploration helps us understand the new Multiobjective Optimization concept and how it can be useful in real world situations, using actual results and important insights. It also discusses the drawbacks and the need for metaheuristic methods for resource management.

Chapter 4 of the document introduces the concept of Metaheuristic methods applied to IoT applications. The chapter provides an explanation of the functioning of the proposed system, including its various components. It also addresses the datasets employed in the experimental phase, and the results derived from these experiments are subjected to detailed analysis, with significant findings being shared. This exploration aids in comprehending the Metaheuristic Optimization concept and its practical relevance, drawing on tangible outcomes and crucial insights. Additionally, the chapter discusses the requirement for federated learning techniques in resource management for intelligent healthcare applications.

Chapter 5 of this technical document sheds light on the critical significance of incorporating federated learning within the realm of IoT applications. The chapter intricately unveils the operational mechanics of their proposed system, employing autoencoders as a key component, and provides a comprehensive overview of its constituent elements. In a diligent manner, the chapter scrutinizes the data employed in their empirical assessments, meticulously examining the outcomes derived from these trials, unearthing pivotal insights. Through this rigorous investigation, a coherent comprehension of the functioning of federated learning emerges, accentuating its practical relevance through outcomes and concepts. Additionally, the discussion extends to the imperative role of blockchain methodologies in addressing resource management necessities for intelligent healthcare applications.

Chapter 6 of this report focuses on how vital it is to use blockchain in IoT applications. The chapter explains in detail how their proposed system works, using Ethereum blockchain as an important part. They also carefully study the information they used in their tests, looking closely at the results from these tests to find important conclusions. This deep investigation helps us understand how federated learning works better, showing its importance through clear results and important ideas. Chapter 7 concludes the findings and gives insight into the future work.

Chapter 2

Literature Survey

2.1 Introduction

According to the National Institute of Standards and Technology (NIST), cloud computing as a model facilitates ubiquitous, useful, and on-demand network access to a larger pool of fully programmable, distributed computing resources which are rapidly provisioned and de-provisioned with minimal interaction and lesser implementable complexity [57]. The cloud is the only technology that can analyze, store, and access the IoT, depending on the deployment model. In recent years, IoT technology has gained significant interest for embedded applications [58]. IoT is a technological innovation capable of changing applications in various fields and achieving effective results [59]. IoT devices have limited memory and processing capacities that lead to problems with performance, reliability, and security. Thus, integrating IoT with the cloud with huge storage and processing capacity will lead to better performance of real-time systems [60]. The emergence of IoT has transformed many applications that include applications in manufacturing, gas and oil plantation, utilities, transportation, public safety, local governance, and health care [61]. IoT technology has gained significant interest in healthcare applications because of its capability to handle the issues in healthcare systems due to the increase in the aging population and chronic diseases. Considering the extensive use of cloud computing, certain IoT applications and healthcare services seem unable to benefit from this popular computing technique due to inherent cloud computing challenges such as latency, location awareness, and flexibility. As a result, edge/fog computing has emerged as a promising technology at the edge of the network to provide elastic services [25]. Edge and fog computing collectively enhance distributed computing by bringing processing closer to data sources, improving latency and real-time capabilities, while the key distinction lies in their scope—edge computing typically involves local devices, while fog computing extends its reach to cover a broader, intermediate layer of the network infrastructure. Edge/Fog computing techniques include connecting things to analyze and respond to big data they produce in a fraction of a few seconds and sending only the required data alone to the cloud for big-data analytics and storage [61]. Latency reduction is the main advantage of edge/fog computing, and hence, it can be used in IoT healthcare applications as they expect the system to be latency-sensitive. Such applications may be provisioned with the help of the edge/fog computing paradigm along with cloud technology [62]. The subsequent paragraphs elaborate on the current status of resource management for IoT applications across various domains, including healthcare, mobility, and microservice implementations within the same domain. It also addresses the existing simulation tools, provides a summary of the literature review, and outlines the research objectives.

In the forthcoming sections, Section 2.2 will explore the landscape of IoT applications within edge, fog, and cloud computing. Following this, Section 2.3 will delve into the forefront of research regarding the transformative impact of IoT on healthcare within these computing environments. Subsequently, in Section 2.4, the focus will shift to investigating efficient resource management within edge, fog, and cloud systems. Section 2.5 will introduce an overview of prevailing metaheuristic methodologies for resource management in IoT applications. Section 2.6 will survey the realm of federated learning, a collaborative learning approach, and its integration within these computing paradigms. In Section 2.7, the examination will extend to the utilization of blockchain technology alongside federated learning for enhanced data security. Additionally, Section 2.8 will analyze the current discourse on the implications of mobility on IoT deployments in edge, fog, and cloud environments. Section 2.9 will spotlight the exploration of microservices' role in enhancing the flexibility and scalability of IoT systems. Finally, Section 2.10 will review simulation tools as indispensable aids for modeling and understanding intricate IoT systems.

2.2 IoT applications in Edge/Fog/Cloud computing

Research in the fog computing paradigm is a growing field for critical real-time applications, with many unresolved problems. Deepika *et al.* suggests Exigency Alert Mobile Cloud (EAMC), a smartphone-based service that provides an easy way to alert the various emergency services like accidents, fire, building collapse, murder, robbery, terrorism, and health using fog technology for preprocessing and Reliable Routing Protocol for workload offloading. The EAMC-installed smartphone has been used as an end node, and private cloud XenServer is used for fog communication. The implementation has been tested for fog and non-fog scenarios and proves that the delay suffered by non-fog scenario is up by nine times compared to the fog scenario [63].

Harshit *et al.* carries out a case study on iFogSim simulation, a latency-sensitive online game and intelligent surveillance distributed camera networks [64]. An EEG sensor provides EEG signals to the online game application that is sensitive to latency and a DISPLAY actuator shows the user the current game scene. Application Model of EEG Game is depicted in Figure 2.1. The cloud-only placement (cloud computing only) and edgeward placement (fog computing and cloud computing) efficiencies were assessed by taking into account parameters such as latency, network use, and energy consumption.

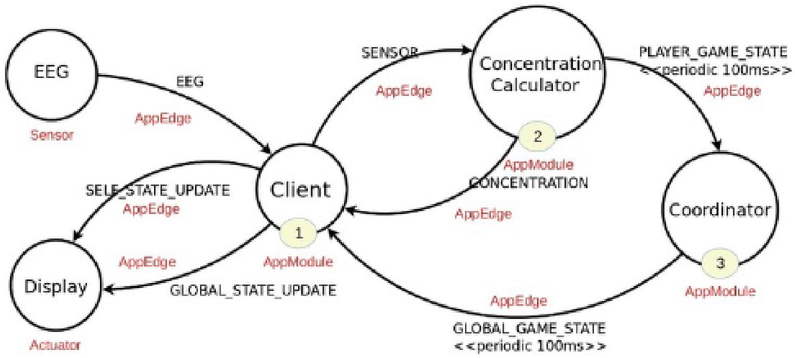


Figure 2.1: Application Model of EEG Game [64]

Moumita *et al.* suggests an intelligent K* heuristic search algorithm to detect anomalies in the data (also called emergency situation) and to determine the shortest path to the victim area for a mission-critical application [65]. The simulation results show that, relative to the cloud-based network, the suggested fog scenario lowers power consumption, average jitter, and average latency by 12–15%, 10–14% and 9–11%, respectively. The simulation findings suggest that saving about 20 percent of resources increase the performance of the system. A real-time cloud-fog-edge IoT collaboration platform, Mobi-IoST, proposed by Shreya *et al.* [66], uses a smart decision-making approach to predict and deliver the processed data to the user interface. But the proposed model suffer from computation offloading issues due to seamless connectivity. The Roadside units (RSU) forward the result to the cloud if the user changes position or gets disconnected. But during data and computation offloading in the wireless network, one of the main challenge faced is seamless connectivity. The preceding studies are summarized in Table 2.1.

Table 2.1: IoT Applications in Edge/Fog/Cloud Computing

Reference	**Application**	**Parameters**
Deepika *et al.* [63]	Emergency Alerts	Delay
Gupta *et al.* [64]	Latency-Sensitive Apps	Latency, Energy
Mishra *et al.* [65]	Anomaly Detection	Power, Jitter, Latency
Ghosh *et al.* [66]	Real-time IoT Collaboration	Computation Offloading

The use of fog computing finds its place in the above-said time-critical applications by improving the QoS and reducing the latency. Among many applications, the healthcare sector deserves to be given priority in terms of the quality of services compared to other industries. The following paragraph discusses the existing technology in the medical sector.

2.3 Healthcare IoT in Edge/Fog/Cloud IoT applications

Among various applications, the healthcare sector stands out as a priority for delivering superior service quality compared to other sectors. IoT-based applications enable essential functions like simultaneous reporting and monitoring, tracking, alerts, and remote medical assistance. According to research conducted by the Center for Connected Health Policy, remote health monitoring systems have shown a reduction in the re-admission rates of heart failure patients by 50 percent [67, 68]. The following paragraph discusses the use of edge/fog/cloud technology in the medical sector.

Heba Nashatt *et al.* introduce an E-health and wellness monitoring application designed to promote a healthier lifestyle. This research gathers and analyzes user behaviors to make predictions and offer personalized recommendations [69]. However, there is an issue with data processing delay, particularly in critical emergency situations. The comparison between multidimensional QoE (MD-QoE) and the QoE aware policy is illustrated in Table 2.2. Adesh Kumar *et al.* propose a real-world cloud-based smart medical system that utilizes communication networking, allowing doctors to provide online treatment to their patients. This application employs mobile devices and wireless body area networks, potentially extending to fog technology. The proposal asserts that this framework is more efficient in computation and communication costs compared to existing protocols in smart healthcare [70].

Table 2.2: Performance Comparison of MD-QoE and QoE-Aware Policy [69]

Metric	**MD-QoE**
Application Placement Time	13.24% more Than QoE
Application Delay	13.28% less than QoE
Network Usage	13.09% less than QoE
End-User Power Consumption (Online Gaming)	21% less than QoE
End-User Power Consumption (E-Healthcare)	24% less than QoE
Control Node Power Consumption	8.04% more Than QoE

Fatema *et al.* [71] has been conducting experiments on brain strokes as time-sensitive data and illustrated the advantages of fog compared to cloud by comparing the parameters like execution time, energy consumption, costs, and network use. The findings of simulation studies show that fog computing performs better than cloud computing when it comes to time-sensitive tasks.

Redowan *et al.* [33] suggest an IoT healthcare application with the integration of cloud-fog services as IoT healthcare time-critical data can be efficiently processed and managed effectively by fog resources. The proposed application model supports modular development and leverages inter-module data dependencies for distributed deployment in constrained fog settings, enabling the customization of cloud-based IoT-Healthcare applications for fog environments. The resources are virtualized and shared in fog nodes as microcomputing instances (MCI). The possibilities are analyzed using the iFogSim simulator, and the performance has been evaluated. The parameters considered are latency, power, data communication and distributed computing factors. The findings show a nominal reduction in cost, energy consumption and network latency. Though it satisfies the single MCI requirement for small-scale healthcare applications, the proposed model cannot accommodate additional MCIs in a single node for large-scale applications. Additionally, the simulation demonstrates service distribution in cloud-fog integration, accounting for varying numbers of sensors and CPU utilization under limited computational resources in the fog environment. Figure 2.2 illustrates the proposed model for the aforementioned paper.

Hindia *et al.* introduce a comprehensive two-stage approach for the deployment of IoT sensor networks, capitalizing on the expanding global acceptance of IoT technologies. The first stage involves sensors efficiently collecting particle measurements through an android application. Subsequently, in the second stage, the gathered data is transmitted over a Femto-Long-Term Evolution (LTE) network, employing a novel scheduling technique. This scheduling strategy, tailored to the application's priorities, distinguishes the proposed approach. The authors validate the efficiency of their technique by comparing it with established algorithms, namely proportional fairness and exponential proportional fairness (PF/EXP), showcasing its efficacy in the realm of healthcare monitoring systems [72]. The performance evaluation is illustrated in Table 2.3 as outlined in the paper.

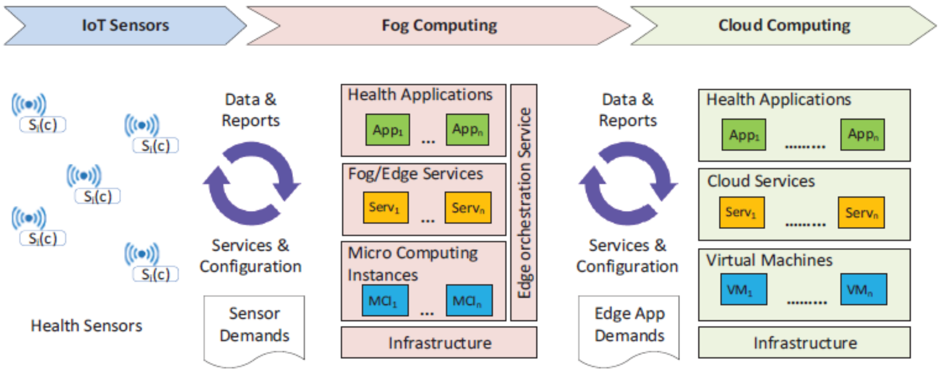


Figure 2.2: Architecture of Cloud-Fog Integration for Interoperable IoT-Healthcare Solutions [33]

Table 2.3: Summary of Performance Metrics [72]

Metric	Proposed Approach	EXP/PF Approach
Throughput	High	Moderate
Delay	Robust	Exponential Growth
Packet Loss Ratio (PLR)	Low	Moderate

George *et al.* [73] recommends various techniques for three categories of patients, namely critically injured, hospitalized, and monitoring patients. The proposed model has been analyzed for its sensitivity to the parameters like latency and real-time interaction but fails to consider the privacy, bandwidth optimization, energy, and scalability. Gill *et al.* [32] proposes an information model which is fog assisted IoT health care cloud service. Simulation is done in iFogSim using heart patients' data. Results claim 22.61-26.78% reduction in network use time, 19.56-29.45% reduction in latency, 23.56% reduction in energy consumption when using fog compared to the cloud. But this model only considers latency, network usage and energy consumption for comparison and fails to compare parameters like cost, bandwidth etc. A comprehensive survey on enabling technologies for fog computing in healthcare IoT systems has been conducted by Ammar *et al.* [74]. The survey identifies the unexplored research domains like computation offloading, load balancing/distribution, and interoperability of IoT healthcare applications. The preceding studies are summarized in Table 2.4.

Table 2.4: Literature Survey on IoT-based Healthcare Applications

Reference	Objective	Parameters Considered	Key Findings
Brahmbhatt *et al.* [67]	Remote health monitoring	Re-admission rates reduction for heart failure patients	50% reduction in re-admission rates
Nashaat *et al.* [69]	E-health and wellness monitoring	User behavior analysis, predictions, personalized recommendations	Data processing delay in critical emergency situations
Porkodi *et al.* [70]	Cloud-based smart medical system	Computation and communication costs, online treatment	More efficient than existing protocols in smart healthcare
Zohora *et al.* [71]	Brain strokes monitoring	Execution time, energy consumption, costs, network use	Fog computing outperforms cloud computing for time-sensitive tasks
Mahmud *et al.* [33]	IoT healthcare application	Latency, power, data communication, distributed computing	Nominal reduction in cost, energy consumption, network latency
George *et al.* [73]	Various techniques for patient categories	Sensitivity to latency, real-time interaction	Lack of consideration for privacy, bandwidth optimization, energy, scalability
Gill *et al.* [32]	Fog-assisted IoT healthcare cloud service	Network use time, latency, energy consumption	Significant reduction in network use time, latency, and energy consumption
Mutlag *et al.* [74]	Survey on enabling technologies	Computation offloading, load balancing/distribution, interoperability	Identifies unexplored research domains

2.4 Resource allocation in Edge/Fog/Cloud IoT applications

The allocation of processing power, network, and storage resources for IoT applications is referred to as provisioning. In fog or edge computing, it is essential for these nodes to supply resources to accommodate IoT requests efficiently. Resource provisioning has been a subject of study for a considerable time and has garnered increased attention in recent literature [75]. This section examines the issues related to resource management in fog and edge computing.

Guneeth *et al.* discuss the challenges arising from the proliferation of IoT sensors and smart devices across diverse domains, necessitating increased computational capabilities from cloud to the edge. The paper comprehensively reviews the complexities of resource management in fog/edge computing, addressing issues such as heterogeneous resources, transactional workloads, edge node discovery, and Quality of Service parameters. The authors emphasize the adoption of AI-based techniques to tackle these challenges and shed light on promising research directions, advocating for the integration of cutting-edge technologies to enhance business intelligence in IoT-based applications [76]. Hong *et al.* investigate the shift from centralized cloud data centers to the decentralized paradigm of fog/edge computing, emphasizing the utilization of resources closer to user devices. Their extensive review, spanning publications from 1991 to 2018, categorizes architectures and algorithms to tackle the resource management challenges inherent in the dynamic and heterogeneous nature of fog/edge resources [77].

Ismael *et al.* underscore the paramount importance of resource management in fog computing, emphasizing its pivotal role in mitigating network congestion and ensuring low-latency services at the network edge. The survey systematically addresses the phases of fog computing implementation, accentuating the critical need for efficient resource provisioning, allocation, and management to support large-scale IoT applications and enhance overall system reliability [78]. Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment framework incorporates a dynamic resource provisioning approach, leveraging the benefits of SDN and employs the nondominated sorting genetic algorithm-III to optimize energy consumption and completion time, thereby attaining well-balanced scheduling strategies [79].

The article introduces a unique resource representation scheme where Edge Devices (EDs) communicate their resource data to an edge node supervisor using standardized mobile Edge Computing (EC) APIs. Resource allocation involves sharing ED resource information with the supervisor, aided by a dynamic resource allocation framework. The scheme's efficacy is confirmed through theoretical and experimental simulations, showcasing its superiority over benchmark approaches across diverse system parameters [80].

Hu *et al.* introduce CEC, a containerized edge computing framework designed for dynamic resource provisioning, particularly in smart connected communities with multiple intelligent applications; CEC incorporates workload prediction and resource pre-provisioning to ensure minimal latency for user service requests and optimal utilization of edge resources [81].

In the context of a cloud-assisted mobile edge framework, Ma *et al.* cast resource provisioning as an optimization challenge and leverage the problem's piecewise convex nature to introduce diverse instances of the Optimal Resource Provisioning (ORP) algorithms. These algorithms aim to enhance edge host computation capacity optimization while dynamically adapting the cloud tenancy strategy [82]. Table 2.5 consolidates the summary of resource management studied in edge/fog computing.

Table 2.5: Summary of Resource Management Studies in Edge/Fog Computing

Reference	**Application Area**	**Findings**
Mahmud *et al.* [75]	IoT Resource Provisioning	Resource provisioning for efficient IoT
Walia *et al.* [76]	IoT-based Applications	Addressing resource management challenges and proposing AI-based solutions
Hong *et al.* [77]	IoT Resource Management	Categorization of resource management complexities
Martinez *et al.* [78]	Fog Computing	Resource management in mitigating network congestion
Xu *et al.* [79]	Workflow Scheduling in Edge Computing	Dynamic resource provisioning to optimize energy consumption and completion time
Amine *et al.* [80]	Edge Computing	Efficient edge device communication and allocation
Hu *et al.* [81]	Containerized Edge Computing	Containerized framework for optimal resource provisioning
Ma *et al.* [82]	Cloud-assisted Mobile Edge	Optimal resource provisioning

2.5 Metaheuristic methods in Edge/Fog/Cloud IoT applications

Metaheuristic methods have emerged as indispensable tools for optimizing resource management, task scheduling, and data processing in the realm of edge, fog, and cloud IoT applications. With the proliferation of IoT devices generating vast amounts of data at the network edge and the need for efficient utilization of computational resources in fog and cloud environments, traditional optimization techniques often fall short in addressing the dynamic and complex nature of these systems. Metaheuristic methods offer a promising approach by leveraging iterative search algorithms inspired by natural phenomena or computational paradigms to efficiently explore large solution spaces and find near-optimal solutions [83]. In this literature survey, the application of metaheuristic methods in edge, fog, and cloud IoT scenarios is clarifying their role in enhancing resource allocation, task scheduling, energy optimization, fault tolerance, security, and privacy. Through an examination of recent research and developments, insights into state-of-the-art techniques and emerging trends in this rapidly evolving field are provided.

Shakarami *et al.* propose an overview of resource provisioning methods in fog computing environments and discuss the open challenges in this area. Machine learning-based, heuristic/meta-heuristic-based, framework-based, game theoretic-based, and model-based are the five primary classifications presented [84]. Masoumeh *et al.* provide a resource provisioning technique that uses a Bayesian learning-based autonomic computing model for decision-making and control loop planning. This work has been carried out using time series prediction models [85]. The work proposed in [86] also uses Bayesian learning along with linear regression and autonomic computing to efficiently allocate the cloud resources. Dinesh *et al.* suggest an improved resource provisioning method based on the JAYA (a sanskrit word meaning victory) approach for placing virtual machines in a data center which aims to reduce energy consumption by effectively organizing the migrated VMs [87]. Literature also presents many heuristic-based and evolutionary-based techniques for task scheduling. Heuristic algorithms are faster than evolutionary algorithms but unsuitable for finding an optimal solution in NP-complete situations. Recently, metaheuristics have also been employed to generate optimal solutions [88].

Mishra et al. use metaheuristic service allocation algorithms for a heterogeneous fog computing system that processes heterogeneous jobs, formulate the linear programming problem for time and energy optimization and uses particle swarm optimization (PSO), binary PSO, and bat algorithm [89]. Hosseinioun *et al.* propose a strategy based on the dynamic voltage and frequency scaling (DVFS) technique that is energy aware and saves it using hybrid invasive weed optimization [90]. Ashkan *et al.* recommend FOGPLAN, a QoS-aware dynamic fog service provisioning framework, by defining it as an optimization problem and evaluating it using a simulation based on real-world traffic traces [91]. Naranjo *et al.* propose a penalty-aware bin packing heuristic algorithm for resource management hosted by each fog node, allowing resource consolidation and admission control by scaling up or scaling down computation frequencies [92]. To reduce the average peak age of information, Fang *et al.* designed the associated time slot allocation problems and, using an exact linear search strategy, found the best solutions to the resulting non-convex problems [93].

FCM-FPA, a new fuzzy clustering with flower pollination method as a resource provisioning model for fog computing proposed in the literature, includes resource normalization and fuzzy clustering and has been evaluated using the Iris and Wine datasets [42]. Abdel *et al.* introduce an energy-aware meta-heuristic approach for task scheduling based on Harris hawk optimization to enhance QoS, which also assesses energy consumption, cost, makespan, flowtime, and carbon dioxide emission [83]. To more effectively address numerous research difficulties, such as resource placement and scheduling, mobility, communication and edge control, many nature-inspired metaheuristic (NIMH) methods have been applied in edge computing. Fuzzy logic, edge network systems, and various research issues are all included in the survey conducted by Adhikari *et al.*, which divides the current NIMH into three categories based on the nature of their work [94]. To reduce Service Level Agreement (SLA) violations caused by the limitations of edge computing resources and to handle the computational complexity of edge computing problems, Adyson *et al.* propose a random and heuristic approach to initialize the population for multi-objective genetic algorithm. The solution thus developed is found to be close to optimal and is employed to examine the placement and load distribution of IoT applications. It performs better than existing benchmark algorithms in response to deadline violation, cost, and service accessibility [95]. Fang *et al.* provide a heuristic PSO approach built on a Lyapunov framework to balance system queue backlog and energy efficiency for trajectory scheduling and allocation of computational resources for Internet of Underwater Things [96].

Mishra et al. use metaheuristic service allocation algorithms for a heterogeneous fog computing system that processes heterogeneous jobs, formulate the linear programming problem for time and energy optimization and uses particle swarm optimization, binary PSO, and bat algorithm [89]. Hosseinioun *et al.* propose a strategy based on the dynamic voltage and frequency scaling technique that is energy aware and saves it using hybrid invasive weed optimization [90]. Ashkan *et al.* recommend FOGPLAN, a QoS-aware dynamic fog service provisioning framework, by defining it as an optimization problem and evaluating it using a simulation based on real-world traffic traces [91]. Naranjo *et al.* propose a penalty-aware bin packing heuristic algorithm for resource management hosted by each fog node, allowing resource consolidation and admission control by scaling up or scaling down computation frequencies [92]. To reduce the average peak age of information, Fang *et al.* designed the associated time slot allocation problems and, using an exact linear search strategy, found the best solutions to the resulting non-convex problems [93].

FCM-FPA, a new fuzzy clustering with flower pollination method as a resource provisioning model for fog computing proposed in the literature, include resource normalization and fuzzy clustering and has been evaluated using the Iris and Wine datasets [42]. Abdel *et al.* introduce an energy-aware meta-heuristic approach for task scheduling based on Harris hawk optimization to enhance QoS, which also assesses energy consumption, cost, makespan, flowtime, and carbon dioxide emission [83]. To more effectively address numerous research difficulties, such as resource placement and scheduling, mobility, communication and edge control, many nature inspired metaheuristic methods have been applied in edge computing. Fuzzy logic, edge network systems, and various research issues are all included in the survey conducted by Adhikari *et al.*, which divides the current NIMH into three categories based on the nature of their work [94]. To reduce SLA violations caused by the limitations of edge computing resources and to handle the computational complexity of edge computing problems, Adyson *et al.* propose a random and heuristic approach to initialize the population for multiobjective genetic algorithm. The solution thus developed is found to be close to optimal and is employed to examine the placement and load distribution of IoT applications. It performs better than existing benchmark algorithms in response to deadline violation, cost, and service accessibility [95]. Fang *et al.* provide a heuristic particle swarm optimization approach built on a Lyapunov framework to balance system queue backlog with energy efficiency for trajectory scheduling and allocation of computational resources for Internet of Underwater Things [96]. Table 2.6 provides a summary of the aforementioned related works in heuristic/metaheuristic methods, outlining their objectives and the employed techniques. Table 2.7 presents an overview of the aforementioned studies focusing on heuristic and metaheuristic methodologies, specifying the parameters they discussed.

Table 2.6: Summary of related works in heuristic/metaheuristic methods

Reference	**Purpose**	**Technique**
Mishra *et al.* [89]	Service allocation	PSO, BPSO, BAT
Hosseinioun *et al.* [90]	Task scheduling	IWO-CA
Ashkan *et al.* [91]	Resource provisioning	Greedy algorithm
Naranjo *et al.* [92]	Resource management	Bin packing based
Porkodi *et al.* [42]	Resource provisioning	Optimal FPA

Table 2.7: Overview of Heuristic/Metaheuristic Studies: Discussed Parameters

Reference	**Energy Consumption**	**Execution Time**	**QoS/QoE**	**Cost**	**Delay**	**Network Use**
Mishra *et al.* [89]	✔	✔				
Hosseinioun *et al.* [90]	✔					
Ashkan *et al.* [91]			✔	✔		
Naranjo *et al.* [92]	✔				✔	
Porkodi *et al.* [42]			✔			
Proposed	✔	✔		✔	✔	✔

As discussed in the previous chapter, among many applications, the healthcare sector deserves to be prioritized in terms of service quality compared to other domains. Critical functions such as simultaneous reporting and monitoring, tracking and alerts, and remote medical aid are all possible with IoT-based apps. The center for connected health policy conducted a study that observed that remote health monitoring systems lower the re-admission rates of heart failure patients by 50 percent [67, 68]. The following paragraph discusses fog and edge technology in the medical sector.

Nashatt *et al.* propose an E-health and wellness monitor application to encourage a healthier lifestyle. This work gathers the user behaviors, analyzes them, and later predicts certain events with personalized recommendations [97]. However, the latency in data processing affects critical emergencies. Adesh *et al.* suggest a real-life cloud-based smart medical system using a communication networking where-in a doctor treats his patients via internet. This proposed application uses mobile, wireless body area networks, and so on, intending to be extended to fog technology. The proposal claims that the suggested framework is more effective in computation and communication expenditure than the existing protocols in smart healthcare [98]. Tuli *et al.* propose a lightweight healthcare fog service that manages cardiac patients' IoT data. The FogBus framework allows efficient edge/fog/cloud integration for reliable and fast results [99]. A comprehensive overview of fog-based technologies in healthcare IoT systems has been conducted by Ammar *et al.* [74]. To identify the challenges and requirements of edge devices for diverse use cases, Morghan *et al.* explore current and developing edge computing architectures and approaches for health care applications [100]. To evenly distribute the load amongst fog nodes when the health monitoring system is installed on a big scale, Asghar *et al.* offer a new load balancing scheme (LBS). It presents the comparison of the parameters, network use, and latency for different placements namely cloud-only implementation, load balancing scheme, and fog node placement [101].

Jayasena *et al.* use a whale optimization based meta-heuristic algorithm for optimal task scheduling in a smart healthcare application model and found that it outperforms PSO, shortest job first and round robin in terms of energy usage and cost [102]. Qiu *et al.* analyze the minimization optimization in fog computing-based Internet of Medical Things, which is considered a non-convex and non-linear problem [103]. The work considers the quality of service, power limit, and wireless constraint as optimization parameters. Abdel *et al.* propose a fog-based IoT platform for real-time diabetic patient tracking, using a hybrid strategy based on type-2 neutrosophic with the help of VIKOR method [104]. Hasse *et al.* present an e-health system that collects general and physiological health indicators from older people using My signals HW V2 technology with the help of a fog computing mobile application for monitoring health [105]. The recommended strategies have the advantage of handover latency. However, they are unsuitable for IoT fog systems since the message notifications, and distributed storage cannot be refreshed when moving [106].

The proposed research identifies genetic algorithm due to the fact that they use several sets in a search space where a search space is a collection of all possible solutions to the problem. It requires one objective function to calculate an individual's fitness and can work in parallel. Genetic algorithms operate on potential solutions' representations, known as chromosomes, rather than the actual solutions. Genetic algorithms are not guaranteed to produce global optimal solutions as well but genetic operators like crossover and mutation increase the likelihood of producing global optimal solutions. Genetic algorithms are stochastic and probabilistic in nature. With the right parameter setting, due to their large solution space, genetic algorithms are highly effective at handling multi-modal problems [107].

Because of the following characteristics, the work also uses the flower pollination method as the other meta-heuristic approach. Swarm intelligence (SI) optimization algorithms, which are modeled after numerous forms of biological behavior found in nature, have the advantages of being easy to use, performing well in optimization, and having strong robustness. The flower pollination algorithm is a meta-heuristic inspired by flowering plants for artificial intelligence. Flower pollination is the process of transferring pollen from one flower to another. Animals, such as birds, bats, insects, and so forth, are the principal actors in such transfers. Flowers and insects will form a flower-pollinator alliance. These blooms can attract birds which are part of the pollination, and these insects are the primary pollinators of the flowers. A flower and its pollen gametes provide a reliable answer to the optimization problem. With only one control parameter, FPA gives a simplified flower analogy with lightweight computing and provides a balanced intensification and diversity of solutions by implementing the Lévy flight and switch condition, which may be used to switch between local and global search. The pollinator transports pollen over greater distances to high-fitting flowers in case of global pollination; however, in other circumstances, local pollination is carried out inside a small area of an exclusive bloom. Local pollination can be used to replace phased elimination. Switch probability is a possibility for global pollination. The flower optimization algorithm, which was developed to address global optimization based on simulating the pollination process of flowers, has successfully addressed several optimization problems. FPA is distinguished by its formulation's simplicity, adaptability, and great computational performance efficiency. According to numerous studies, it can also outperform other well-known meta-heuristic optimization methods. As a result, FPA has been incorporated into several optimization studies and successfully used to solve numerous optimization issues in a variety of scientific domains.

To summarize, current research on edge/fog resource allocation does not sufficiently address resource allocation issues in mobility-aware microservice-based IoT applications using metaheuristic methods. Edge/Fog computing allows real-time processing of data generated by medical devices and wearable sensors, thus enabling remote patient monitoring, faster diagnosis, and more personalized treatment. Efficient resource provisioning is crucial for healthcare applications in edge/fog computing because of the reasons such as low-latency requirements, limited network bandwidth, and resource constraints. Therefore, we conducted a review of existing literature on resource allocation methods employed in healthcare applications, which led us to consider utilizing metaheuristic techniques for resource provisioning. We chose GA and FPA because similar applications of this category of heuristics, such as resource management on cloud infrastructure, have produced promising results.

We aim to develop a framework based on a meta-heuristic approach for mobile-aware IoT microservices that could be implemented on edge/fog computing scenarios for medical IoT applications. Our proposed approach involves modifying and integrating the mobility module within iFogSim2. Along with latency addressed in the existing works, other parameters like energy consumption, network use, cost, and execution time are also to be considered while developing a healthcare IoT system. The focus of this part of the work is to develop a resource provisioning solution using IoT microservices with mobility management by deploying metaheuristic methods for healthcare applications. The implementation details are discussed in Section 4.2.

2.6 Federated Learning in Edge/Fog/Cloud IoT applications

Federated Learning is appropriate for edge/fog/cloud computing applications and can use the computation power of servers and data gathered from widely scattered devices. Effective aggregation of client models is essential to create a generalized global model. The fundamental approach aggregates models from the distributed clients and obtains a new general global average model. The resultant model is then distributed to clients again for further training. Federated Learning makes use of different aggregation strategies for global model update. The following paragraphs discuss state of the art in aggregation methods in federated Learning, FL in edge/fog/cloud IoT applications, anomaly detection and Smart Decision Making module implementations in FL, and smart healthcare applications using FL.

2.6.1 FL aggregation methods

The literature proposes FedAvg as a privacy, security-preserving, and efficient communication aggregation algorithm for FL over-edge/fog devices. FedAvg assumes uniform involvement from all participants and excludes clients responding slowly [108]. The FedMA aggregation approach's foundation is a layer-wise learning strategy that matches and merges nodes with comparable weights. Here,the independently trained layers interact with the server [109]. FedProx addresses the heterogeneity issue in federated networks by allowing each participant device to execute a different amount of work. It incorporates partial information from stragglers and adds a proximal term to account for heterogeneity, which promises a steady and precise convergence behavior [49]. The principle of the FedPer approach is that the model is divided into personalized and base layers. While the personalized layers are not communicating with the server, the base layers are aggregated using transfer learning methodologies by the federated server [110]. FedDist is a Federated Learning aggregation algorithm based on the Euclidean distance dissimilarity measurement. This algorithm includes a few advantages of FedAvg, and FedMA [111]. Separating the local update process from the global aggregation results in a decrease in mobile devices' overall communication and computation costs. Also, in varying bandwidth conditions, empirical testing shows that the suggested EdgeFed is comparatively more efficient than state-of-the-art algorithms, with a decrease in the computational cost and the cost of interconnection for mobile devices. This is achieved by offloading a few calculations from mobile clients to the edge server [112].

A comparison of the above-discussed aggregation algorithms is presented in Table 2.8. Our proposed model uses FedAvg due to its easy deployment and less complicated implementation on Edge/Fog devices, thus resulting in reduced communication overhead.

Table 2.8: Comparison of Aggregation Algorithms in Federated Learning

Algorithm	**Complexity**	**Accuracy**	**Convergence**	**Cost**	**Speed**
FedAvg [108]	Low	High	Slow	Low	High
FedMA [109]	Moderate	Moderate	Moderate	Moderate	Moderate
FedProx [49]	Moderate	Moderate	Moderate	Low	Moderate
FedPer [110]	Low	High	Moderate	Low	High
FedDist [111]	High	Moderate	Moderate	Moderate	Moderate
EdgeFed [112]	High	High	High	High	High

2.6.2 Federated Learning in Edge/Fog/Cloud IoT applications

Xia *et al.* give new insight into federated learning's edge applications, development tools, communication effectiveness, privacy & security, scheduling, and migration [113]. Imteaj *et al.* examine the difficulties and problems of implementing FL in an IoT scenario [114]. Yu *et al.* offer a neural-structure-aware resource management approach with module-based federated learning, in which mobile clients are allocated with various sub-networks of the global model based on the condition of their local resources using both white box and black box approaches. Experiments show the effectiveness and flexibility of the strategy in utilizing resources [115]. Nguyen *et al.* evaluate the potential of FL for enabling a vast range of IoT services, including caching and data offloading for IoT devices, attack detection, location, crowd-sensing on mobile devices, and IoT privacy and safety. Additionally, a thorough analysis of the usage of FL in various critical IoT applications such as smart healthcare, unmanned aerial vehicles, smart transportation, smart cities, and smart industry is discussed [116].

A greedy heuristic method proposed in literature help in choosing the best fog node to act as a global aggregator. This helps in communication between the edge and the cloud and can lower the reliance on server-based execution. This FogFL architecture uses fog nodes to decrease energy consumption and communication latency of resource-constrained edge devices without influencing the rate of convergence of the global model, hence enhancing system dependability. Extensive deployment and testing claim that, in addition to fewer global aggregation rounds, FogFL holds 85% less energy and 92% less communication delay than state-of-the-art [117]. Zhou *et al.* utilize the combination of Paillier homomorphic encryption and blinding against model attacks to achieve the security aggregation of model parameters and enable IoT device data to fulfill differential privacy in resisting data attacks. Additionally, the proposal validates the scheme's ability to withstand collision attempts performed by numerous malevolent actors, guaranteeing both model and data security. The study implemented on the Fashion-MNIST dataset claims that the proposed technique is effective for real-world applications as well [118]. EdgeFed, draws inspiration from edge computing and aims to enhance the learning efficiency and reduce global communication frequency. It achieves this by separating the process of updating the local model, which is done independently by mobile devices. The edge server aggregates the outputs of these devices [112].

In order to reduce the model training loss and the overall time consumption, Zaw *et al.* develop an energy-aware resource management for mobile edge computing-enabled FL that takes into account the energy constraint of mobile devices and performs solution's convergence analysis and compare its effectiveness to the conventional FL technique [119]. To deliver FL as a Service (FLaaS) to industrial customers deployed on edge devices, Hiessl *et al.* suggest a FL system made up of a process description and software architecture. By grouping customers into cohorts with comparable data distributions, this method addresses skewed data [120].

2.6.3 Anomaly detection in IoT applications

Hasan *et al.* compares the effectiveness of various machine learning models to accurately predict attacks and anomalies in IoT systems. The machine learning algorithms evaluated include Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN) [121]. Abusitta *et al.* present an anomaly detection method for IoT, which utilizes deep learning to capture and learn resilient and beneficial features that remain unaffected by unstable environments. The extracted features are then used by the classifier to enhance the accuracy of malicious IoT data detection. The proposed deep learning model is based on a denoising autoencoder, which is utilized to derive features that can withstand the heterogeneous environment of IoT [122]. Chatterjee *et al.* provide an overview of the techniques used to identify abnormalities in IoT systems. It also discusses the algorithms that could be used for anomaly detection [123].

ECG Anomaly detection

Andrysiak *et al.* proposes a technique that combines the benefits and features of sparse representation of the analyzed ECG signal with the classification characteristics of the modeled neural network, in order to create a method that is both uncomplicated and efficient [124]. To overcome the limitations of current wearable devices used in ECG detection, Gu *et al.* suggest a heart rhythm abnormality classification model that is both lightweight and highly accurate based on traditional convolutional neural networks and the hardware acceleration techniques [125]. Nawaz *et al.* introduce an intelligent system that can automatically evaluate cardiovascular activity by detecting and classifying anomalies in raw one-dimensional (1D) ECG signals from end to end. The raw ECG data is carefully pre-processed before being stored in the cloud, and then analyzed in detail to identify any anomalies. For anomaly detection in the 1D ECG time-series signals, a deep learning-based auto-encoder (AE) algorithm is employed [126]. Ji *et al.* introduce a technique for detecting anomalies in univariate time series data using a long short-term memory (LSTM) algorithm. This method learns the structural characteristics of non-anomalous training data, and then applies a statistical approach to detect anomalies based on prediction error in the observed data [127].

2.6.4 Smart Decision Making in IoT applications

Cambra *et al.* showcase the benefits of using a tool that utilizes data in real time decision-making. The data includes variable rate irrigation and specific parameters derived from field and weather conditions. The decision making system processes data obtained from periodic sampling of field parameters, vegetation indices estimated through aerial images, and irrigation events like flow level, pressure level, and wind speed. The data is analyzed using a learning prediction system combined with the Drools rule engine in making decisions [128]. Kaur *et al.* propose a model that employs embedded sensors within a smart industrial system to gather data and identify the different industrial activities of employees. The identified activities are classified as positive, negative, or neutral. This information is then used to make cognitive decisions for employees based on game theory. The model aims to automate the cognitive employee evaluation system and decision-making process in smart industries, and it does so effectively and efficiently [129]. Bokhari *et al.* aim to explore the direct and indirect connections between Artificial Intelligence, Social Innovation, and Smart Decision Making. The results thus obtained help local governments to establish smart cities, where social innovation is incorporated into the decision making process. The study also emphasizes that smart decision making should involve social innovations and share collected data with entrepreneurs, businesses, industries, and social innovators to benefit the society and all the relevant stakeholders [130].

SDM in smart healthcare applications

Decision support systems (DSS) aim to provide experts with timely and relevant information. They offer tools for data processing, models, and knowledge to assist experts in making more informed decisions in various scenarios [131]. Zhou *et al.* suggest an approach for utilizing healthcare big data and involves a framework that enables smart and proactive data processing without requiring user interventions with an aim to maximize the utilization of data in decision making. The framework comprises of five stages: intelligent data cleaning, customized data fusion, analysis mapping, exploratory visualization analysis, and generation of decision-making reports [132]. Quasim *et al.* suggest a method for evaluating the technological integration efficiency of healthcare management using a Smart Healthcare Management Evaluation and Fuzzy Decision Making approach [133]. In this proposed study, the suggested strategy for SDM anomaly detection includes the utilization of an autoencoder, which is a type of unsupervised learning technique within the realm of AI. This neural network architecture is capable of compressing input data into a lower-dimensional representation and then reconstructing the input from this representation. This AI-based SDM can be implemented across various layers, including edge, fog, or cloud.

2.6.5 Federated Learning in healthcare

Among many applications, the healthcare sector deserves to be prioritized in terms of service quality compared to other domains. Critical functions such as simultaneous reporting and monitoring, tracking and alerts, and remote medical aid are all possible with IoT-based apps. The center for connected health policy conducted a study that observed that remote health monitoring systems lower the re-admission rates of heart failure patients by 50 percent [67]. Machine learning will not be able to realize its full potential or, eventually, make the leap from academic study to the clinical application without access to enough data. Rieke *et al.* examine the major contributing causes to this problem, evaluates the challenges faced in the field of digital health and discuss how Federated Learning can provide a solution [134]. Chen *et al.* propose FedHealth, a system that uses federated and transfer learning to aggregate and create reasonably personalized models. The model uses homomorphic encryption to ensure that no user data is leaked [135]. The design of a new aggregation protocol uses a secure hardware component and an Ethereum-native encryption toolkit to prevent the user data from leakage [136]. Kumar *et al.* present a framework that collects a modest amount of data from multiple hospital sources and uses blockchain-based Federated Learning to train a global deep learning model [137].

In order to train deep neural networks, Yuan *et al.* suggest an enhanced Federated Learning framework that helps the IoT device and the associated centralized server to overlook the training computation. The communication overhead is found to be decreased by the sparsification of activations and gradients. According to empirical research, the proposed system only necessitates less synchronization traffic than plain-vanilla Federated Learning while guaranteeing a low accuracy loss [138]. Analysis of the various Federated Learning systems, highlight the implications and potentials in healthcare and also summarises the general difficulties in using Federated Learning in the bio-medical domain [139]. Nguyen *et al.* present a state-of-the-art overview of the use of FL in healthcare domains, including smart health data management, remote medical monitoring, medical imaging, and COVID-19 detection. The major takeaways from the study are also emphasized, along with an analysis of several recent smart healthcare projects in Florida [140].

Chen *et al.* present a communication-efficient federated learning framework to address the critical challenge of communication delays in resource-limited IoT environments. Their approach introduces a probabilistic device selection scheme, prioritizing devices that significantly enhance FL convergence speed and training loss. Through novel quantization methods and efficient wireless resource allocation, the proposed framework demonstrates a notable improvement of up to 3.6% in identification accuracy and an 87% reduction in FL convergence time compared to standard FL methodologies [141]. In addressing the substantial privacy risks and communication challenges associated with executing machine learning tasks on human-generated data in the IoT, Briggs *et al.* emphasize the significance of FL as a privacy-preserving approach. The review underscores the critical need to mitigate communication costs linked to data transmission, adapt to heterogeneous conditions in IoT environments, and implement additional privacy safeguards within the FL framework. By evaluating various methods applied to FL, the authors offer valuable insights into its strengths and weaknesses, while also outlining future research directions, particularly emphasizing privacy-preserving FL applications in the context of IoT [142].

In the quest for large-scale and efficient deployment of AI through Edge Intelligence, Lim *et al.* advocate for the integration of FL as a privacy-preserving machine learning paradigm. Recognizing the communication inefficiency as a significant bottleneck in the envisioned FL network with thousands of heterogeneous distributed devices, the authors propose the Hierarchical Federated Learning (HFL) framework. Addressing the critical issues of resource allocation and incentive design within HFL, the article introduces a two-level mechanism employing evolutionary game theory for cluster selection at the lower level and a deep learning-based auction mechanism for model owner and cluster head interaction at the upper level, showcasing unique stability and revenue-maximizing properties through performance evaluation [143]. Beltrán *et al.* assert the growing importance of FL as a means to collaboratively train models without compromising sensitive data. Recognizing the limitations of Centralized FL (CFL) in terms of latency, vulnerability to failures, and trust concerns, the authors advocate for Decentralized Federated Learning (DFL) as a solution. Their comprehensive analysis addresses the gaps in the literature by examining the distinguishing aspects between DFL and CFL, evaluating DFL frameworks, reviewing application scenarios, and exploring mechanisms to optimize critical DFL fundamentals, thereby providing valuable insights for advancing decentralized approaches in FL [144]. Saha *et al.* propose a fog-enabled federated Learning framework, FogFL, to address challenges in distributed learning for delay-sensitive applications within resource-constrained IoT environments. Recognizing the communication overheads and security vulnerabilities associated with traditional FL, the authors introduce geospatially placed fog nodes as local aggregators to enhance the efficiency of the training process. Their approach employs a greedy heuristic for optimal fog node selection, mitigating the dependence on a centralized server and reducing communication latency and energy consumption by 85% and 92%, respectively, while maintaining the reliability of the global model's convergence rate [117]. Wu *et al.* highlight the challenges posed by device, statistical, and model heterogeneities in IoT environments for traditional FL. They propose a personalized federated learning framework in a cloud-edge architecture to address these issues and enable the deployment of intelligent IoT applications. The authors investigate emerging personalized federated learning methods that effectively mitigate the negative effects of heterogeneities, and through a case study on IoT-based human activity recognition, they demonstrate the efficacy of their proposed framework in achieving the requirements of fast-processing capacity and low latency for intelligent IoT applications [145].

Federated learning in the edge layer for medical anomaly detection is a promising approach to enable the development of accurate and efficient anomaly detection models while preserving the privacy and security of sensitive medical data. However, there are several research gaps that need to be addressed to fully realize the potential of FL in this domain. One of the research gaps is the incorporation of SDM modules across multiple layers of computing. To the best of our knowledge, no publications have addressed ECG anomaly detection using IoT microservice applications using SDM. This proposal aims to implement a microservice based Federated Learning model for one of the critical medical applications, ECG monitoring, which has improved data privacy, increased data diversity, more efficient use of resources and real-time updates. The proposed FedSDM model predicts the

ECG data anomalies by applying Federated Learning in edge, fog, and cloud layers and brings out a policy of usage at the appropriate level. The implementation details for the proposed approach are presented in Section 5.2.

2.7 Blockchain based Federated Learning in Edge/Fog/Cloud IoT applications

Recent research in the domain of Blockchain-based Federated Learning for edge, fog, and cloud IoT applications has seen a surge in interest. In their innovative work titled "Blockchain-Enhanced Federated Learning with Reputation Mechanism for Smart Home Systems," Zhao *et al.* introduce a FL system with a reputation mechanism to aid home appliance manufacturers in developing smart home systems. The system employs a two-stage workflow, where customers train an initial model using mobile phones and mobile-edge computing servers, and subsequently, models are signed, sent to the blockchain, and used for decentralized aggregation. The authors address privacy concerns through differential privacy (DP) on extracted features, demonstrating the superiority of their normalization technique over batch normalization under DP protection, while also incorporating an incentive mechanism to attract customer participation in the crowdsourcing FL task [146].

In response to the challenges posed by privacy concerns and communication costs in the context of the widespread utilization of IoT data, the work of Xuan *et al.* underscores the adoption of blockchain and federated learning technologies to address security issues related to collusion and privacy leakage. However, the study recognizes the emergence of "free-rider attacks" and "model poisoning attacks" in federated learning, necessitating the auditing of training models, which, in turn, amplifies communication costs. To mitigate this problem, the authors propose a communication cost optimization method based on security evaluation, introducing a double-layer aggregation model that combines competing voting verification methods and aggregation algorithms to effectively reduce the communication cost associated with node security verification in the blockchain-based federated learning process [147].

In their work, Zhao *et al.* address the limitations of traditional surveys in capturing comprehensive customer behaviors for IoT companies. Their intelligent system leverages FL technology to enable IoT device manufacturers to harness customer data and build accurate machine learning models predicting user requirements and consumption behaviors. The FL framework, consisting of collaborative training between customers and the edge computing server, employs differential privacy for feature extraction and utilizes blockchain to replace the centralized aggregator, offering enhanced privacy and security. Additionally, Zhao et al. introduce a novel incentive mechanism, awarding participants with coins to foster greater customer engagement in the crowdsourcing FL process, as suggested by their research [148].

Lu *et al.* propose a secure data-sharing architecture that leverages blockchain technology to enable distributed data sharing among multiple parties by incorporating privacy-preserving federated learning. In their design, federated learning is integrated into the consensus process of a permissioned blockchain which allows the computing work required for consensus in the blockchain to be used for federated training as well [149]. Pokhrel *et al.* use an autonomous design for federated learning in vehicular communication networking, utilizing a blockchain-based approach. This design aims to ensure privacy and efficiency in the system by exchanging and verifying local on-vehicle machine-learning model updates in a distributed manner [150].

Aich *et al.* suggests a system where its work flow involves two stages: customers train an initial model using their mobile phones and the mobile edge computing (MEC) server, then send the signed models to the blockchain for protection against malicious activities. In the second stage, manufacturers choose customers as miners to calculate the averaged model using the received models, employing differential privacy and a novel normalization technique to safeguard customer privacy and enhance test accuracy [151]. Lu *et al.* suggest an asynchronous federated learning method to train models using edge data, optimizing efficiency by carefully selecting participating nodes to reduce total costs. Additionally, boost model reliability by incorporating learned parameters into the blockchain and ensuring their quality through a two-stage verification process [152]. Lu *et al.* introduce a blockchain-based federated learning model that replaces the central authority with a specially designed blockchain featuring decentralized privacy protocols where local updates from end devices are uploaded to fog servers, generating and storing global updates. The blockchain efficiently maintains only the pointer to global updates, while data is saved using a distributed hash table (DHT), ensuring robustness, privacy, and protection against poisoning attacks on fog servers [153].

In practice, hospitals and relevant organizations are hesitant to share patients' data to safeguard patient privacy, making it challenging to access critical information for cognitive disease detection. Nevertheless, wearable devices and advancements in computing technology enable the collection of valuable health information, and smart healthcare leverages machine learning models trained on abundant user data while preserving privacy through blockchain integration [154].

The following paragraphs present the current state of the art in blockchain-based federated learning methods used in medical applications. FedHealth employs federated learning and homomorphic encryption to aggregate data from multiple organizations, creating personalized models through transfer learning while strictly preserving user privacy without any data leakage during the parameter sharing process [135]. A novel Secure Aggregation protocol is proposed by Passerat *et al.*, combining a secure hardware component and an encryption toolkit native to Ethereum [136]. Kumar *et al.* introduces a framework that gathers a small amount of data from multiple sources, such as various hospitals and employs blockchain-based federated learning to train a global deep learning model [137].

Passerat *et al.* introduce the aggregation actor as a trusted third-party or a central entity with secure hardware like Intel SGX, acts as the primary source of centralization by collecting updates from participants to create a new model version. However, this centralization comes with risks, such as training interruption if the server fails or potential malicious actions affecting the training process. To address these issues, it uses blockchain as a decentralized alternative to coordinate the process [155]. The presence of incorrect masked gradients and unmasked shares uploaded by dishonest local trainers to the parameter server, undermines the integrity of FL and hinders its ability to attract sufficient distributed training data and computation power. To address this, Bao *et al.* proposes FLChain, a decentralized, public auditable, and incentivized FL ecosystem that ensures trust and incentive where FLChain nodes collect and combine locally documented gradients, then submit the aggregated results back to FLChain [156].

Various methods exist for determining the mining process participants and the type of data being updated on the blockchain, as well as the specific location where the aggregation of local models into a global model occurs. These methods depend on the architecture and design of the blockchain-based federated learning IoT application. The selection of devices participating in the mining process can be based on factors such as computational power, network connectivity, or pre-defined roles. The data being updated on the blockchain can include local model updates, training progress, or consensus-related information. The specific use case and the desired level of transparency and security, influence the decision on what data to store on the blockchain. The aggregation of local models can happen either at a central server or through a distributed consensus algorithm. In the central server approach, all participating devices send their local model updates to a centralized entity responsible for aggregating the models into a global model. On the other hand, in a distributed consensus algorithm, devices collaborate directly with each other to collectively update the global model. The choice between these methods depends on factors like the scale of the IoT network, communication latency, privacy requirements, and the desired level of decentralization. Each approach has its advantages and trade-offs, and the final design should align with the specific needs and goals of the blockchain-based federated learning IoT application [157, 158].

Several methods have been proposed in the state-of-the-art literature for developing a global model in a blockchain-based federated learning IoT application using fog/edge devices. One method involves fog/edge devices actively participating in the mining process and collaboratively developing the global model with the assistance of consensus algorithms. The updated global model is then broadcast to end devices, ensuring all participants have the latest version. In another method, fog/edge devices also participate in the mining process, but instead of directly developing the global model, they send their local model updates to the network. End devices receive these updates and collectively generate the global model through aggregation. Lastly, a combination of fog/edge devices and end nodes collaboratively participate in the mining process to develop the global model collectively. Each method offer unique advantages and challenges, and the choice depends on factors like network scale, privacy concerns, and the desired level of decentralization [159–163]. A summary of the related work on blockchain can be seen in Table 2.9.

The utilization of blockchain based federated learning in the edge/fog/cloud layer for medical anomaly detection holds great promise for developing accurate and efficient anomaly detection models while ensuring the privacy and security of sensitive medical data. However, several research gaps remain that require attention to fully harness the potential of blockchain and FL in this field. One such gap pertains to incorporating SDM modules across multiple layers of computing. As of now, no published works have addressed ECG anomaly detection in IoT microservice applications using SDM. In the current literature, there is a lack of research on actively involving both participating edge/fog devices and end devices in the mining process to collectively enhance the quality of the global model in blockchain-based federated learning IoT applications. This work aims to propose a blockchain-based federated learning model for a critical medical application, ECG monitoring, which offer improved data privacy, increased data diversity, more efficient resource utilization, and real-time updates by enabling active participation from both edge/fog devices and end devices in the mining process to improve the global model's quality collaboratively. The proposed FedSDM model predicts ECG data anomalies by implementing federated learning in edge, fog, and cloud layers while also providing guidelines for usage at the appropriate level. The implementation details for the proposed approach are detailed in Section 6.2.

Table 2.9: Summary of related works on Blockchain

Reference	Application Area	Advantages	Disadvantages
Lu *et al.* [149]	Industrial IoT	Privacy preserving FL with secure data sharing	Consensus on permissioned Blockchain
Pokhrel *et al.* [150]	Vehicular Communication Networking	distributed exchange and verification of on-vehicle model updates	On-vehicle update complexities.
Aich *et al.* [151]	MEC	Two-stage privacy-protecting workflow	Blockchain overhead potential.
Lu *et al.* [152]	Edge Data Processing	asynchronous federated learning and blockchain-based parameter integration	Asynchronous updates integration complexity.
Qu*et al.* [153]	Decentralized Privacy	Decentralizing authority against poisoning attacks	Decentralized privacy protocol challenges.
Chen *et al.* [135]	Medical Applications	Homomorphic encryption and data confidentiality in federated learning	Homomorphic encryption implementation complexity.
Passerat*et al.* [136]	General Blockchain	Enhanced security through a secure aggregation protocol	Ethereum ecosystem reliance for encryption toolkit.
Kumar *et al.* [137]	Medical Applications	Blockchain-based federated learning for global deep learning model	Multi-source cooperation requirement.
Passerat *et al.* [155]	Various Applications	Risk-mitigating decentralization	Third-party trust reliance possibility.
Bao *et al.* [156]	Various Applications	Incentivized participation in decentralized and auditable FL ecosystem	FLChain implementation complexity.

2.8 Mobility in Edge/Fog/Cloud IoT applications

In the realm of IoT applications, incorporating mobility is paramount to unlock the full potential of connected devices and ensure the effectiveness of the deployed solutions. Mobility introduces dynamic and ever-changing environments where devices, sensors, and users are not static but constantly on the move. By accounting for mobility, IoT applications can seamlessly adapt to changing contexts, enabling real-time data acquisition and analysis irrespective of the location of devices. This is particularly crucial in scenarios such as smart cities, healthcare, and industrial settings, where the movement of people, assets, or equipment is inherent. A mobile-aware IoT application enhances scalability, responsiveness, and the system's overall efficiency, ensuring that it remains resilient and continues to deliver meaningful insights and services even in dynamic and diverse operational landscapes. Consequently, by acknowledging and accommodating mobility in the design and implementation phases, IoT applications can provide more agile and robust solutions that better align with the dynamic nature of the connected world. The upcoming section provides an overview of the literature survey focusing on mobility within applications spanning edge, fog, and cloud computing environments.

Jayasena *et al.* utilize a whale optimization meta-heuristic algorithm for optimizing task scheduling in a smart healthcare application model, demonstrating superior performance over PSO, shortest job first, and round robin in terms of energy usage and cost [102]. Qiu *et al.* analyze minimization optimization in fog computing-based Internet of Medical Things, addressing a non-convex and non-linear problem, taking into account factors like quality of service, power limits, and wireless constraints as optimization parameters [103]. Abdel *et al.* propose a fog-based IoT platform for real-time diabetic patient tracking, employing a hybrid strategy based on type-2 neutrosophic logic with the assistance of the VIKOR method [104]. Hasse *et al.* present an e-health system that collects general and physiological health indicators from older individuals using MySignals HW V2 technology, employing a fog computing mobile application for health monitoring. This approach offers an advantage in terms of handover latency [105]. However, these strategies are not suitable for IoT fog systems since message notifications and distributed storage cannot be refreshed while in motion [106].

Table 2.10: Literature Survey on Microservices in IoT Applications

Reference	**Objectives/Findings**
Benayache *et al.* [164]	Microservices architecture offers high-service decoupling and is suitable for IoT applications
Yu *et al.* [165]	Microservices architecture demonstrates exceptional performance in IoT applications
Zhao *et al.* [166]	Proposed microservice architecture for low-latency fog computing applications
Abdullah *et al.* [167]	Introduced predictive autoscaling for microservice applications in fog computing, reducing rejected requests and SLA violations
Samodha *et al.* [168]	Analyzed microservice integration for IoT applications in fog computing, focusing on application modeling, placement composition, and performance evaluation
Thanh *et al.* [169]	Introduction of broker-less architecture in the Internet of Healthcare Things platform for healthcare applications

2.9 Microservices in Edge/Fog/Cloud IoT applications

Microservices play a significant role in edge/fog/cloud IoT applications by enabling a modular and scalable architecture that allows for efficient resource utilization and rapid deployment of services across these distributed environments, ultimately improving responsiveness and flexibility in managing IoT workloads. This approach also enhances fault tolerance and enables better management of resources, making it well-suited for handling the dynamic and diverse nature of IoT data processing and analytics.

The microservice architecture is geared towards breaking down the system into small, self-contained components that are interconnected by shared services. This architecture offers a higher degree of service decoupling compared to traditional service-oriented and monolithic architectures, as noted by Benayache *et al.* [164]. Microservices architecture is gaining prominence due to its intrinsic characteristics, including small granularity and low coupling, making it a preferred design approach for deploying and updating IoT applications. Yu *et al.* [165] highlights its exceptional performance and suitability in the context of IoT applications. In this architecture, each microservice is responsible for a specific sub-task or service, which translates to reduced computational resource requirements and lower communication overhead.

Zhao *et al.* propose an architecture based on microservice containers in the fog system, specifically designed for executing mobility applications that demand low latency and cost-effectiveness. Their cost calculation method includes both computation and communication expenses [166]. Abdullah *et al.*, on the other hand, introduce a novel approach involving predictive autoscaling of microservice applications within containerized fog computing infrastructure, demonstrating fewer rejected requests and SLA violations compared to existing systems [167]. Samodha *et al.* delve into the unique aspects of scheduling microservices-based applications in fog computing, distinct from other application models. Their work involves the analysis of microservice integration for IoT applications, encompassing application modeling, placement composition, and performance evaluation [168]. This microservices architecture is recommended for its simplicity in updating and deploying fog-based IoT applications, underpinned by its fundamental attributes such as small granularity and low coupling.

Finally, the introduction of the Internet of Healthcare Things (IoHT) platform in the healthcare domain employs a broker-less architecture for various purposes, including data collection, user management, device management, and remote device control [169]. The preceding studies are summarized in Table 2.10.

2.10 Simulation Tools

In the dynamic landscape of edge, fog, and cloud computing, simulation tools serve as indispensable aids for researchers and practitioners in comprehending and optimizing complex architectures and applications. Specifically, tailored simulation tools for edge and fog environments play a crucial role in evaluating diverse scenarios, resource allocation strategies, and application deployments. A literature survey on simulation tools for edge and fog environments is essential to provide a comprehensive understanding of existing tools, methodologies, and frameworks. Such a survey facilitates the identification of key features, capabilities, and limitations of various simulation tools, aiding in informed decision-making and staying abreast of the rapidly evolving landscape.

Puliafito *et al.* introduce MobFogSim, an extension to the iFogSim simulator designed to account for user mobility, wireless connectivity, and the virtual machine/container migration process [45]. Isaac *et al.* propose YAFS, Yet Another Fog Simulator tailored for fog computing environments, which models network failures and enables the evaluation of service placement solutions under failure scenarios by dynamically creating/deleting cloudlets and network links, along with runtime event implementation [170]. FogNetSim++ is a framework for constructing network simulators that extends OMNeT++14 to replicate all aspects of energy consumption, pricing, mobility, and handoff mechanisms [171]. The preceding studies are summarized in Table 2.11.

The market is highly competitive in terms of simulators designed for edge, fog, and cloud device simulations. The choice for modeling and simulating edge/fog/cloud computing infrastructures and services for the proposed system is iFogSim2, an extension of Cloudsim. This framework allows for the development and execution of experiments involving edge/fog/cloud devices, covering aspects like compute, memory, I/O, VM allocation, and VM power models. iFogSim2, building upon the iFogSim simulator, possesses features such as service migration, distributed cluster establishment across fog nodes, and microservice orchestration, crucial for validating the proposed approach's performance in fog computing environments. Its components, including mobility, clustering, and microservices, are modular and can be adapted for various simulation scenarios. Notably, iFogSim2 sets itself apart by incorporating real datasets to evaluate different service management strategies in fog computing contexts, a feature lacking in most existing solutions. It offers methodologies for node clustering, mobility management, and microservice orchestration, serving as valuable benchmarks for performance comparison [44].

Table 2.11: Literature Survey Summary of Simulation Tools

Reference	**Tool**	**Realtime or Simulation**	**Mobility**	**Migration**	**Clustering**
Puliafito *et al.* [45]	MobFogSim	Simulation	✓	✓	
Lera *et al.* [170]	YAFS	Simulation	✓		
Qayyum *et al.* [171]	FogNetSim++	Simulation	✓		
Proposed	**iFogSim2**	**Simulation**	✓	✓	✓

2.11 Summary of Literature Survey and Research gaps

2.11.1 Multiobjective Optimization

The literature review could be summarized as follows: various papers evaluated performance for healthcare systems, taking into account parameters such as latency, real-time processing, response time, decision-making, scalability, mobility deployment, dynamic configuration, network traffic, battery, energy consumption, and bandwidth. The primary shortcoming of the existing works is that many of the works concentrate on single-use cases and therefore only discuss infrastructure and services that are adequately specialized. The current solutions are either single objective optimization of a single parameter or a combination of single parameter optimizations. Considering all parameters of simulation such as latency, real-time computation, response time, decision making, scalability, mobility implementation, dynamic configuration, network traffic, power, energy consumption, cost and bandwidth into a single problem in fog computing is still an open research. The current implementations have not addressed the provision of resources to meet expected service response times for efficient service in case of emergency applications. The consolidated literature survey on the parameter used for comparison is presented in Tables 2.12 and 2.13 summarizes the paper along with their application domain and the parameter used. It also highlights if the model is experimental or real time or simulation.

To address a few of the above issues, the proposed approach explores smart resource provisioning and decision-making in fog computing where objective function maximizes application efficiency by considering multiple parameter optimizations. The parameters considered are energy, network use, execution time, cost and delay. The problem description of the proposed model is explained in the next section.

2.11.2 Metaheuristic methods

The literature explores meta-heuristic algorithms for resource provisioning in edge and fog computing, notably in healthcare-focused IoT microservices. However, a research gap exists in addressing challenges posed by mobility-aware microservice-based IoT applications in healthcare. Current studies lack specialized investigation into healthcare IoT microservices' unique requirements and integrating effective mobility management techniques within metaheuristic resource provisioning models. Bridging these gaps is crucial for developing effective meta-heuristic approaches to address critical healthcare application needs, such as real-time patient monitoring and diagnostics in dynamic healthcare environments.

2.11.3 Federated learning

Federated Learning in the edge layer for medical anomaly detection is a promising approach to enable the development of accurate and efficient anomaly detection models while preserving the privacy and security of sensitive medical data. However, there are several research gaps that need to be addressed to fully realize the potential of FL in this domain. One of the research gaps is the incorporation

Table 2.12: A Summary of Related Work based on application domains

Reference	Use case	Research domain	Mode	Feature comparison				
				Energy	Cost	Delay	NU	ET
Harshit *et al.* [64]	Online game	Real-time processing	Simulation	✔		✔	✔	
Mishra *et al.* [65]	Mission critical	Real-time processing	Simulation	✔		✔		
Shreya *et al.* [66]	Time-Critical	Hierarchical processing	Real time	✔		✔		
Das *et al.* [172]	Geospatial	Geospatial data processing	Simulation	✔		✔		
Nashaat *et al.* [69]	IoT	IoT data processing	Simulation	✔		✔	✔	
Kumari *et al.* [70]	Smart Healthcare	Security and privacy	Simulation		✔			
Zohora *et al.* [71]	Time sensitive	IoT medical data processing	Simulation	✔		✔	✔	✔
Mahmud *et al.* [33]	Smart Healthcare	IoT medical data processing	Simulation	✔		✔	✔	✔
Tuli *et al.* [99]	Remote healthcare	IoT medical data processing	Real time	✔			✔	
[Proposed]	Smart healthcare	IoT medical data processing	Simulation	✔	✔	✔	✔	✔

Table 2.13: Comparison of Various Evaluation Parameters

Parameter	Reference					
	M Ahmad et.al 2016	Fatema Tuz Zohora et.al 2017	Jianhua Li et.al 2015	Jayneel Vora et.al 2017	Redowan Mahmud et.al 2018	George et.al. 2018 Gill et.al 2018
Execution time	✓	✓				
Response time						
Delay/Latency			✓	✓	✓	✓
Bandwidth		✓				✓
Energy consumption		✓			✓	✓
Real time processing					✓	
Cost		✓				
Throughput						
Data overloading		✓		✓		
Data consistency						
Fault tolerance						

of smart decision making (SDM) modules across multiple layers of computing. To the best of our knowledge, publications have addressed ECG anomaly detection using IoT microservice applications using SDM. This work aims to propose a microservice-based federated learning model for one of the critical medical applications, ECG monitoring which has improved data privacy, increased data diversity, more efficient use of resources and real-time updates. The proposed FedSDM model predicts the ECG data anomalies by applying federated learning in edge, fog, and cloud layers and brings out a policy of usage at the appropriate level.

2.11.4 Blockchain based Federated learning

In the present state of the field, there is a noticeable research gap surrounding the collaborative involvement of fog/edge devices and end nodes in the mining process, working together to develop a global model collectively. This innovative approach diverges from the traditional method of constructing the global model directly. Instead, these entities choose to distribute their local model updates with the network, with the global model generation taking place at the end device.

Since the healthcare issues related to ECG anomaly detection in microservice-based IoT systems are not sufficiently addressed by existing research were motivated to do this study.

2.12 Objectives and Research goals

The significant contributions of our work are as follows:

- To propose a resource provisioning solution using IoT microservices with mobility management for healthcare applications.
- To implement multiobjective optimization using the weighted sum method and to optimize the key parameters associated with the application
- To utilize modified metaheuristic scheduling techniques for efficient resource provisioning in fog and edge devices
- To design an early warning system for ECG anomalies using Smart Decision Making module
- To integrate Blockchain-based Federated learning, a privacy-preserving method, into critical healthcare applications to protect end-user data.
- To identify the most suitable placement policy for deploying the Blockchain based Federated learning module within the architecture's Edge, Fog, and Cloud layers
- To simulate a set of experiments to validate the effectiveness of our proposed solution under real workloads in terms of energy consumption, network use, cost, execution time, and latency.

Chapter 3

Resource Provisioning based on Multiobjective Optimization

3.1 Introduction

This section describes the proposed solution to the resource provisioning problem using multiobjective optimization methods. The approach addresses the resource provisioning problem as a multiobjective optimization problem with the objective of minimizing the evaluation parameters considered in this work.

A literature survey on different multiobjective optimization methods for parameter optimization in edge and cloud computing reveals a diverse range of techniques used to balance key factors in achieving efficient and responsive systems. Different approaches are described in the below paragraph. Many studies utilize Pareto-based methods, such as Non-dominated Sorting Genetic Algorithm (NSGA) and Strength Pareto Evolutionary Algorithm (SPEA), to find a set of solutions that represent the Pareto front—solutions that cannot be improved in one objective without deteriorating another. These methods provide a comprehensive understanding of the trade-offs between parameters by offering a range of optimal solutions. Evolutionary algorithms, such as Genetic Algorithms and Particle Swarm Optimization, have also been widely employed for multiobjective optimization in edge and cloud computing. These algorithms iteratively evolve a population of potential solutions by mimicking natural selection. They allow for exploring the trade-off between parameters by optimizing resource allocation, task scheduling, and data distribution parameters.

Multiobjective particle swarm optimization extends the traditional PSO algorithm to handle multi-objective optimization problems. It uses particle movement to explore the pareto front, making it well-suited for optimizing parameters in edge and cloud computing scenarios. Fuzzy logic has been applied to model and optimize the relationship between energy and latency. Fuzzy-based methods enable handling uncertainties and imprecise information, which are common in edge and cloud environments. These approaches provide a more realistic representation of the problem by considering factors like data variability and dynamic workload. Ant Colony Optimization (ACO) is inspired by the foraging behavior of ants and has been used for solving multiobjective optimization problems in edge and cloud computing. It optimizes parameter settings by simulating ant behavior, allowing for the discovery of optimal trade-offs between energy and latency. Machine learning techniques, including reinforcement learning and neural networks, have been applied to optimize parameters in edge and cloud systems. These methods can dynamically adjust parameters to achieve desired energy-latency trade-offs by learning from historical data and real-time observations. Hybrid optimization techniques combine multiple methods to leverage and mitigate their strengths. For example, a hybrid approach might combine GA and PSO to achieve better convergence and diversity in the solution space. Game theory approaches consider multiple entities' interactions and strategic decisions in a system. Such approaches can model the interactions between edge devices and cloud resources to optimize energy-latency trade-offs while considering the behaviors of different players. Other biologically inspired optimization methods, such as Bee Colony Optimization and Firefly Algorithm, have also been explored for multiobjective optimization in edge and cloud computing. These methods draw inspiration from natural phenomena to search for optimal solutions [173–176].

The weighted sum method is a commonly used technique in various fields, including mathematics, engineering, economics, and decision making, to aggregate multiple factors or criteria into a single composite score. This method involves assigning weights to each factor based on its relative importance and then computing a weighted sum of these factors to arrive at an overall value. The weighted sum method can be categorized under the "Pareto-based Approaches" as it aims to find a set of solutions that represent the pareto front by evaluating trade-offs between different objectives. In the context of multiobjective optimization for parameter optimization in edge and cloud computing, the weighted sum method can be considered as a simplified form of pareto based optimization. The weighted sum method calculates a single composite score for each solution by linearly combining the individual objectives using predefined weights. These weights represent the relative importance of each objective. While the weighted sum method does not explicitly generate a diverse set of pareto-optimal solutions like some other pareto-based algorithms, it effectively explores the trade-offs between objectives based on the given weights.

By adjusting the weights assigned to the parameters, decision makers can navigate the trade-off space to find solutions that align with their preferences. However, the weighted sum method assumes a linear relationship between objectives, which might not accurately capture complex interactions and trade-offs. In summary, while the weighted sum method shares similarities with pareto-based approaches in addressing multiobjective optimization, it offers simplicity and ease of interpretation. It provides a way to explore the trade-offs between parameters by assigning weights to objectives, making it a practical and accessible method for decision-making in edge and cloud computing scenarios.

The weighted sum method offers several advantages in multi-criteria decision making and analysis. By assigning appropriate weights to individual factors, decision makers can effectively reflect the relative significance of each criterion in the decision process. This allows for a structured approach to consider multiple factors simultaneously, promoting a holistic evaluation of options. Furthermore, the weighted sum method provide a straightforward interpretation of results. The computed composite score directly represents an option's aggregated preference or performance across various criteria. This transparency aids in communicating the rationale behind decisions to stakeholders and facilitates a clear understanding of the decision making process.

Flexibility is another advantage of the weighted sum method. It accommodates a wide range of criteria types, whether quantitative or qualitative, making it adaptable to diverse decision scenarios. The method's flexibility also enables decision makers to adjust weights based on changing circumstances or preferences, ensuring that the model remains responsive and reflective of evolving priorities. In cases where precise measurements or data may be lacking, the weighted sum method can still be employed by relying on expert judgment to assign weights and evaluate criteria. This allows for informed decisions even in situations with limited available information.

In conclusion, the weighted sum method offer a practical and intuitive approach to multi-criteria decision making. Its simplicity, interpretability, flexibility, and capacity to handle a variety of criteria types make it a valuable tool for aiding decisions across numerous domains while also demanding thoughtful consideration of weight assignments and potential model limitations. Therefore, we have opted for the weighted sum technique in our proposed approach.

3.2 ECG anomaly detection

The proposed approach uses ECG dataset to detect the anomalies in the ECG signal. Real time ECG abnormality detection is one of the applications in medicine that has several advantages for patient care. First and foremost, it allows healthcare providers to quickly identify and respond to cardiac abnormalities, potentially saving lives. Early detection and treatment of cardiac abnormalities can prevent more serious and costly health issues down the road. Moreover, real time ECG anomaly detection can help reduce healthcare costs and improve patient outcomes. By continuously monitoring ECG signals in real time, the system can immediately detect anomalies and alert healthcare providers, who can take action to diagnose and treat the patient. Another benefit is that real time ECG anomaly detection can improve the accuracy of diagnoses. In some cases, anomalies may be missed or misinterpreted when relying on visual inspections alone. With automated detection, the system can analyze the ECG signals with greater precision, reducing the risk of errors and false negatives. Additionally, real time monitoring can help identify potential issues before they become acute, reducing the likelihood of hospitalizations and emergency room visits. Overall, real-time ECG anomaly detection has the potential to improve patient care, increase accuracy, and reduce healthcare costs, making it a valuable tool in healthcare.

3.3 Architecture for Resource Provisioning based on Multiobjective Optimizations

This section elaborates on the modules and the components needed for fog based architecture in the proposed healthcare application. The current implementation uses a cloud based structure where all the edge devices communicate with the necessary services with the help of the cloud. However, this may introduce a certain delay and create latency for the entire process, which may be critical for the patients in an emergency. To overcome this, the architecture of the proposed system uses an edge/fog integrated cloud implementation, which reduces the delay and latency when compared with the existing one.

3.3.1 Existing Architecture - Cloud based

All the components used in the work are assumed to be connected via connectivity technologies for IoT wireless applications like ZigBee, Wi-Fi, Z-wave, and Bluetooth so as to enable machine-to-machine interaction and human-to-machine interaction. The modules in the existing cloud architecture are detailed below.

- Cloud datacenter: Data centers are centralized places with the combination of multiple servers that can handle huge amounts of data computing, storage, and networking. For medical IoT applications, medical data from the sensors is transmitted to cloud data centers, and the required action commands for the actuators are transmitted from the data centers. Data pertaining to medical history is also processed permanently in the cloud at the same time.

- Resource management components: The key components of resource management include provisioning, scheduling, and monitoring. Resource provisioning involves identifying, deploying, and managing software and hardware resources. Resource scheduling is a set of actions that will effectively allocate the resources to complete the task based on resource availability. Cloud monitoring deals with the assessment, tracking, and management of software and hardware resources in the cloud that are being used by applications.

- Smartphones

  Technologies such as Bluetooth, RFID monitoring, and Near-field Communications (NFC) allow smartphones, as part of IoT medical health care, to collect sensor data and perform necessary action commands to actuators.

- Wearable sensors and actuators

  The principal role of the smart sensor node is to track environmental conditions [177]. Medical Sensors are used for detecting and responding to changes in the human body. Sensing technology is evolving so quickly in such a way that within a few years, we will see trillions of medical sensors being deployed for complex health care applications. A medical actuator is a device for medical equipment movement and control.

The data between the modules in the above-said cloud architecture is shown in Figure. 3.1. The flow of data is as follows

1. Transfer of captured sensor data to the network.

2. Transmission of raw data through resource management components.

3. Transmission of data to the cloud for processing.

4. Response of the resultant communication after cloud processing.

5. Response command transmission through network elements.

6. Transfer of actuation command to the concerned module of operation.

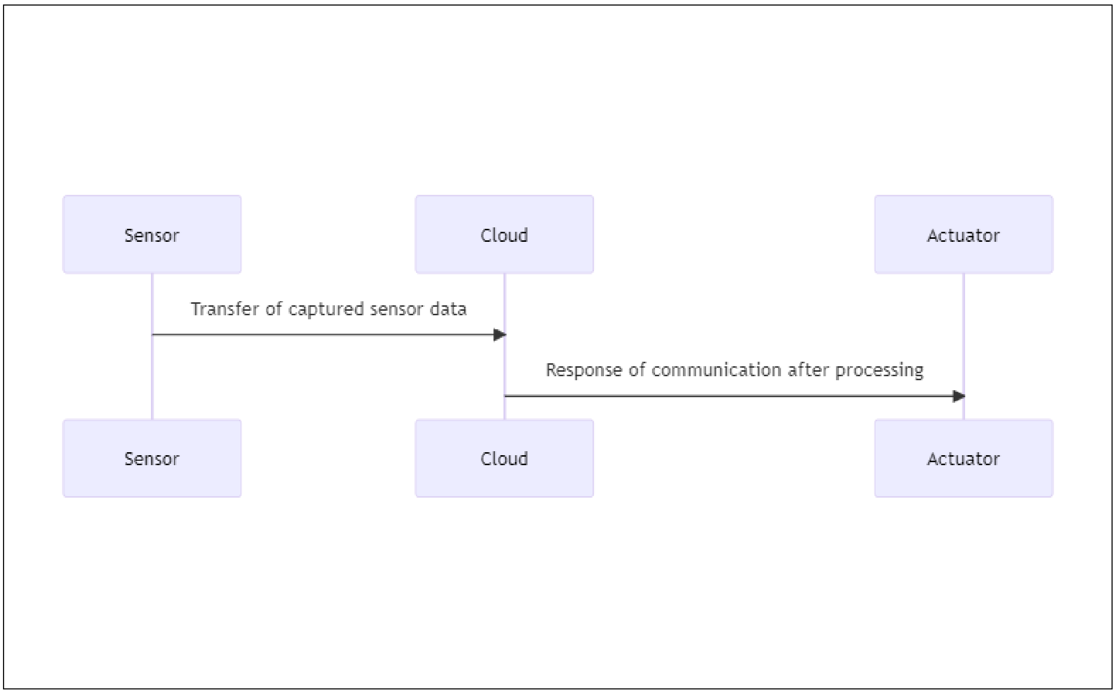


Figure 3.1: Data flow in Cloud-based architecture

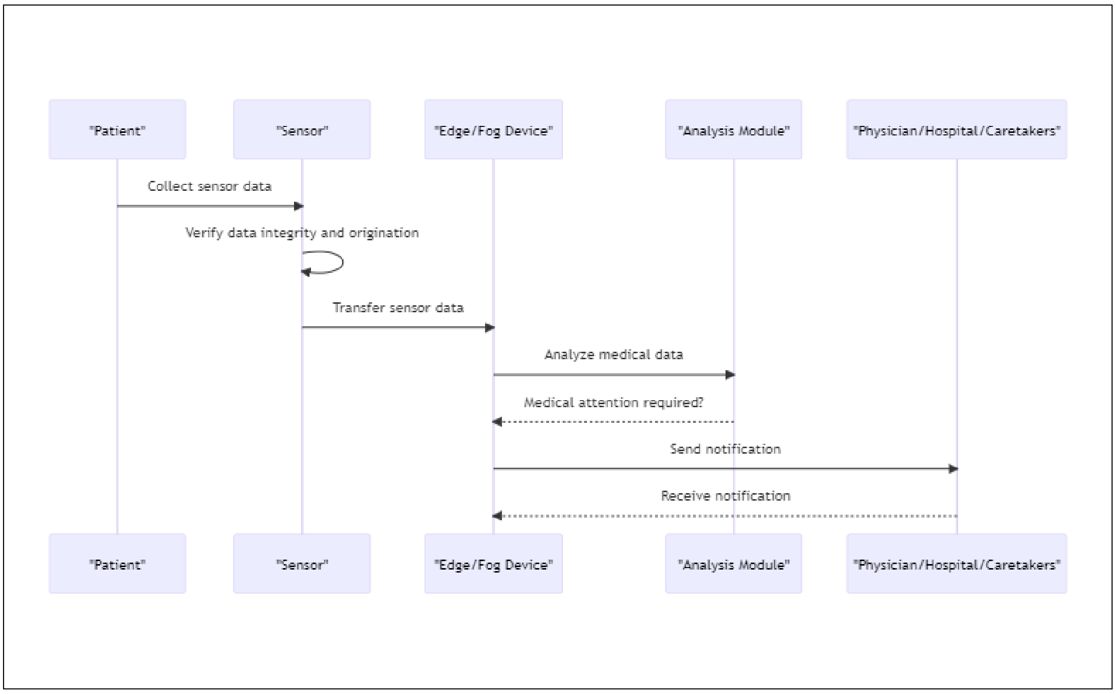


Figure 3.2: Data flow in Edge-based architecture

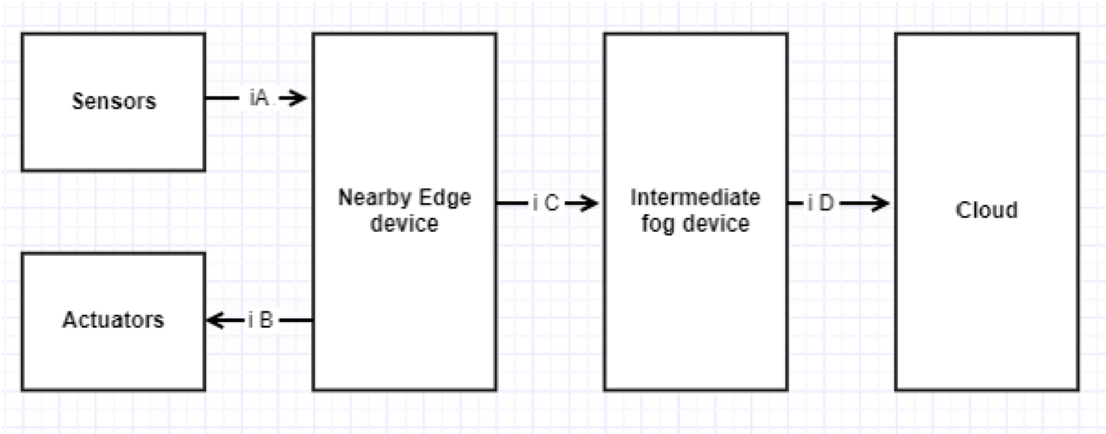


Figure 3.3: Flow of data in architecture based on Edge computing

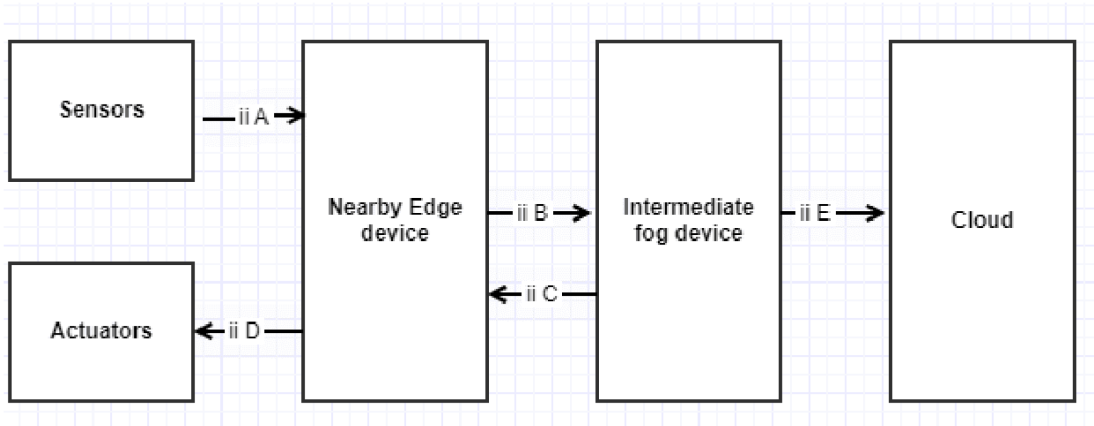


Figure 3.4: Flow of data in architecture based on Fog computing

3.3.2 Proposed architecture - Edge/Fog based

The architecture is based on a set of expectations that are extracted from the essential application of health care services. If successfully applied, edge/fog computing can reduce the latency experienced in QoS and minimize the bandwidth usage in any healthcare application and later can be extended to other time-critical applications. The main difference between fog-based and cloud-based systems is the computing and storage capability of the fog devices between the patient and the cloud data center. With the idea of fog technology, the underutilization of intermediate devices can also be addressed. An edge/fog-based integrated IoT application system architecture presented in Figure 3.2 has the following steps.

1. Collection of sensor data from the patient.
2. Verification of sensor data for its integrity and origination.
3. Transfer of sensor data to nearby edge/fog devices such as the patient's smartphone or laptop in the patient's room.

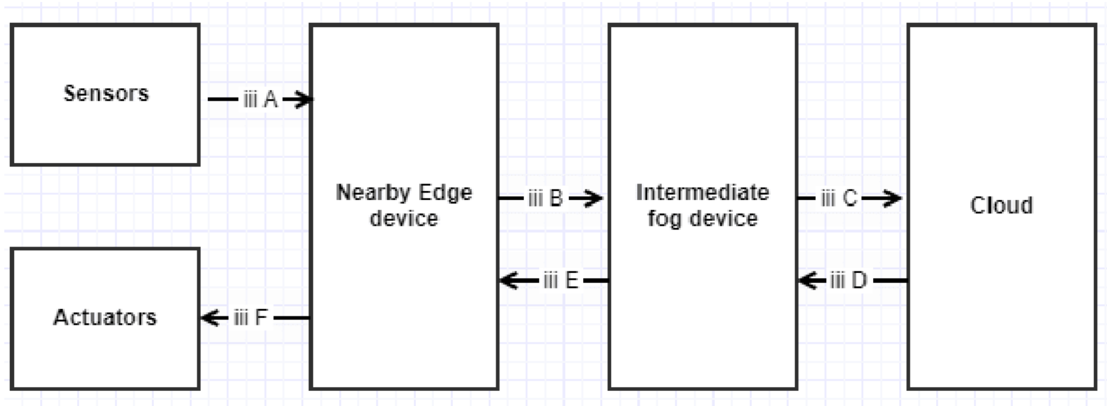


Figure 3.5: Flow of data in architecture based on Cloud computing

4. Analysis of medical data for any medical attention requirement.

5. Sending regular notifications to the physician/hospital/ caretakers concerned.

Edge/Fog computing based solution consists of advanced networking devices called fog nodes and edge devices for computing tasks.

To conclude the above, the steps involved in the proposed edge/fog based architecture include the following

- Sensor data validity testing
- Data filtering and processing of authorized sensor data
- Comparison of ECG data with preset threshold value
- Transmission of output data to the actuator

In the case of cloud architecture, all the above-described steps happen only in the cloud data center and all the intermediate nodes just act as data forwarding devices, while in the case of fog, these steps can happen at three levels, namely

- Processing at the nearest edge device
- Processing at the intermediate fog devices
- Processing at the cloud data center

Below is a description of the data flow for the above types. The detailed steps that each of the above-described levels follow are described as types. The numbering mentioned at the beginning of the step is related to the numbering given in the architectures presented in Figures. 3.3, 3.4 and 3.5.

Type i

i A Transmission of ECG sensor data to the nearest edge device such as the smartphone of the patient or the laptop placed in the room of the patient.

i B Verification of authenticity, filtering and comparison of ECG sensor data with the preset threshold and automatic calling of nearby emergency services.

i C Forwarding permanent medical history of patient data to intermediate fog device.

i D Storage of permanent medical history of patient data to the cloud.

Type ii

ii A Transmission of ECG sensor data to the nearest edge device such as the patient's smartphone or the laptop placed in the patient's room.

ii B Transfer of the task to the next fog device, if the nearest edge device is not capable enough to do the processing.

ii C Verification of authenticity, filtering and comparison of ECG sensor data in the respective intermediate fog device, automatic calling of nearby emergency services and forwarding suitable response to the edge device.

ii D A suitable response forwarded to the edge device that is sent to the actuator to perform the required actuator actions.

ii E Storage of the permanent medical history of patient data to the cloud.

Type iii

iii A Transmission of ECG sensor data to the nearest edge device such as the smartphone of the patient or the laptop placed in the room of the patient.

iii B Transfer of the task to the next fog device, if the nearest edge device is not capable enough to do the processing.

iii C Transfer of the task to the next cloud data centre, if the nearest fog device is not capable enough to do the processing.

iii D Verification of authenticity, filtering, and comparison of ECG sensor data in the cloud data centre, automatic calling of nearby emergency services, and permanent storage of patient history.

iii E A suitable response from the fog device being forwarded to the edge device.

iii F A suitable response forwarded to the edge device that is sent to the actuator to perform the required actuator actions.

In type i fog architecture, authenticity verification, filtering, and comparison of ECG sensor data takes place in the nearest edge device, occurs in the intermediate fog device in type ii, and occurs in the cloud in case of type iii. Type iii would be the same as if the data is entirely processed as in the case of cloud-based architecture. Notifications are sent to the concerned persons in all three types of architecture.

3.4 Proposed Model

The healthcare industry is currently facing substantial challenges due to the ongoing pandemic and the prevalence of chronic diseases. The proposed architectural framework is rooted in insights derived from practical healthcare applications. If effectively implemented, edge/fog computing has the potential to alleviate latency issues in QoS and reduce bandwidth consumption across healthcare applications, with the prospect of extending its utility to other time sensitive scenarios. The primary distinction between edge/fog based and cloud based systems revolve around the computational and storage capabilities of fog devices situated between patients and cloud data centers. Fog technology also addresses the issue of underutilized intermediate devices. Resources like virtualized processing cores, storage, and memory are considered valuable assets at fog nodes. Requests that align with resource requirements, encompassing CPU, memory, and bandwidth, can be accommodated by the current fog or edge device. Otherwise, they may be directed to neighboring devices. The proposed fog based architecture introduces the concept of partitioning fog node virtual machines to efficiently manage data from medical IoT devices. The presence of multiple processes on a single edge/fog node can lead to congestion, impeding task execution. To tackle this challenge, edge/fog nodes can establish virtual machines dedicated to allocating computational resources to specific tasks, each operating as a discrete module. In the realm of medical IoT applications, there is a growing shift towards implementing modular architectures, leveraging the microservices approach. This approach accommodates time-sensitive operations within fog/edge environments and latency-tolerant tasks within the cloud, thus prompting the selection of a microservice architecture for designing and modeling critical real-time medical applications. Further elaboration on the application model will be provided in subsequent sections.

A multitier architectural approach has been chosen in the envisioned integrated fog healthcare application. At tier 0, we have IoT devices like sensors and actuators. These sensors, all operating at the same frequency, capture ECG signals from patients, which are then transmitted to fog nodes through smartphones. The transmission rates of these sensors are regulated in our system using the 'transmitDistribution' attribute within the Sensor class of the iFogSim2 simulator. Specifically, the transmitDistribution has been set using a DeterministicDistribution (EEG_TRANSMISSION_TIME), with a chosen value of 5ms in our proposed system. Actuators are responsible for executing corresponding actions based on the applications' outcomes. The details of the simulator is explained in the following sections.

Moving up the tiers, tier 1 and tier 2 comprise the fog nodes, including proxy servers and gateways. In our fog-based smart healthcare system, this fog layer serves as a crucial intermediary for processing and analyzing real-time critical healthcare data, positioned in close proximity to the end-users. Here, the fog nodes process the data received from the sensor IoT devices, allowing for the patient's health condition to be promptly relayed to their smartphone. Given their strategic placement at the network's edge, patient's experience a fast and a real time response.

The cloud layer, forming tier 3, represent the uppermost level of our proposed system. Here, the permanent healthcare data of patients is stored. In case fog devices face challenges in meeting incoming request requirements, the cloud steps in, providing additional processing power and storage resources. Users connected to the application can access the stored data in the cloud at any time. This multitier architecture for our health monitoring system is illustrated in Figure 3.6.

3.4.1 Problem formulation

Resource allocation, a systematic strategy for distributing available resources to clients, plays a pivotal role in edge/fog computing. Each edge/fog device has a data center with capabilities similar to the cloud, computing elements, and storage capacity. Consequently, the edge/fog computing model faces limitations in its resources to cater to incoming user requests characterized by stringent latency demands. Within the edge/fog network, client applications are capable of running as cloudlets within virtual machines, where cloudlet tasks are executed within their corresponding VMs. By creating multiple virtualized replicas of the underlying hardware, VMs offer vital resources to applications. In such distributed environments, efficient task scheduling becomes paramount as it involves judiciously allo-

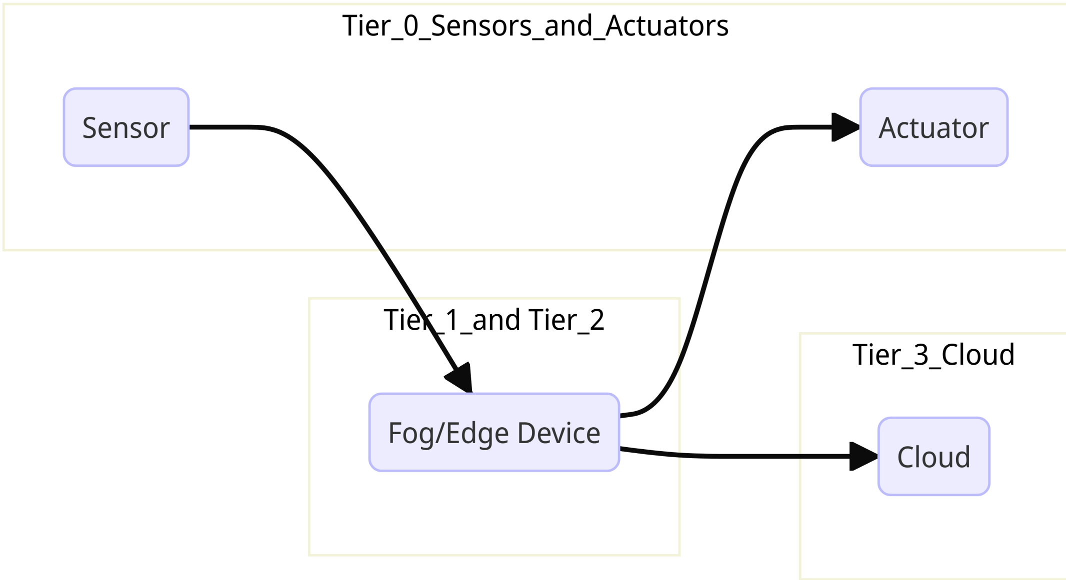


Figure 3.6: Proposed Model-Multitier Architecture

cating resources to tasks. Task scheduling is entrusted to a task scheduler, responsible for mapping an application's tasks to the available resources to meet specific requirements. Given the cost-effectiveness of each VM, task scheduling presents a challenge. To mitigate these challenges, existing literature explores the application of multiobjective optimization approaches, for resource allocation. This study proposes a resource provisioning model for a mobility-aware IoT healthcare application, employing multiobjective optimization for scheduling. The proposed approach seamlessly integrates mobility considerations, clustering strategies, and microservice methodologies into healthcare applications while conducting a comprehensive analysis encompassing energy consumption, network utilization, cost, execution time, and latency parameters. To evaluate this proposed system, we utilize the IFogSim2 simulator [44]. The subsequent subsections delve deeply into the intricacies of the proposed method.

In the envisioned system model, a task scheduler is strategically positioned at the edge/fog nodes, entrusted with the responsibility of scheduling tasks for execution once they are submitted. Consequently, a resource provisioning technique is employed within the nodes to facilitate an efficient match between tasks and available resources.

The primary objective of this research is to achieve efficient resource allocation, taking into account user mobility while minimizing processing time. The applications under consideration in this study are time sensitive, making it less desirable to handle user requests solely through a cloud based approach due to potential latency issues. The involvement of fog/edge devices become imperative

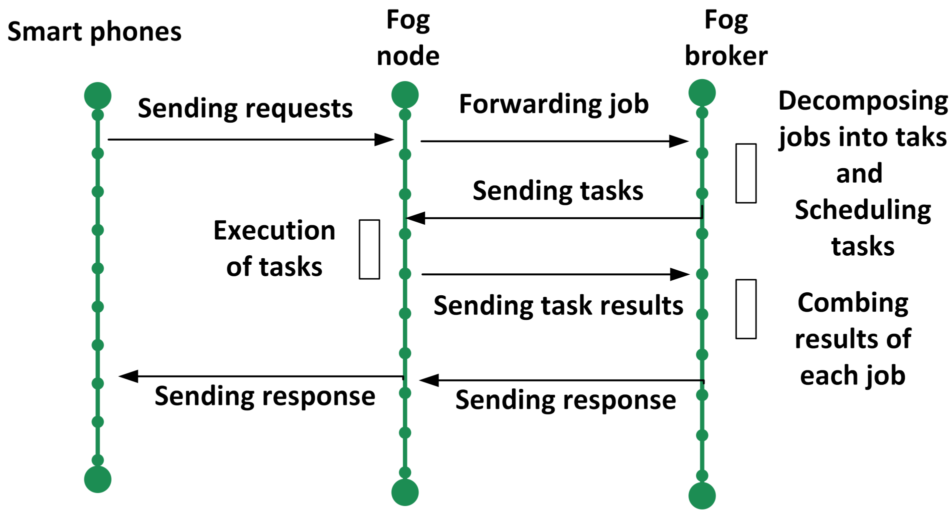


Figure 3.7: Proposed Task scheduling diagram

to ensure timely task completion. The proposed model must address the inherent limitations of fog/edge devices, including their highly distributed nature and limited resource availability. Furthermore, it should accommodate the hybrid fog-cloud environment, accounting for user deadlines and dynamic behavior. Notably, previous research has overlooked resource allocation in fog environments while considering users' changing locations. As previously mentioned, this proposed approach explores the mobility of devices and presents a comprehensive analysis of results under varying conditions. The workflow of this module in the proposed approach is as follows:

- A mobile user initiates a request, which is processed by the linked fog node
- The fog broker receives the job request and disassembles it into a series of tasks for distribution across the distributed system, determining the resource requirements for each task
- The fog broker manages task and node data, employing a scheduling algorithm to optimize resource assignments for each task
- Tasks are then dispatched to the respective fog nodes
- Each fog node is responsible for executing its allocated tasks before sending the completed work back to the fog broker
- Subsequently, the connected fog node transmits the response to the mobile user

Figure 3.7 presents the above steps diagrammatically. An edge/fog computing system comprises a fully interconnected array of "m" servers, with each server housing “n” virtual machines. Virtualization facilitates the creation of multiple virtual machines on a single host or server. It is important to note that all resources are considered uniform in terms of computing capacity and capability. Each host may be assigned various services, and the system’s workload consists of tasks submitted to the scheduler. Tasks are independent scheduling entities that cannot be interrupted and generally represent user compute or service requests. Efficiently provisioning resources in edge/fog systems can be framed as an optimization problem aimed at minimizing execution time. This optimization is particularly crucial for time-critical applications, which exert a significant influence on this parameter, a focal point in our model.

User application requests are decomposed into smaller, independent tasks upon reaching the fog layer, where they undergo processing within the cloud-fog computing infrastructure. To effectively harness the advantages of fog computing, the approach to application development has shifted from monolithic design to microservice architecture. Our system’s application model comprises numerous microservices, underpinned by the microservices methodology. This methodology enables the construction of applications from multiple small services, each operating in its dedicated process and using straightforward protocols for communication. Microservices empower the development of systems composed of numerous self-contained components, each capable of managing its data. The adoption of microservices yields benefits like heterogeneity, robustness, scalability, ease of deployment, organizational alignment, and composability, thus facilitating the creation of large-scale IoT applications. In the modern landscape, microservice deployments hold greater significance due to their high performance and suitability for IoT applications. Figure 3.8 provides a visual representation of the proposed system’s application model, consisting of an array of microservices. Each microservice is denoted as a vertex, while the edges illustrate the data connections between them. Within this design, three distinct microservices are identified.

- Client Microservice: Serving as the forefront of the healthcare system based on fog computing, the client microservice operates on users’ smartphones. Its primary function is to receive sensor ECG signals linked to the patient. Subsequently, it forwards this sensor data to the preprocessing microservice, hosted on the fog device.

- Preprocessing Microservice: The preprocessing microservice is responsible for conducting data validation and cleansing operations on the ECG sensor data. This crucial step aims to reduce noise in the data before it is transmitted for further processing.

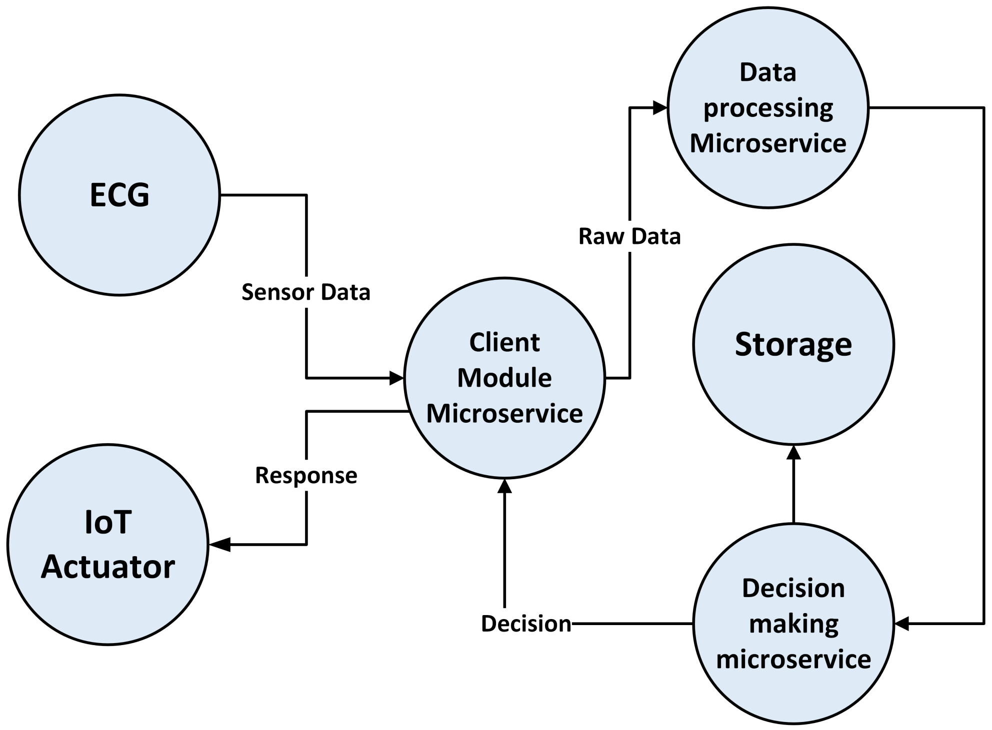


Figure 3.8: Data flow diagram with microservice modules

- Decision Making Microservice: To alert the client microservice regarding the patient's health status, the decision-making microservice plays a pivotal role. It must assess the real-time data to determine whether an emergency situation exists. This critical decision-making process is executed within this microservice.

For the purpose of user health monitoring, these microservices engage in intercommunication. The deployment of time-critical microservices, responsible for preprocessing and decision-making, can vary based on the placement policy, with options including deployment in either the edge/fog or the cloud. The storage module is responsible for handling data that is intended for permanent storage in the cloud. This research introduces a resource provisioning mechanism of notable utility, capitalizing on a multi-level hierarchical fog architecture. Within this architecture, multiple levels of application placement requests are processed at the edge/fog nodes. It employs a decentralized placement approach to distribute microservices across the edge/fog environment, creating a model for a critical IoT medical application.

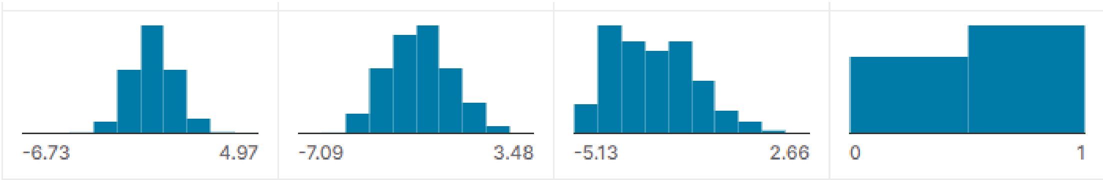


Figure 3.9: Dataset description

3.4.2 Datasets

Datasets - Mobility

In this study, the EUA dataset [178] is employed, offering precise positioning data for edge/fog nodes strategically deployed within the Central Business District areas of major Australian cities, including Melbourne and Sydney. The dataset is partitioned into distinct regions and further subdivided into blocks. Within each block, one node is randomly selected as the proxy server to ensure unbiased representation. In the same block, all nodes, except the proxy server, function as gateways for IoT devices. This repository houses a compilation of EUA datasets sourced from real-world data origins, generously shared with the public to support ongoing research in edge computing. These datasets specifically pertain to the Australian region, enhancing the fidelity of real time environment simulations. A sample representation of data set is presented in Appendix A.1.

ECG dataset

The dataset [179] considered for this work has 140 columns representing the ECG readings and a label encoded as 0 or 1, denoting whether the ECG is abnormal or normal. Columns 0-139 contain the ECG data point for a particular patient. These are floating point numbers. The first three column value ranges along with the final label description, are presented graphically in Figure 3.9 to understand the spread of values in the dataset. The first column of Figure 3.9 displays the maximum and minimum values for the first data point of the patient set in the dataset, while columns 2 and 3 represent the same information for the second and third data points respectively. The final illustration in the figure provide a label indicating whether the ECG is classified as normal or abnormal. The dataset has 58% of the tuples belonging to the normal class and the remaining belonging to the abnormal class. A sample of the dataset is shown in Appendix A.2.

3.5 Evaluation parameters for proposed approach

The proposed work has been evaluated for identified metrics which is detailed in this section. A set of n independent tasks are delivered to the system at each time, assuming that T_k represents the k^{th} task denoted as follows:

$$T = \{T_1, T_2, T_3, ..., T_n\} \tag{3.1}$$

The assumed infrastructure comprises edge/fog/cloud nodes, which are processors with characteristics such as CPU rate, CPU usage cost, bandwidth usage cost, and memory usage cost. The set of m processors is made up of fog nodes as mentioned below:

$$N = \{N_1, N_2, N_3, ..., N_m\} \tag{3.2}$$

where N_i represent the i^{th} processing node. The processor N_i allocated with job T_k is denoted by T_k^i.

A set of one or more tasks may be assigned to one processor for computing:

$$N_iTasks = \{T_x^i, T_y^i, ..., T_z^i\} \tag{3.3}$$

The subsequent information discuss the performance metrics employed to assess the implementation of the proposed approach across the edge, fog and cloud layers.

Execution time The execution time (EXT) required by node N_i to finish a set of N_i*Tasks* assigned to it is:

$$EXT(N_i) = \sum_{T_k^i \in N_iTasks} EXT(T_k^i) \tag{3.4}$$

$$EXT(T_k^i) = \frac{length(T_k^i)}{CPU_i} \tag{3.5}$$

where length(T_k^i) denote the number of instructions in the task T_k. The node N_i's CPU rate is represented by CPU_i and depends on factors such as clock rate, core count, instruction level parallelism, etc. Total execution time is the total time taken by the system to complete all the tasks, defined from the time when the request is received until the last task, or the time when the last machine completes. Total execution time is determined by the formula:

$$TXT = \sum_{m}[EXT(N_i)] \tag{3.6}$$

The time used to complete the job while utilizing system services is included in the task's execution time. Execution times differ amongst tasks because they rely on how intensive the processing and input-output activities are.

Latency The application's control loop is Client Microservice $\rightarrow$ Preprocessing Microservice $\rightarrow$ Decision Making Microservice $\rightarrow$ Client Microservice. The control loop in our proposed approach has a relatively shorter latency, indicating better coordination and placement of computational resources. The iFogSim2 simulator provide multiple methods for module placement, including the one called edge-ward placement which has been utilized in this work. This method involves shifting microservices towards the top of the fog hierarchy, so that it leads to a deployment of a singular instance of each microservice along the route from the edge to the cloud. Due to the restricted resource capacity of fog nodes, this strategy places microservice instances in higher fog layers, increasing average latency.

$$TL = \sum_{m} CAL \tag{3.7}$$

Total latency (TL) represented in Equation 3.7 directly depends on the allocation of VMs in fog devices in which the tasks are distributed for execution. Total latency is the summation of the current average delay (CAL) experienced by every VM inside the host for m servers, where CAL is calculated as follows.

$$CAL = CC - ET \tag{3.8}$$

Here, CC denote the simulator clock and ET is the execution time of the tuple. CC, the simulator time is recorded by the simulator when the response is received by the IoT device and ET, when the request is received by the edge/fog node. The difference between these times give the total latency experienced by the tuple, which include all the delays in the communication path, such as transmission delay, propagation delay, processing delay, and queuing delay.

Energy consumption Energy consumption is one of the significant parameters that we have in computing systems. This typically includes infrastructure for data communications for backup power supplies, environmental controls, including cooling systems, fire control, and different security technologies. Infrastructure operational costs are impacted by the power supply. Hence methods must be put in place to lower these costs. The edge-ward method places the majority of microservices on cloud VMs, which increases the energy consumption of cloud resources. The amount of energy used depends on how many microservices are active on each tier. Efficiency, affordability, availability, dependability, and environmental protection of devices are all greatly impacted by energy consumption reduction.

We model the server or host's energy consumption as the sum of two components: the fixed energy for the server in an idle state and the variable energy for server utilization while processing the requests. Energy consumption depends on the server's number of VMs and the allocated MIPS allocated for each VM.

The variable energy for server utilization while processing the requests is EN_{i_k}. E is the total energy consumption which can be calculated by

$$\sum_{i=0}^{m}(\sum_{k=1}^{n} EN_{i_k} + E_0) \tag{3.9}$$

$$EN_{i_k} = e_1 * EXT(T_k^i) \tag{3.10}$$

where EN_{i_k} is the energy consumption by the task T_k running on the virtual machine or node i. Operating the data centre requires E_0, the fixed energy of the server in idle state, and e_1, the energy consumption per unit time in node N_i. The suggested method makes some fixed assumptions regarding the simulation setup, including the distance between fog nodes, energy efficiency, and power consumption of communication devices. However, we are aware that these factors typically affect the amount of energy used for communication, therefore we will address this in the future to improve our model.

Network usage A crucial metric for comparing various approaches is the overall volume of data delivered across a network. Particularly on large networks, high data transfer may result in network congestion, service interruptions, or an increase in the control loop's average delay for the applications. In comparison to cloud operations, the latency can be significantly reduced if the edge of the network can manage the portion of the workload. Additionally, the edge-to-cloud traffic is to be maintained. Data transmission size could be considerably decreased by data pre-processing at the edge and fog devices. However, bandwidth conservation is essential because many endpoints connect to the network and many database servers are needed to run them. Network usage depends on the latency experienced by the network and the tuple size of the data for $'n'$ VMs in the host as listed in Equation 3.11.

$$NU = \sum_{n}(l * TNS) \tag{3.11}$$

where l denote the latency experienced by the network and TNS denote the tuple network size. Tuple network size refers to the number of tuples that can be processed simultaneously within the network. Total network use depends on the number of VMs in fog devices in which the tasks are distributed for execution.

Cost Costs include network hardware, infrastructure, network communications, processing, and storage costs. The investment made by service providers in edge/fog computing also include the placement of processing and communication workloads in the edge/fog device. One of the main issues with edge/fog computing is cost saving.

Processing cost is defined as:

$$cost(T_k^i) = c_1 * EXT(T_k^i) + c_2 * M(T_k^i) + c_3 * Bw(T_k^i) \tag{3.12}$$

where c_1 denote the CPU usage fee per time unit in node N_i, and EXT(T_k^i) is given in Equation 3.5. c_2 denote the memory usage fee per data unit in node N_i and M(T_k^i) represent the memory needed by task T_k. Task T_k processed in node N_i needs an amount of bandwidth Bw(T_k^i), which is the sum of input and output file size. c_3 is the bandwidth usage fee per data unit. The following formula is used to determine the cost of each task in the Edge-Fog-Cloud system in total.

$$Totalcost = \sum_{T_k^i \in N_i Tasks} cost(T_k^i) \tag{3.13}$$

Table 3.1 contains the list of acronyms used in this section.

3.6 Implementation and Results - Weighted sum method

3.6.1 Experimental setup

The simulator iFogSim2 makes it possible to simulate fog computing environment to analyze resource management policies and scheduling for IoT applications. It is an open-source toolkit used to simulate fog, edge and IoT applications. iFogSim2 works in conjunction with CloudSim, a commonly used cloud-based environment simulation and resource management tool. Fog device, Sensor, Actuator, Tuple, Application, Monitoringedge and Resource management system are few of the classes of iFogSim2 that are needed to simulate the fog network. The following section describes the parameter setup and metrics of optimization considered for simulation.

Table 3.1: Notations used in evaluation metrics

Symbol	Meaning
T	Tasks
E_0	Power required for the server in an idle state
N	Nodes
NU	Network use
m	The Number of servers
l	Latency experienced by the network
n	Number of VMs inside the host
TNS	Tuple network size
$EXT(N_i)$	The execution time required by node N_i
$cost(T_k^i)$	Cost for processing task T_k^i
TXT	Total execution time
$M(T_k^i)$	Memory needed by task T_k^i
TL	Total latency
$Bw(T_k^i)$	Bandwidth needed by task T_k^i
CAL	Current average latency
CC	Simulator clock
ET	Execution time of the tuple
E	Total energy consumption
EN_{ik}	Energy consumption by the task T_k

3.6.2 Parameter set up for simulation

Simulation requires creating cloud data centre, fog devices, sensors and actuators in iFogSim2. To do so, one can make use of iFogSim2 physical topology classes such as fogdevice, sensor and actuator and fundamental classes such as controller, module mapping, module placement and application. The placement policy used is edge-ward placement. The edge/fog devices are hierarchically linked. edge/fog devices at low level are connected directly to the relevant sensors and actuators. edge/fog device is generated with different instruction processing rate and parameters of energy consumption such as busy and idle power indicating its capability and energy efficiency. For defining createFogDevice() function, the standard values are used for the parameters like mips, ram, upBw, downBw, ratePerMips, busyPower and idlePower. Creating sensor tuples with specific deterministic distribution using random function and generation of sensor tuples are event driven. The settings are as follows for the five-layer approach, cloud in level 4, proxy and gateway in level 3 and 2, mobile in level 1 and sensors/actuators in level 0. Similarly for the four-layer approach, the settings are, cloud in level 3, gateway in level 2, mobile in level 1 and sensors/actuators in level 0. Cost per million instruction handling is pre-set for the physical component configuration at simulation time. Once the simulation is completed, the controller object collects cost (dollars), network usage (kilobytes), and energy consumption (megajoules) results from fog devices. It uses the hands-off mechanism for edge and fog node mobility which is the default set-up in iFogSim2. Execution and analysis of the performance parameters considered in this proposed work are cloud energy, router energy, network usage, cost and delay which are described in the following section. This implementation aims to illustrate the effectiveness of fog based application over cloud based systems.

3.6.3 Objective function for weighted sum method

The parameters considered for optimization are energy consumption, network use, execution time, cost and latency represented by the weight vectors w_1, w_2, w_3, w_4, w_5 respectively. They hold the values between 0 and 1 and sum to 1. These weights are subjective and define the contribution of each parameter in the solution space. The multi criteria decision making methods like analytical hierarchical process could be used to derive the weight vector. Any point of a convex pareto front can be obtained by altering the weights. The model is solved for each combination of weighted coefficients, and the objective function values are saved in the pareto set. The objective functions considered in this work are presented in Equations 3.14, 3.15, 3.16, 3.17, and 3.18. The multi-objective optimization problem is given in Equation 3.19.

$$obj_1 = min(TXT) \tag{3.14}$$

$$obj_2 = min(TL) \tag{3.15}$$

$$obj_3 = min(E) \tag{3.16}$$

$$obj_4 = min(NU) \tag{3.17}$$

$$obj_5 = min(Totalcost) \tag{3.18}$$

$$min(w_1 * obj_1 + w_2 * obj_2 + w_3 * obj_3 + w_4 * obj_4 + w_5 * obj_5) \tag{3.19}$$

where

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1 \tag{3.20}$$

The constraints for this problem ensure that the total CPU requirements of all jobs assigned to a host do not surpass the host's capacity. Additionally, each virtual machine should be allocated to only one host. The start time for the simulation should be greater than zero, and the tuple network size must not exceed the network size limit.

Table 3.2: Weight Selection for Weighted Sum Method

w_1	w_2	w_3	w_4	w_5	Objective function value
0.165	0.143	0.143	0.221	0.329	2.21E+06
0.18	0.12	0.14	0.21	0.32	2.40E+06
0.18	0.06	0.06	0.1	0.58	2.40E+06
0.18	0.05	0.08	0.08	0.5	2.40E+06
0.2	0.2	0.2	0.2	0.2	2.67E+06

Analytical Hierarchical Process (AHP) is a structured technique that is used to solve multi-criteria decision making problems in many areas including e-commerce, transportation, portfolio selection, supplier selection etc. It helps in organizing and analyzing complex decisions, based on mathematics and psychology, where alternatives are ranked using the pairwise comparison of multiple criteria [180]. The weighted sum for each alternative is computed by multiplying each objective value with its corresponding weight and summing up the result, ending in a singular value representing the performance of each alternative. These alternatives are then ranked in descending order based on their weighted sum values, creating a list from best to worst. Beginning with the top alternative, non-dominated solutions are identified by comparing their weighted sum value to those of the alternatives below it. If an alternative has a better or equal weighted sum value for at least one objective and a better-weighted sum value for at least one other objective, it is considered non-dominated. This process is repeated for all alternatives, and the preferred solution is selected from the non-dominated set. However, this weight assignment is subjective and can vary from problem to problem. The use of AHP in this proposed study leads to the set of weight vectors listed in the Table 3.2 for the parameters considered.

3.6.4 Use cases considered

The proposed work compares the performance of the cloud and the fog architectures by varying the number of gateways and the number of devices per gateway deployed at the varying levels in the implementation. These are described as various case scenarios in this section and they represent the different types of fog architectures described in section 3.3.2 in order.

Case Scenario i

Scenario i is implemented with a single gateway deployed with a single device. This is equivalent to the real-time scenario where patient information is transmitted to a single mobile phone (patient's phone) and mobile data is transmitted to the cloud through a single network provider gateway. The simulation results infer that the fog-based architecture has a steep reduction in the cost and the network use while the router energy consumed is at an ignorable level. This basic scenario motivates the concept of using the fog nodes for critical computations than sending them to the cloud directly. Implementing the described fog-based use case scenario in real-life healthcare can enhance patient well-being by providing swift and reliable access to critical medical information on their mobile devices. This approach ensures timely communication with healthcare providers, fostering more informed decision-making and improving overall healthcare quality. These observations can be seen in the results listed in Table 3.3.

Case Scenario ii

Scenario ii is implemented assuming the use case where patients sensor data is transferred to four mobile phones (patients phone, caretakers phones etc) and mobile data is transmitted to the cloud through a single network provider gateway. Thus, this implementation in iFogSim2 uses a single gateway each deployed with four devices. The simulation results thus obtained are presented in Table 3.4. The inference drawn from this table is as follows:

- The cost of the computation and the network usage in fog based implementation is approximately reduced by 96% and 93% respectively when compared to the cloud model.
- Meanwhile, the router energy of the fog model increases by 1.5% when compared to the cloud model. This is due to the fact that the computations are performed in the fog nodes and this increase at the fog level will reduce the overload on the cloud.

Since the proposed architecture describes a time-critical medical application, it has to be fault tolerant (i.e.) even in case of failure of a device, the system should continue functioning and provide correct results with minimal delay. In order to make the proposed architecture fault tolerant, scenario ii has been tested with four and five layers as can be seen in Figure 3.10, the additional layer is the use of proxy server which replaces the main gateway in case of failure. The intention of this test case is to analyze the parameters with the additional level and check if the fog architecture still shows up a better or an acceptable performance when compared

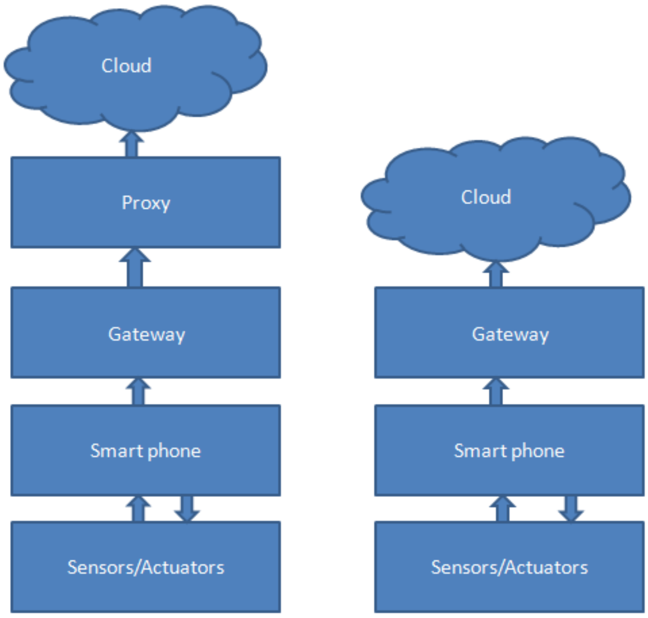


Figure 3.10: Two Approaches

to the standard cloud model. Table 3.5 presents the results of both four and five-layer model. The default implementation of case scenario ii is the layer 4 approach. When considering the presence of the proxy server, it could be noticed that the cost of the computation increases than the default model which is as expected, but this increase does not compete with the increase attained from the cloud architecture. To conclude, the network use and the cost seems to be approximately doubled in case of fog architecture. Even if all the intermediate devices are replicated to remain fault tolerant, edge/fog based approach yields a better performance compared to the cloud model. This use case scenario facilitates collaborative care by providing health professionals with comprehensive and timely information, fostering improved decision-making and personalized healthcare interventions for better patient outcomes.

Table 3.3: Case Scenario i

G1D1	**Cloud energy**	**Router energy**	**Cost**	**Network use**
Fog Based	1.33E+07	895,309.6105	7,504.486	2,821.18
Cloud Based	1.35E+07	834,333	244,466.8	41,661.72

Table 3.4: Case Scenario ii

G1D4	**Cloud energy**	**Router energy**	**Cost**	**Network use**
Fog Based	1.33E+07	1,048,835.431	26,169.34	11,223.52
Cloud Based	1.38E+07	834,333	721,848.9	166,344

Table 3.5: Comparison of layer 5 and layer 4 approach

Approach	Fog/Cloud	Network Use	Cost
Layer 5	Fog	22,834	47,788
Layer 5	Cloud	10,577,241	1,118,855
Layer 4	Fog	11,223.52	26,169.34
Layer 4	Cloud	166,344	721,848

Table 3.6: Case Scenario iii

G4D4	Cloud energy	Router energy	Cost	Network use
Fog Based	1.34E+07	1,048,835.431	86,975.14	46,036.88
Cloud Based	1.51E+07	834,333	2,587,193	664,999

Table 3.7: Proposed work parameter selection comparison with literature

Reference	Latency	Network Use	Cost	Cloud Energy	Router Energy	ET
[181]	✔					
[71]	✔	✔		✔		
[33]	✔		✔	✔		
[32]	✔	✔		✔		
Proposed	✔	✔	✔	✔	✔	✔

Table 3.8: Parameter Improvements results data from the Literature Survey Papers

Reference	Latency	Network Use	Cost	Cloud Energy	Router Energy	ET
[181]	73%	-	-	-	-	-
[71]	-	65%	-	-	-	-
[33]	46%	-	43%	23%	-	-
[32]	19.56%-29.45%	22.61%-26.78%	-	23.56%	-	-

Case Scenario iii

Scenario iii is implemented assuming the case where the patient sensor data is transferred to more end-to-end mobile phones (patient phones, caregivers, emergency services, hospitals, doctors) and mobile data is transmitted to the cloud through multiple network provider gateways. This scenario uses four gateways each deployed with four devices in the simulation process. In practical terms, the implementation of Scenario iii, where patient sensor data is shared with a broader

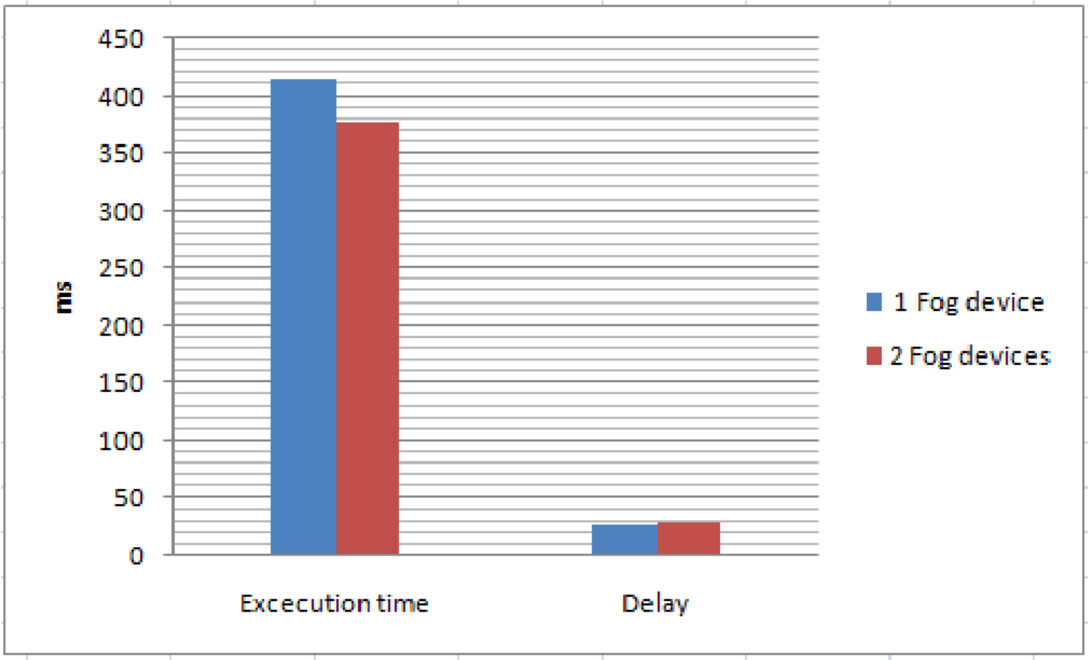


Figure 3.11: Distributed architecture results for the Proposed Architecture

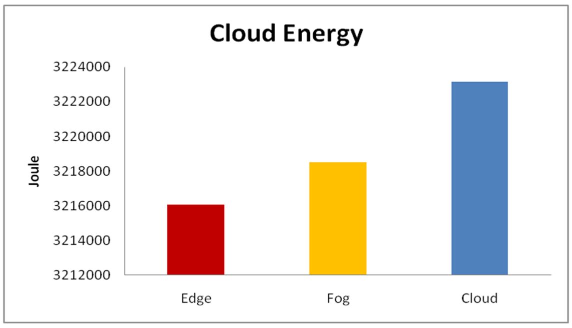
network of end-to-end mobile phones, offers substantial benefits for patients, caretakers, and health professionals. This enhanced connectivity enables more immediate access to vital health information, fostering improved collaboration among healthcare stakeholders, quicker emergency response times, and enhanced patient care management. The simulation results listed in Table 3.6 indicates that the observations are similar as discussed in the previous two cases.

The observations drawn from the above results are:

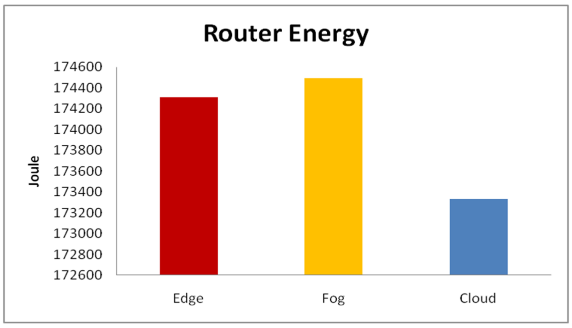
- In case of fog-based architecture, the fog nodes act at intermediate levels and perform the computation utilizing the router energy completely and transmit the end result to the cloud for storage, while in the case of cloud-based architecture, the fog nodes though present, just transmit the data to the cloud system where the computations occur. Thus, the router energy of the fog-based model is seen to be high in all the case scenarios

- Fog nodes though compute the results at the intermediate level, it uses the cloud system for the final storage. Hence, it makes use of a nominal cloud energy at every stage. This can be seen by comparing the cloud energy usage parameters in the results section

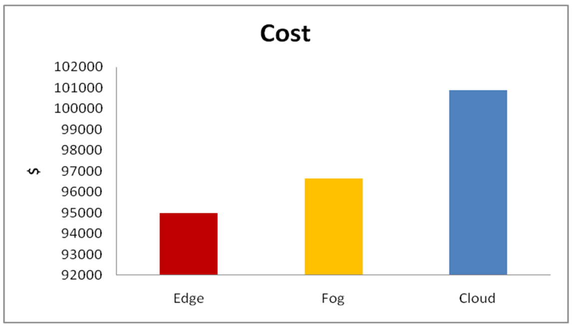
- As discussed, the network usage and the cost parameters turn out to be beneficial for the users in the fog-based architecture. This in turn makes the proposed system more efficient when compared to the existing cloud-based implementation



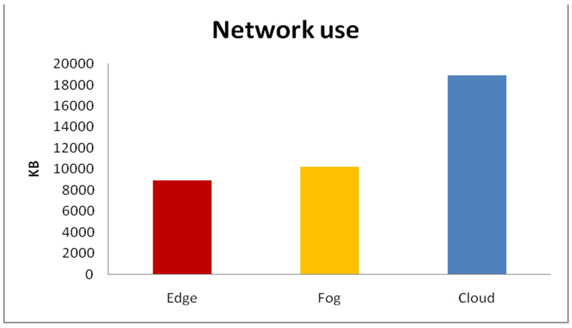
(a) Cloud Energy



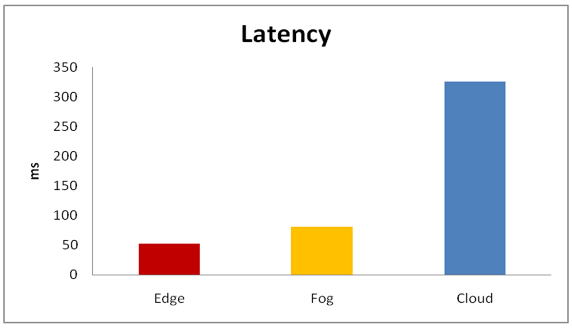
(b) Router Energy



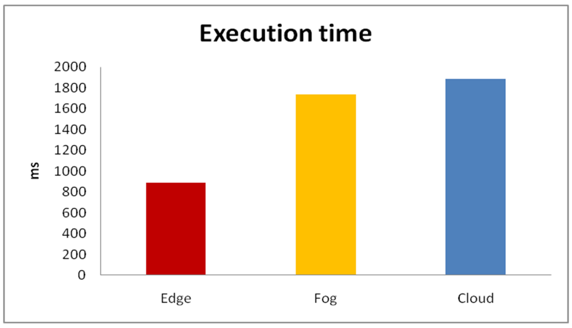
(c) Cost



(d) Network use



(e) Latency



(f) Execution time

Figure 3.12: Results of Edge, Fog and Cloud Deployment for the proposed model

In addition to the above case scenarios, the proposed system has been tested for its behavior in a distributed environment (i.e.) at intermediate level, the data transmission which generally happens with a single fog node was replaced with two fog nodes which transmits the data to the next level. This enables load sharing where each overloaded node can share its workload with the adjacent less loaded node in the transmission process. The expectation here is that the execution time and the delay parameters should showcase an improved performance. The results presented in Figure 3.11 concludes that the addition of two fog nodes to do the stipulated task reasonably reduces the execution time with a slight increase in the delay parameter. The increase in the delay is due to the fact that the load sharing involves its own initial overhead in distributing the task. Thus, the proposed fog-based architecture can support load sharing too in addition to the various other advantages like network use and cost. Hence, it can be concluded that the proposed model can be preferred over the existing cloud based methods for time-critical applications. It has undergone testing across cloud, fog, and edge scenarios, with the outcomes illustrated in Figure 3.12.

Tables 3.7 and 3.8 presents the consolidated list of parameters addressed by the existing literature along with the performance improvement in the simulator used. Among the existing fog architectures proposed approach improves network use by 31% compared to [71] and 71% compared to [32]. Cost improvement of 53% compared to [33], Latency improvement of 49% compared to [33], 22% compared to [181] and 75% compared to [32]. With this aggregation of results, it could be concluded that the proposed model outperforms the existing fog based architectures additionally supporting fault tolerance and load sharing and hence would be a preferred model for the time critical health care application. Heterogeneity, privacy, reliability and scalabilty could be open issues for future work [182].

3.6.5 Limitations of weighted sum method

However, it is important to acknowledge that the weighted sum method has certain limitations. One notable challenge is the potential for subjectivity in assigning weights, which can introduce bias and influence the final outcome. Additionally, the method assumes that the relationships between criteria are linear and additive, which may not always hold true in complex decision landscapes. While the weighted sum method offer simplicity and ease of implementation, it comes with certain limitations that can impact its effectiveness, particularly in complex optimization scenarios like energy-latency trade-offs in edge/fog and cloud computing.

One significant limitation is its assumption of linear relationships between objectives. The weighted sum method assumes that changes in one objective have a consistent and linear impact on the overall score. However, in real-world scenarios, the relationships between objectives might be nonlinear, intricate, or even contradictory. For instance, increasing energy efficiency might not always result in a proportional decrease in latency. Consequently, the weighted sum method might fail to accurately capture the nuanced trade-offs between energy consumption and latency, leading to suboptimal solutions.

Furthermore, the subjectivity involved in assigning weights to objectives can introduce bias and uncertainty. Weight assignment requires decision makers to have a clear understanding of the relative importance of objectives, which can be challenging, especially when dealing with complex, interdependent factors. Incorrect weight assignments can lead to skewed optimization results, as the method heavily relies on the accuracy of these weights. In scenarios where preferences change or where there is no clear consensus on the importance of objectives, the weighted sum method's reliability diminishes.

3.6.6 Need of Metaheuristic Methods for Parameter Optimization

To address the limitations of the weighted sum method and tackle the complexities of parameter optimization in energy-latency trade-offs for edge and cloud computing, researchers are increasingly turning to metaheuristic methods. Metaheuristic methods, such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization, offer several advantages that make them well-suited for these optimization tasks.

Metaheuristic methods can handle complex and nonlinear relationships between objectives without relying on explicit mathematical models. These methods explore the solution space more comprehensively by using adaptive search strategies that can adapt to the problem landscape. They can capture intricate trade-offs and discover optimal solutions in situations where linear methods like the weighted sum fall short.

Moreover, metaheuristic methods are versatile and customizable, allowing researchers to tailor them to specific problem characteristics. These methods can be applied to various optimization scenarios, and their parameters can be adjusted to balance exploration and exploitation efficiently. Additionally, metaheuristic algorithms often incorporate mechanisms to escape local optima and explore diverse regions of the solution space, thus increasing the likelihood of finding globally optimal solutions.

In conclusion, metaheuristic methods offer a robust alternative to the weighted sum method by addressing nonlinear relationships, uncertainty, and subjective weight assignments. Their adaptability, ability to explore complex landscapes, and potential for finding high-quality solutions make them a promising choice for parameter optimization tasks, especially in the intricate domain of energy-latency trade-offs in edge/fog and cloud computing environments.

3.7 Summary

Prompt response is very significant for medical applications. However, the reduction in response to the application is difficult due to several factors. The invention of IoT that can transfer data without human intervention has come to the rescue of patients in need of urgent care. As a solution to one of the major problems in the healthcare industry, this work proposes a edge/fog based architecture that can transmit data at a faster pace than the current cloud-based implementation. The proposed model uses a simulator to experiment with the use cases of ECG signal and tests for its performance. The results show that the model outperforms the existing literature in many of the parameters compared. In addition, it also enables load sharing and fault tolerance. The overall performance of the proposed work is better, as seen in the explanations earlier in this thread. Based on our comprehensive literature review, it has become evident that enhancing the efficiency of the proposed application can be achieved through the utilization of metaheuristic techniques for resource provisioning. Therefore, we propose the incorporation of a metaheuristic approach in resource provisioning which is explained in the following chapter.

Chapter 4

Resource Provisioning based on Metaheuristic Methods

4.1 Introduction

This section describes the proposed solution to the resource provisioning problem using metaheuristic multiobjective optimization methods. The approach addresses the resource provisioning problem as a multiobjective optimization problem with the objective of minimizing the evaluation parameters considered in this work. Resource provisioning and scheduling is a significant problem due to heterogeneity, mobility, and dispersion of edge/fog/cloud resources. Scheduling aims to match tasks with the right resources, which are included within the scope of NP-hard problems, and it takes a long time to discover the optimal solution.

Metaheuristic-based techniques have been proven to achieve near-optimal solutions within a reasonable time for such problems [89]. Finding global optimum solutions to several complicated multi-modal design problems in engineering and industry seems to be very difficult. In such scenarios, conventional optimization techniques perform inadequately because they may become locked in local optima. The utilization of metaheuristic algorithms derived from nature is hence proposed. Due to their ability to avoid stagnation in local optima and high convergence speed in the right direction of the near-optimal solution, meta-heuristic optimization algorithms have greatly impacted many fields in recent decades. These algorithms tackle many optimization problems, especially problems in the engineering domain. Metaheuristic algorithms function as optimization models for solving various optimization problems in a reasonable amount of time due to their significant results. By making a few assumptions about the optimization problems, metaheuristics offer a set of solutions that can be applied to various issues. These algorithms use less processing to find feasible results.

According to the source of their inspiration, metaheuristic algorithms have been divided into four groups: human-based, swarm-based, physics-based, and evolutionary algorithms. Among these evolution-based algorithms, imitate biological evolution by using reproduction, mutation, recombination, and selection to create new offspring that are more powerful than their parents. The majority of evolutionary algorithms, such as genetic algorithms, evolution strategy, genetic programming, biogeography-based optimizer, and probability-based incremental learning, have been extensively used for different optimization issues. The swarm-based or social behavior-based algorithms include the harris hawks algorithm, particle swarm optimization, cuckoo search, whale optimization algorithm, slime mold algorithm, marine predators algorithm, grey wolf optimizer, ant colony optimization, bat algorithm, and flower pollination algorithm.

The best way to solve task scheduling issues in fog computing is to use metaheuristic algorithms that can quickly handle a large search area and find the best solution. Hence we have selected metaheuristic approaches for the proposed method.

4.2 Proposed Model

A literature survey pertinent to this proposed model is discussed in Section 2.5 which review existing methodologies and frameworks of metaheuristic methods in edge/fog/cloud IoT applications.This section explains the solution to the resource provisioning problem using the metaheuristic methods. This work uses the modified version of the genetic algorithm and the flower pollination algorithm which is described below.

4.2.1 Fitness function of the proposed Metaheuristic methods

A fitness function serves as the guiding compass for metaheuristic methods, encapsulating the problem's core objectives into a quantitative measure. By evaluating potential solutions and assigning numerical values, the fitness function allows metaheuristic algorithms to navigate the solution space effectively. Crafting an adept fitness function necessitates domain understanding, streamlined computation, and consideration of scale, smoothness, and multiobjective aspects. Its pivotal role in influencing the optimization process demands careful design, validation, and potential adaptation to dynamic scenarios, ensuring that the algorithm converges towards optimal or near-optimal solutions efficiently.

Fitness function Execution time is the the most important factors influencing energy consumption, network use, cost, and latency. Thus, an objective function employed for evaluating the candidate solutions can be represented as follows:

$$Fitness = \sum_{m} [EXT(N_i)] \tag{4.1}$$

$$EXT(N_i) = \sum_{T_k^i \in N_i Tasks} EXT(T_k^i) \tag{4.2}$$

where length(T_k^i) denote the number of instructions in the task T_k and the node N_i's CPU rate is represented by CPU_i.

Algorithm 1 Modified Genetic Algorithm

```
Input: An application's set of resources available and unmapped tasks.
Output: Output mapping
Construct a list of the available resources.
Create population
for Each chromosome do
    Determine the optimum resources (best fit) for every activity based on the
    time it will take to complete it.
    Go to the next resource in the list.
    if The counter of index = last resource then
        Go to the first resource on the list.
    end if
end for
Evaluate all chromosomes using the fitness function
while Termination condition not reached do
    Random selection and crossover
    Mutation
    select best chromosomes
end while
Save the best solution
Map the tasks on resources
```

4.2.2 Modified Genetic algorithm

In the Modified Genetic Algorithm, the initial population is generated using the best-fit strategy instead of the traditional random method. This modification ensures that the initial population comprises individuals with favorable traits, thereby enhancing the algorithm's efficiency and effectiveness. By commencing with a population demonstrating promising characteristics, MGA expedites convergence towards optimal solutions, thus reducing computational burden and enhancing performance. This strategic initialization not only accelerates the evo-

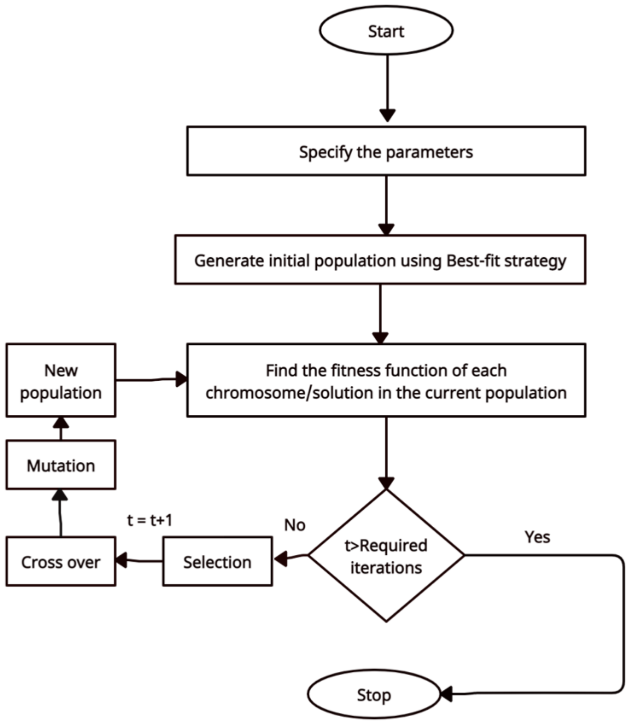


Figure 4.1: Modified genetic algorithm flowchart for the Proposed Model

lutionary process but also increases the likelihood of discovering superior solutions within fewer generations. Leveraging the best-fit strategy for population generation in genetic algorithms significantly augments their ability to produce high-quality outputs efficiently. By taking into consideration the fog/cloud system features, a modified meta-heuristic method based on GA is proposed in this approach.

- Initial population: In a standard genetic algorithm, it is generated at random. In edge/fog time critical applications, creating a good and goal-oriented initial population that leads to finding the response promptly is preferable. Hence to build the initial population, the tasks/cloudlets are sorted by execution times and their processing capacity in MIPS. The chromosome's first row of genes is occupied with sorted cloudlets. The best-fit technique is to select a suitable VM with the shortest operating time for each task from the virtual resource list.

- Crossover: A random gene selection is used in this case. Two parents are selected randomly, and their genes are chosen at random. Then, two other solutions are developed by altering resource regions of selected genes.

- Mutation: A chromosome and one of its genes along with a resource from a virtual list is chosen at random. If the execution time is faster than the last candidate, the selected resource will be replaced with the selected gene. The mutation leads to the speedy discovery of a good solution.

- Evaluation and selection solutions: It uses a fitness function with efficient parameters to recognize the value of a solution. The fitness function is applied to all the solutions, and their values are determined in GA. Then the solution with the best value is identified as the maximum or minimum for the fittest solution using parameter placement guidelines.

The proposed pseudo-code is presented as Algorithm 1. Figure 4.1 presents the flowchart for the same. In the proposed approach, solutions with minimum execution time can be considered the best value after fitness function estimation. The number of virtual resources in the list is denoted as $'m'$. Each solution's fitness value is calculated using Equation 4.1.

The chromosome with minimum fitness value is considered as the best solution among others. The target is to minimize the fitness function. The method tries to find a solution by using crossover and mutation operations to reduce the fitness value as much as feasible. Single-point crossover has been applied in the proposed approach due to its advantages, which include enhanced genetic diversity, expedited convergence, effective exploration of the search space, and retention of favorable genetic material. By utilizing single-point crossover, the diversity of solutions can be increased, and the search for optimal solutions can be accelerated. The population selection strategy opted is the elite selection, which is a simple method to implement. It only involves selecting a fixed number of the most best individuals from the population. Elite selection helps quicken the convergence, enhance the genetic diversity, and keep the good genetic material by preserving the best individuals in the population. It prefers some of the best chromosomes for the next iteration, considered elites. The time complexity of GA for the resource allocation problem considered in this work depends on the number of generations (g), the population size (n), and the complexity of the fitness function (m) represented as O(g * n * m). The memory complexity of the GA implementation depends on the representation of the chromosome, the population size, and the size of the problem. The memory required to store the population is proportional to the number of chromosomes (n) and the length of each chromosome (l) denoted as O(n * l).

4.2.3 Modified Flower pollination algorithm

The flower pollination algorithm is a meta-heuristic inspired by flowering plants for artificial intelligence. In the Modified Flower pollination algorithm, the ini-

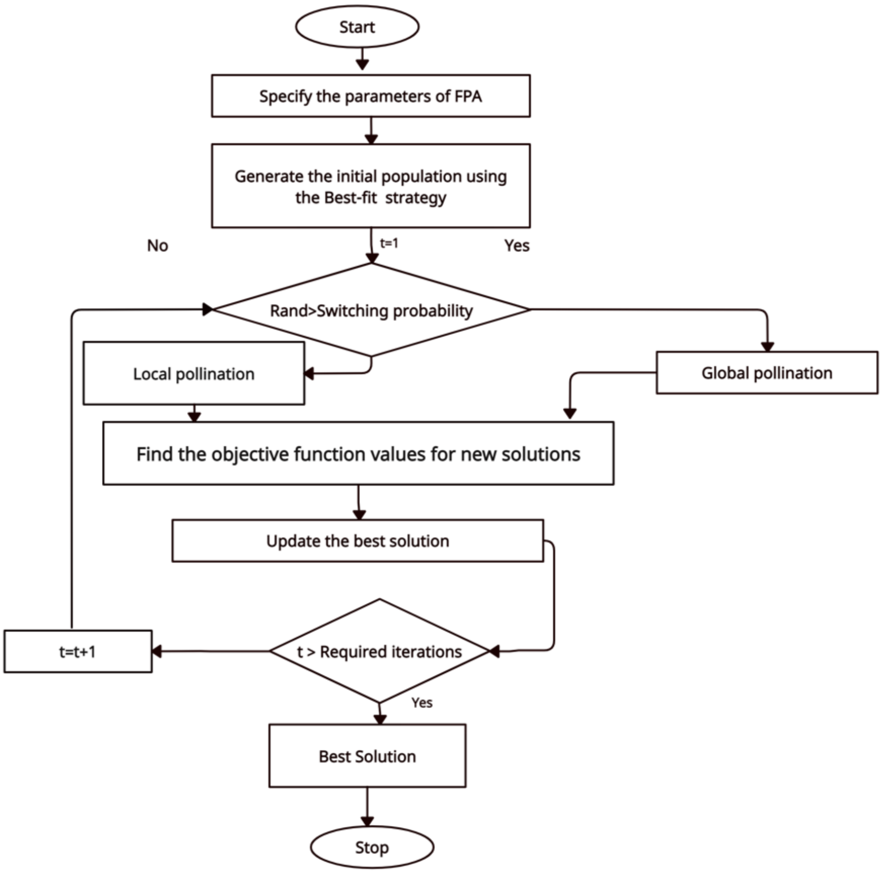


Figure 4.2: Modified flower pollination algorithm flowchart for the Proposed Model

tial population is formed using the best-fit strategy rather than the traditional random method. This change ensures that the starting group consists of individuals with favorable traits, making the algorithm more efficient and effective. By beginning with a population showing promising characteristics, MFPA speeds up the process of finding the best solutions, reducing the workload and improving performance. This strategic start not only accelerates the evolution but also increases the chances of finding superior solutions in fewer attempts. Using the best-fit strategy for population generation in genetic algorithms significantly improves their ability to produce high-quality outputs efficiently.

Algorithm 2 Modified Flower pollination Algorithm

```
Input: An application's set of resources available and unmapped tasks.
Output: Output mapping
Construct a list of the available resources.
Create population
for Each solution do
    Determine the optimum resources (best fit) for every activity based on the time it will take to complete it.
    Go to the next resource in the list.
    if The counter of index = last resource then
        Go to the first resource on the list.
    end if
end for
Find the fitness of each solution in the population using the fitness function
Find the best solution
while Termination condition not reached do
    if (rand) is less than switching probability (.8) then
        performs global pollination
        else
        performs local pollination
    end if
    Find fitness of the new solution
    if new solution better than existing solutions then
        swap with the new solution
    end if
end while
Save the best solution
Map the tasks on resources
```

The steps of modified FPA in the proposed method are described below:

- Step 1: Initialize the population with the help of the best fit strategy to get the response rapidly. Hence to build the initial population, the tasks/cloudlets are sorted by execution times, and their processing capacity in MIPS. Cloudlets are allocated to VMs in the virtual resource list using the best-fit strategy.

- Step 2: Evaluate performance for each solution in the initial population using the fitness function in Equation 4.1. The fitness function is applied to all the solutions, and their values are computed in FPA, after which the solution with the best value is selected as the maximum or minimum for the fittest solution using parameter placement guidelines. In the proposed approach, solutions with minimum execution time can be considered the best value after fitness function calculation. Each solution's fitness value is calculated using Equation 4.1.

- Step 3: Find the best solution among all. The best solution among the others is the one with the lowest fitness value. The goal is to minimize the fitness function as much as possible.

- Step 4: Define switch probability. In the proposed approach, the switch probability is considered as 0.8.

- Step 5: Check the stop criteria. In the proposed approach, the algorithm stops once it reaches the required number of iterations.

- Step 6: Start the main loop of Flower pollination. According to switch probability, perform local pollination or global pollination.

- Step 7: Update new solution and compare with old solutions.

- Step 8: Display the best solution among all.

Figure 4.2 shows the flowchart for the same. The proposed algorithm's pseudocode is included in Algorithm 2.

The switching probability p, the scaling parameter, and the population size n are the parameters in FPA. Empirical findings and numerical simulations imply that a small population is adequate, regardless of whether the real-world population sizes are large. The time complexity of FPA for this implementation depends on the number of iterations (t), the population size (n), and the complexity of the objective function or the number of decision variables (m) which is given as O(n * m * t). The memory complexity of FPA for this work relies on the population size (n) and the number of decision variables (m) denoted as O(n * m).

The proposed approach described in Algorithm 3 is the implementation of investigating resource provisioning based on meta-heuristic methods for microservice-based IoT medical applications in a fog computing environment utilizing mobility components of iFogSim2.

Algorithm 3 Microservice-based IoT application in fog environment

Input: Sensor data
Output: Response to an actuator
Create new FogBroker
Create new Application
Create new dataparser object and location handler object
Adding microservices (Client module, datapreprocessing, decision making, storage) to the application
Using edges to connect application modules in the application model
Defining the relationships of the input and output microservice modules
Defining the microservice application loop as sensor-client module-data preprocessing-decision making-client-actuator
Add application to fogbroker
Create fog devices in the physical topology
Create Cloud datacenter
for each i to locator.getLevelWiseResources **do**
    Create proxy fog device
    Apply modified genetic algorithm for task scheduling and Apply modified flower pollination algorithm for task scheduling
end for
for each i=0 to locator.getLevelWiseResources **do**
    Create gateway fog device
end for
Initialize microservice mapping
Create MicroservicesMobilityClusteringController and submit the controller to the application
Start Simulation
Stop Simulation
End

Figure 4.3: Directional mobility of the user in Melbourne Central Business District [44]

4.3 Experimental setup

The market is very competitive with simulators for simulating cloud, fog, and edge devices. One such discrete event network simulator is NS3, which enables us to create various virtual nodes and install devices, internet stacks, programs, etc., on our nodes with the aid of various classes. For modeling and simulating edge/fog/cloud computing infrastructures and services, we have chosen iFogSim2, an extension of Cloudsim, since this framework can be used to develop and deploy experiments for edge/fog/cloud devices that handle compute, memory, I/O, and VM allocation, as well as VM power models, among other things. The iFogSim2 simulator, an extension of the iFogSim simulator, holds the properties of service migration, distributed cluster building across fog nodes, and microservice orchestration. This simulator helps validate the proposed approach's performance in the fog computing environment. The components of the iFogSim2, such as mobility, clustering, and microservices, are loosely coupled and can be utilized for simulation in such scenarios. iFogSim2 incorporates real datasets for assessing the performance of different service management strategies in fog computing settings, unlike most existing solutions. It includes node clustering, mobility management, and microservice orchestration methodologies that can be used as benchmarks for comparing performance [44].

All iFogSim core classes, such as FogDevice, Actuator, sensor, and AppModule, have object references in the Controller class, and it can access the Tuple class through an Application object. The Clustering element of iFogSim2 allows distributed dynamic coordination and collaboration among multi-fog nodes. To

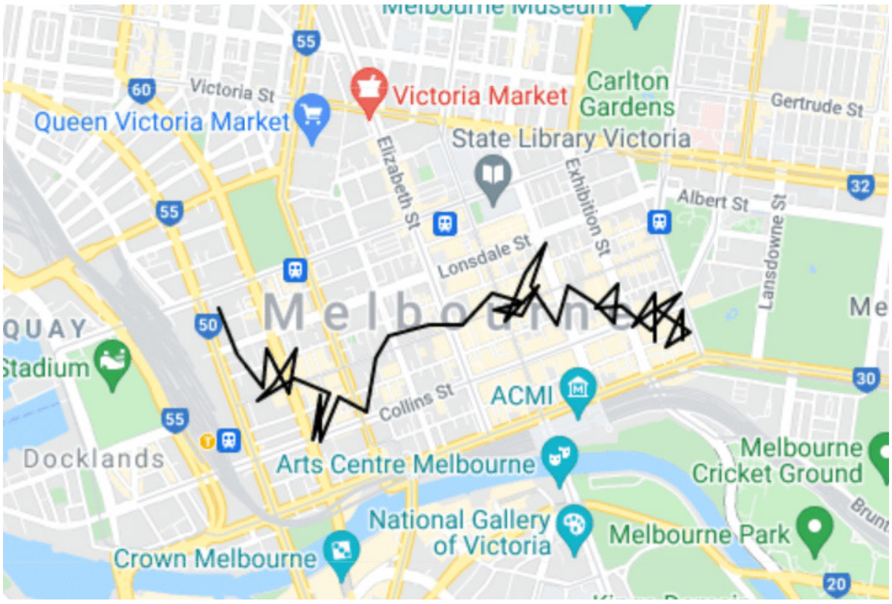


Figure 4.4: Random mobility of the user in Melbourne Central Business District [44]

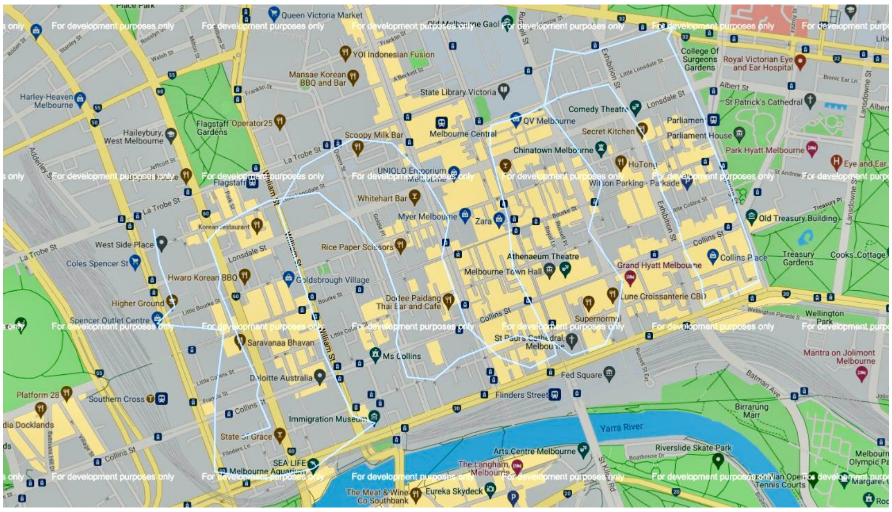


Figure 4.5: User mobility Pattern in Melbourne Central Business District [44]

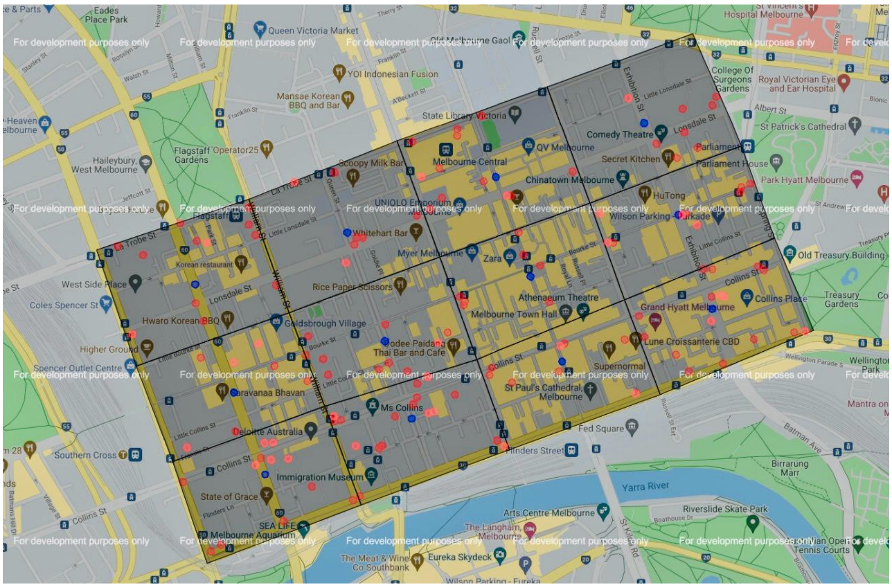


Figure 4.6: Block-wise Edge/Fog computing nodes in Melbourne Central Business District [44]

add the modified mobility component, which is customized for the proposed use case to the iFogSim2 simulation, it includes classes such as DataParser and MobilityController. The functions of these classes are described below:

- The DataParser class allows the separation and assimilation of location data from many IoT end devices so that application services may be handled based on their unique mobility patterns.
- MobilityController class dynamically starts required sequential or parallel actions on separate FogDevice and AppModule referenced objects for mobility management.

In considering the diverse movements of users, the proposed system explores scenarios where users either adhere to consistent daily movement patterns or follow different random paths each day. To address this variability, during simulations, the proposed model considers two different types of mobility patterns namely 'DIRECTIONAL MOBILITY' and 'RANDOM MOBILITY'. The 'DIRECTIONAL MOBILITY' model is being used, which has a significant number of consecutive coordinates lying at the same distances across the Melbourne central business district (CBD) for a user/IoT device. In 'RANDOM MOBILITY' model, the time period between any two of these motions is made to be equal to ensure that the user/IoT device maintains a constant speed. Based on those coordinates, events are constructed to simulate the movement of the associated end IoT device. There are numerous random mobility patterns to represent users 'RANDOM MOBILITY' model behaviors. According to numerous mobility criteria, user speed, direction, stopping time at each location, and duration within each edge/fog node's communication range can be produced by the RandomMobilityGenerator class and can be expanded for multiple random mobility models. Figures 4.3 and 4.4 depict directional and random user movements of a user in the Australian region [178]. Figures 4.5 and 4.6 illustrate user mobility and resource location characteristics within the dataset. The diagrammatic representation of the connection between the mobility components of iFogSim2 are presented in Figure 4.7.

This section explains the simulation environment used to evaluate the suggested approach's performance. The sensors detect ECG details and regularly send the data to the fog nodes through a smartphone. Data is processed and analyzed on the fog nodes to determine whether the patient's health status is normal or critical. The results are subsequently sent to the patient's smartphone and to the cloud for storage. The fog nodes' connection to the cloud server is established through a proxy server. The client module is embedded in IoT devices to get the sensor data. The processing module is embedded in fog nodes to process and analyze the incoming data and to assess the patient's health status. The fog node then communicates the results to the associated IoT device, which displays them. It must define values for numerous parameters such as CPU length, bandwidth, RAM, and so on, in iFogSim2 when generating fog devices. The settings used for device configuration in iFogSim2 [64] are listed in Table 4.1. The typical unit of measurement for latency is milliseconds, which indicates the amount of time it takes for a tuple to travel from the sensor to the mobile device, from the mobile device to the Fog, from the Fog to the Proxy, from the Proxy to the Cloud, and between clusters of Fog nodes. Table 4.2 displays the values assigned to these parameters in the configuration settings of iFogSim2.

Table 4.1: Configuration parameters [64]

Parameter	**Cloud**	**Fog**	**Smartphone**
CPU length (MIPS)	44800	2800	2800
RAM (MB)	40000	4000	4000
Uplink BW (MB)	100	10000	10000
Downlink BW (MB)	10000	10000	10000
Busy power (J)	16*103	107.339	87.53
Idle power (J)	16*83.25	83.433	82.44

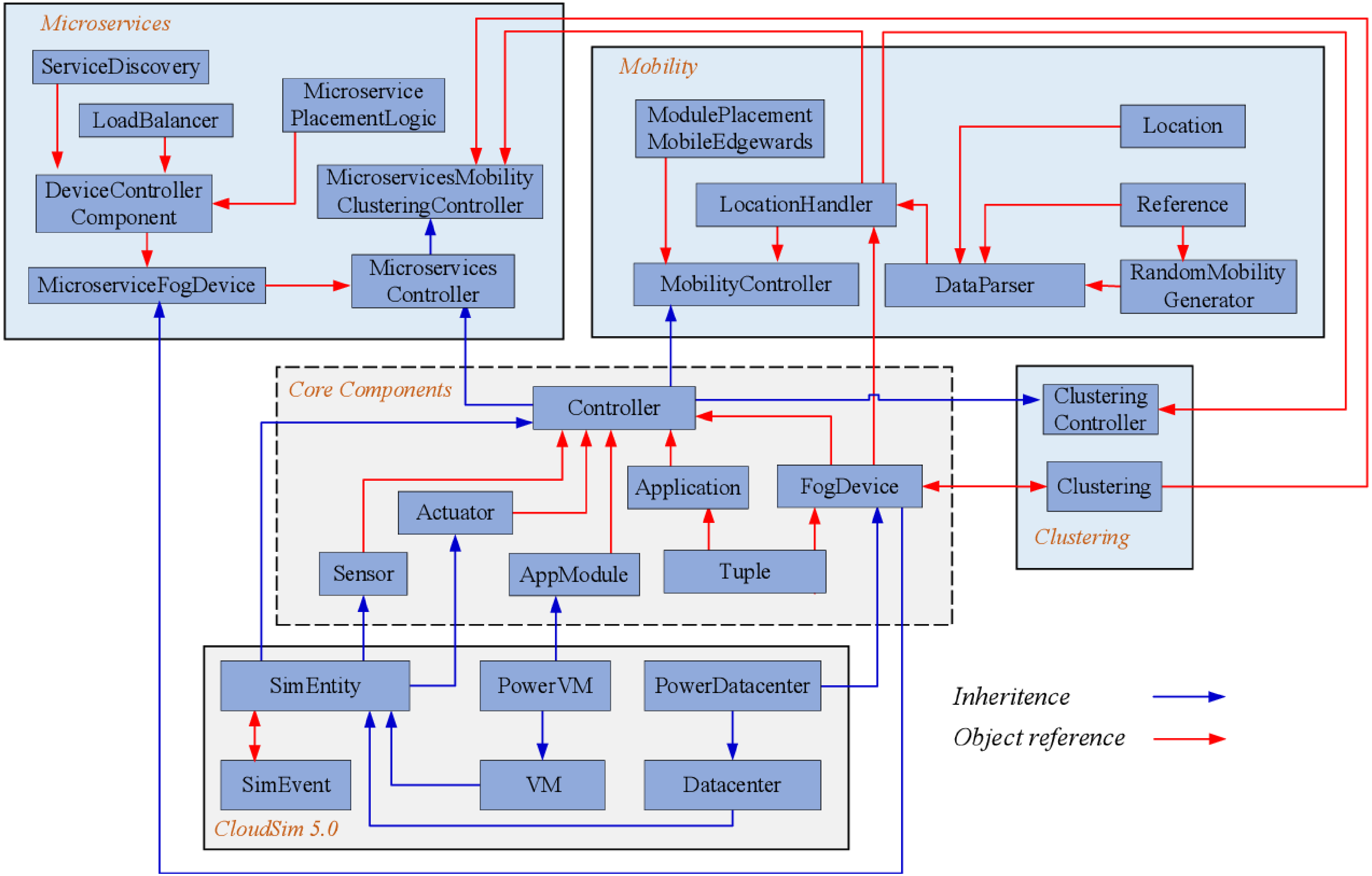


Figure 4.7: iFogsim2 interconnection between components [44]

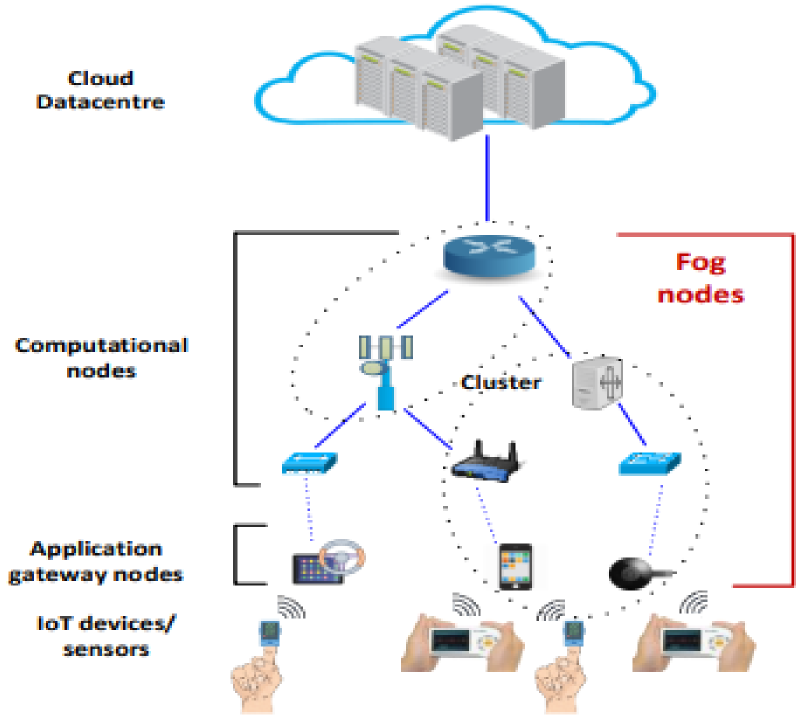


Figure 4.8: iFogsim2 Layered configuration of devices [44]

Table 4.2: Latency and Value Table for the Proposed Approach

Latency	**Value**
Sensor to mobile	5 ms
Mobile to Fog	20 ms
Fog to Proxy	20 ms
Proxy to Cloud	30 ms
Fog node clusters	2 ms

In the simulation, edge/fog devices are the computational devices in iFogSim2. Computational gadgets, on the other hand, come in various levels. On Level 3, the parent node is a cloud server. The fog nodes are connected to the cloud server via a proxy server at Level 2. Fog nodes are located closer to the user at Level 1, giving more frequent computational and storage capacities. Sensors and actuators are used in Level 0 IoT devices. The MicroserviceFogDevice, Actuator, and Sensor classes of iFogSim2 simulate the physical topology. The layers within the architecture of iFogSim2, as described above, are depicted in Figure 4.8.

These scenarios were simulated on an Intel Core i7 CPU running at 1.80 GHz and 4GB of RAM. The fractional selectivity of the input-output relationship inside a module is set to 1.0.

4.4 Results and analysis

In our research, a comprehensive approach was undertaken by integrating specific datasets to evaluate the proposed meta-heuristic-based resource provisioning model for healthcare IoT microservices, as elaborated in Chapter 3.4.2. The incorporation of a mobility dataset allowed us to simulate the dynamic movement patterns of IoT devices, mimicking real-world scenarios where devices exhibit mobility. Additionally, the utilization of an ECG dataset enabled the emulation of authentic healthcare data generation, providing a realistic foundation for evaluating the performance of our system in healthcare applications. Chapter 3.5 meticulously details the pivotal parameters considered for evaluation, including energy consumption, cost, latency, execution time, and network usage. These parameters were chosen based on their critical relevance to the efficiency and effectiveness of resource provisioning in healthcare IoT microservices. Our methodical description and utilization of diverse datasets, coupled with the meticulous examination of key evaluation parameters, contribute to the robustness and applicability of our research findings within the realm of healthcare IoT microservices.

The proposed work compares the system's performance with two approaches, namely, cloud-only and edge-fog with cloud, changing the deployment of meta-heuristic algorithms for resource management. The section also presents the results with and without mobility considerations of IoT devices. Parameters, namely cloud energy, router energy, cost, network use, latency, and execution time, are analyzed with and without mobility, the results of which are presented in Figure 4.9. The proposed system's performance is evaluated using the modified genetic algorithm for resource provisioning, considering the identified parameters, and are presented in Table 4.3 by utilizing the evaluation metrics discussed in section 3.5. The proposed system's performance is also evaluated using the modified flower pollination algorithm, considering the same parameters, and is presented in Table 4.4. The fitness of solutions for modified FPA and modified GA are depicted in Figure 4.10. The comparison of fog implementation, edge implementation, and cloud-only implementations for the identified parameters using modified GA, modified FPA, the multiobjective optimization method and the existing First Come First Serve (FCFS) method in the simulator are presented in Figure 4.11.

Table 4.3: Results of Modified Genetic algorithm based resource provisioning

GA	Edge	Fog	Cloud
Cloud energy (J)	2728189	2664000	2717900
Router energy (J)	174607	174718	172110
Cost ($)	67162	81529	92414
Network use (B)	8309	9961	13907
Latency (ms)	23.4	23	247
Execution time (ms)	747	1417	1660

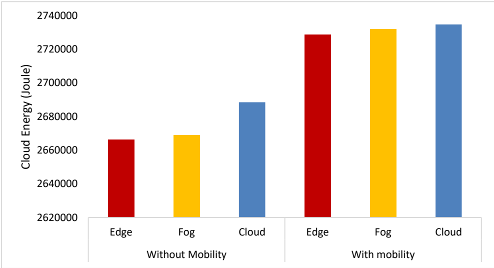
The results presented in Figure 4.12 and Figure 4.13 describes the results of modified FPA and modified GA for resource provisioning while taking into account directional and random mobility models for user movements with microservice placement and clustering approach and compares the proposed models for different movement patterns of a patient.

Table 4.4: Results of Modified Flower pollination algorithm based resource provisioning

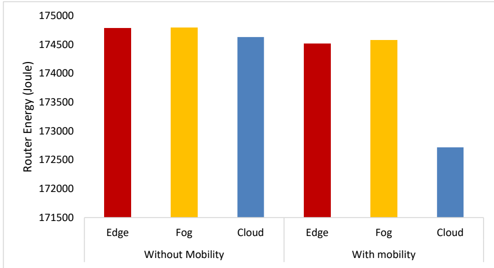
Flower pollination algorithm	Edge	Fog	Cloud
Cloud energy (J)	2728556	2732075	2734776
Router energy (J)	174608	174718	172719
Cost ($)	58354	68384	79714
Network use (B)	8709	10045	13773
Latency (ms)	23.3	25.6	249
Execution time (ms)	781	1337	1381

4.4.1 Observations from the integration of mobility and microservice features in edge-fog-cloud computing environments

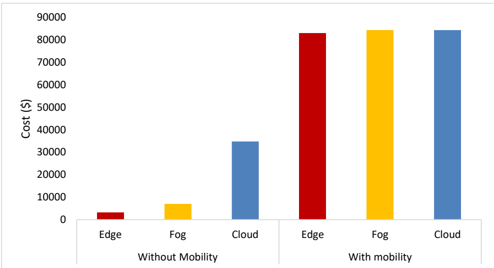
- In comparison to simulations lacking mobility components, simulations with mobility systems are practical
- Cloud energy, cost, network usage, execution time, and latency are all reduced in the fog computing model since processing happens at the lower



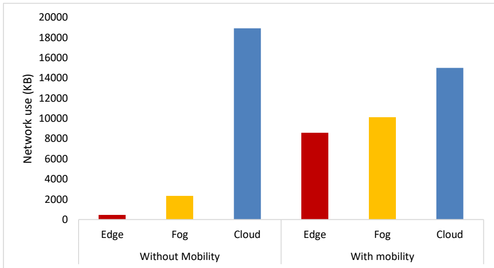
(a) Cloud Energy



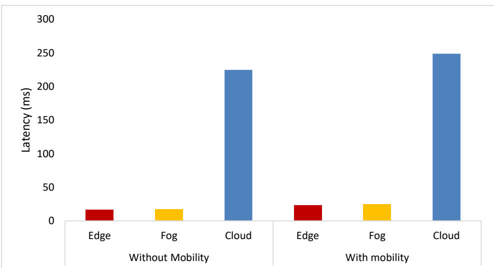
(b) Router Energy



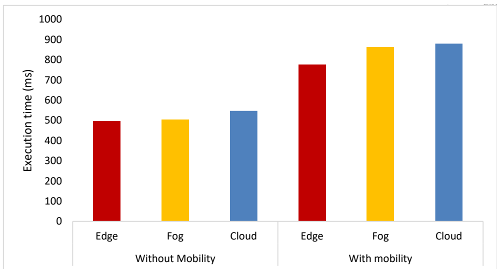
(c) Cost



(d) Network use

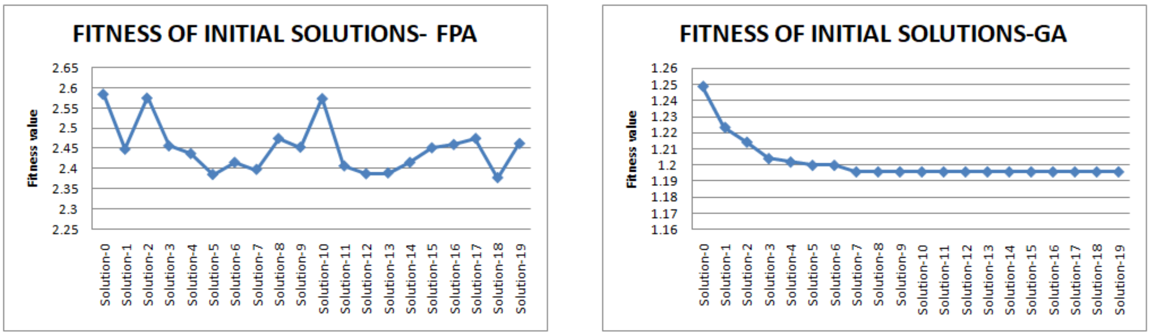


(e) Latency



(f) Execution time

Figure 4.9: Result graphs with and without mobility for the Proposed Approach



(a) Modified FPA (b) Modified GA

Figure 4.10: Fitness values

fog level, which is very close to the end device both in mobility and without mobility implementations. This brings practical proof of performing the computation at the fog level

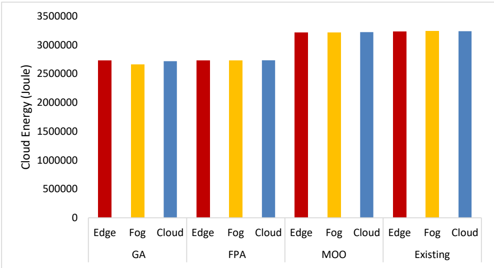
- Router energy consumption is less for cloud-only implementations if we deploy mobility and microservice concepts since microservice modules are loosely coupled, which consume less energy than monolithic architecture

- If we include mobility in our application, the network use, execution time, and delay for cloud and fog scenarios is slightly longer in milliseconds since mobility requires service migration which needs more network requirements and causes an increase in execution time and delay

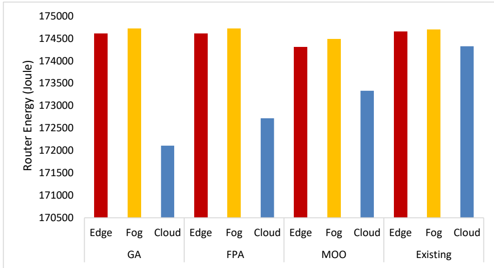
4.4.2 Observations from the integration of meta-heuristic methods in edge-fog-cloud computing environments

- Metaheurtic methods provide efficient provisioning compared to existing systems. The reason being, meta-heuristic methods converge to optimal or sub-optimal solutions at a faster rate when compared to multi-objective optimization approaches

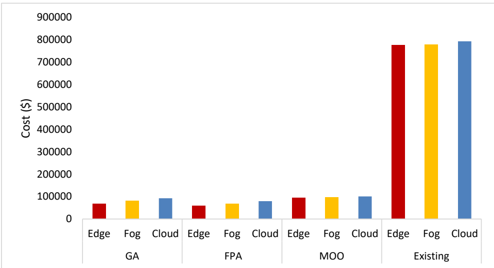
- The energy consumption of routers are same for GA and FPA since more computations are happening in the fog layer



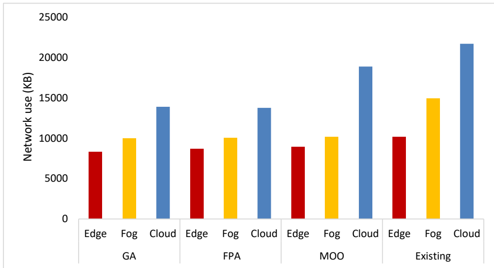
(a) Cloud Energy



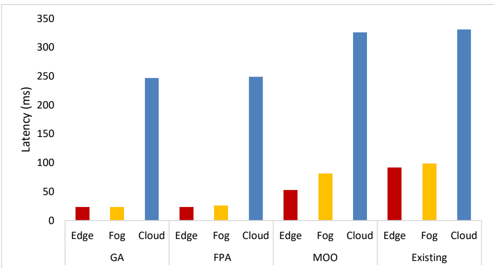
(b) Router Energy



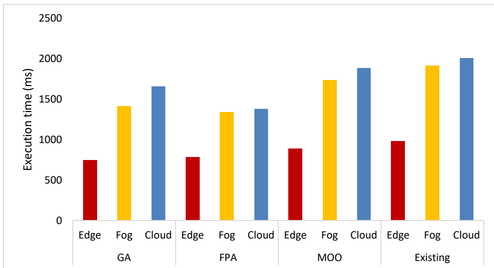
(c) Cost



(d) Network use

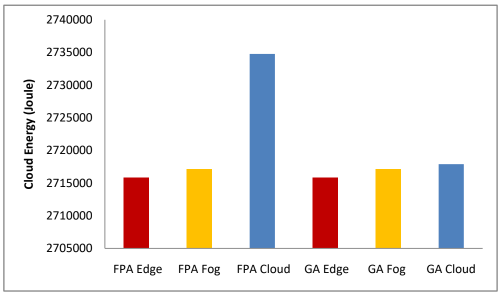


(e) Latency

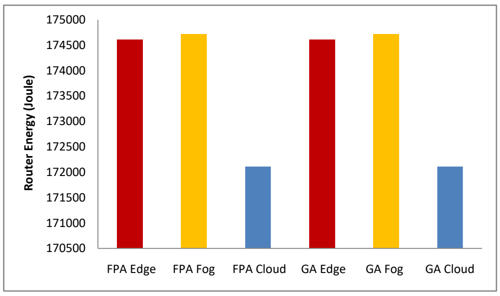


(f) Execution time

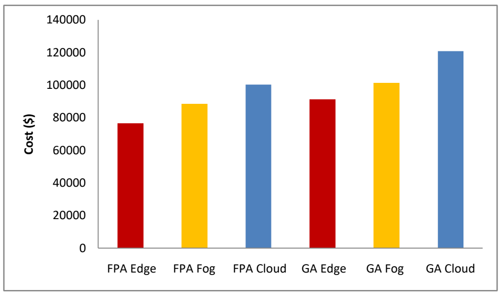
Figure 4.11: Result graphs comparing modified GA, modified FPA, multi-objective optimization method and existing FCFS method



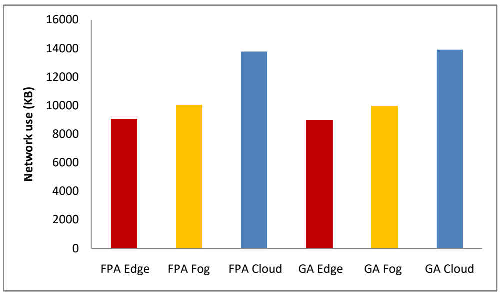
(a) Cloud Energy



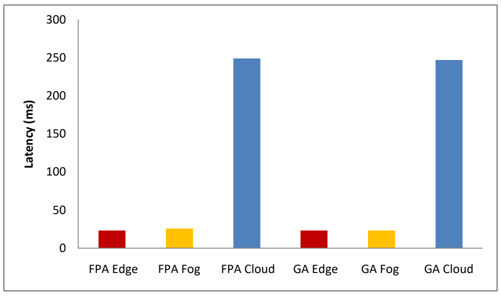
(b) Router Energy



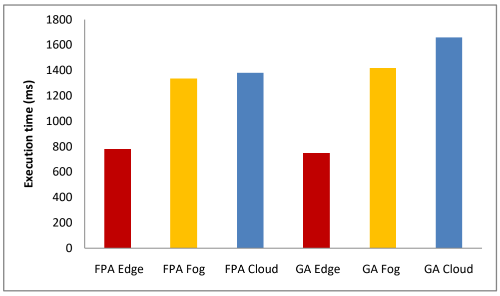
(c) Cost



(d) Network use

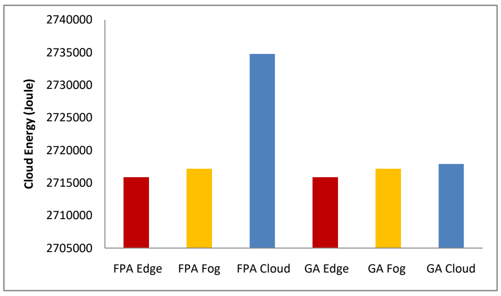


(e) Latency

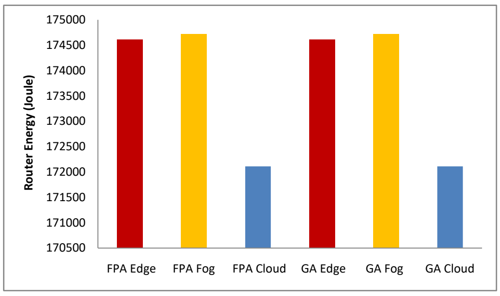


(f) Execution time

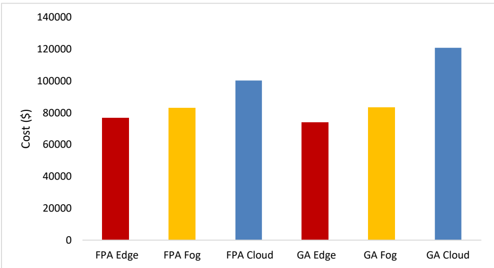
Figure 4.12: Results of modified FPA and modified GA in fog and edge computing using directional mobility user model



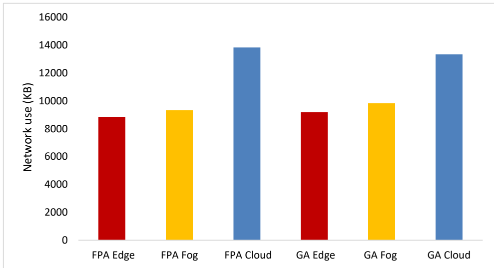
(a) Cloud Energy



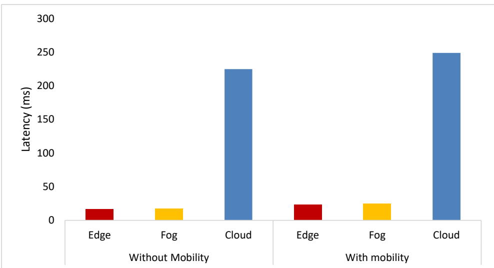
(b) Router Energy



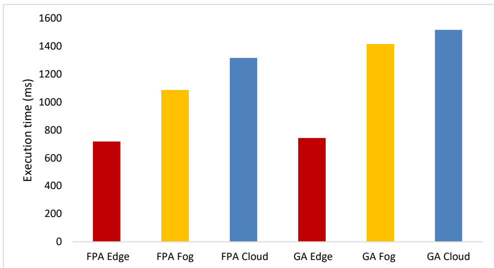
(c) Cost



(d) Network use

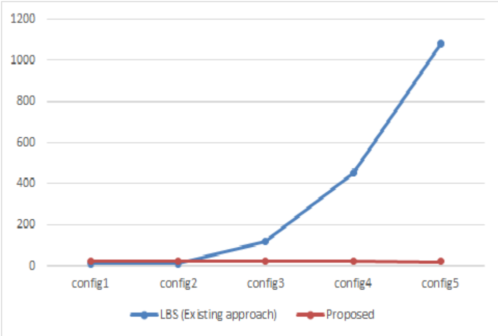


(e) Latency

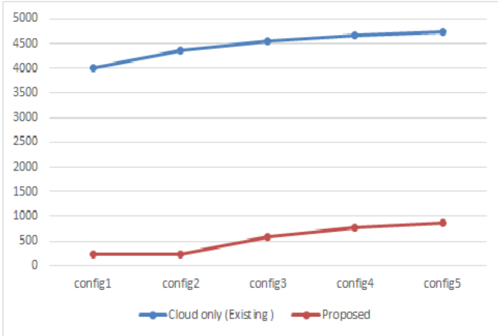


(f) Execution time

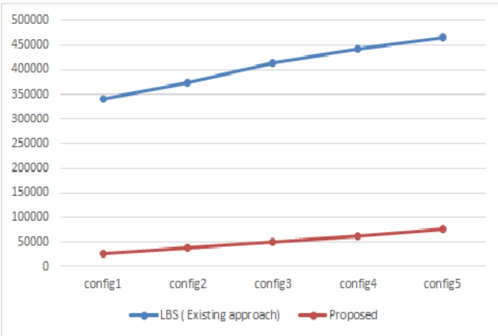
Figure 4.13: Results of modified FPA and modified GA in fog and edge computing using random mobility user model



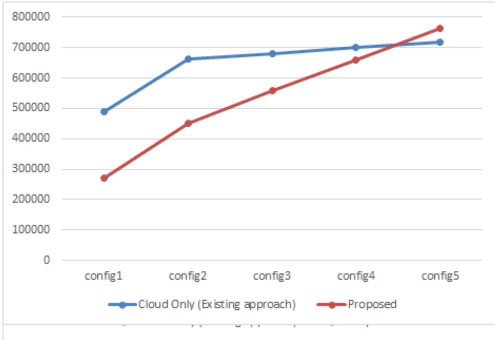
(a) Latency fog



(b) Latency cloud



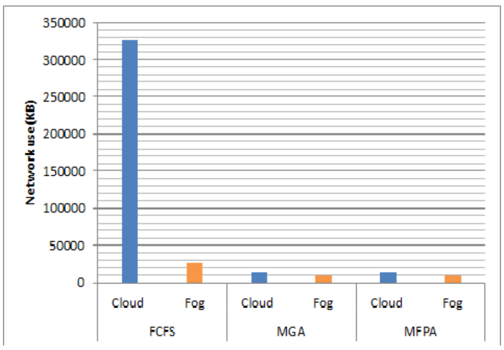
(c) Network use fog



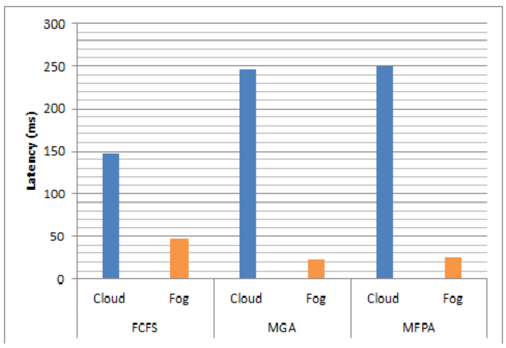
(d) Network use cloud

Figure 4.14: Comparative analysis [101]

- When compared to GA, FPA has less cost of execution since FPA converges faster than GA. GA and FPA have almost the same network use and latency
- In cloud scenarios, FPA takes less time to execute since FPA converges fast, but GA and FPA take about the same amount of time to execute in edge/fog scenarios
- While considering the parameters, cloud energy, cost, network use, execution time, and latency, genetic algorithm, and FPA outperforms the existing



(a) Network use comparison with the existing method



(b) Latency comparison with the existing method

Figure 4.15: Comparative analysis [183]

resource provisioning methods since meta-heuristic methods converge to optimal or sub-optimal solutions fast. However, router energy is slightly higher than traditional resource provisioning methods since more computations are happening in the fog layer

- In comparison, cloud energy consumption decreases by 15%, network use by 7%, cost by 29%, execution time by 16% and latency by 55% when using meta-heuristic based resource provisioning approach for edge computing. While cloud energy consumption decreases by 18%, network use by 33%, cost by 16%, execution time by 18% and latency by 72% when using meta-heuristic based resource provisioning approach for fog computing
- The router energy consumption is 0.1% more for edge/fog computing since more computations are happening in the fog layer for the meta-heuristic methods considered in this work

4.4.3 Observations of directional and random mobility user models in edge-fog-cloud computing environments

- Cloud energy is lesser for edge computing than fog computing since computation occurs only in edge devices
- Router energy is the same for edge computing compared to fog computing because computations are happening in the router for edge and fog computing in a similar manner
- Since computations are happening in the edge layer, edge computing has less cost of execution, network use, latency, and execution time than fog computing for directional and random user mobility models

4.4.4 Comparative analysis

In order to have an effective conclusion, the proposed approach has been compared for its network use and the latency parameters against the existing results presented in the literature [101] and [183] which are presented in Figures 4.14 and 4.15. The comparison of the parameters for the different architectures named config 1, 2, 3, 4, and 5, described in Table 4.5 also prove the conclusion statement presented in the previous subsection.

Table 4.5: Topologies for simulation for the Proposed Approach

Configurations	End IoT devices	Total IoT devices
config 1	4	24
config 2	6	36
config 3	8	48
config 4	10	60
config 5	12	72

4.4.5 Sensitivity analysis

Sensitivity analysis is important for analyzing the robustness and reliability of resource provisioning mechanisms implemented through metaheuristic methods. By systematically varying input parameters, sensitivity analysis enable to evaluate the responsiveness and stability of the provisioning system to changes in the input factors. This analysis helps identify critical parameters that significantly influence resource allocation decisions and performance metrics, guiding optimization efforts for enhancing system efficiency and effectiveness. Moreover, sensitivity analysis facilitates risk assessment by uncovering potential vulnerabilities and uncertainties in the provisioning process, enable to develop mitigation strategies and ensure the resilience of the system in real-world scenarios. Overall, sensitivity analysis plays a crucial role in validating, optimizing, and enhancing resource provisioning mechanisms based on metaheuristic methods, contributing to the development of robust and scalable solutions. The proposed model has been tested for its sensitivity to minor variations in the input feed. The results thus obtained conclude that the small changes in the input cause little to no change in the output measurements. This proves that the model is less sensitive to minimal changes in the input and is robust as can be seen in the results presented in Tables 4.6 and 4.7.

Table 4.6: Sensitivity analysis for Modified GA

(a) One percentage of increase in input for GA Sensitivity analysis

Genetic algorithm	**Edge**	**Fog**	**Cloud**
Cloud energy (J)	2720126	2721507	2729185
Router energy (J)	173851	173934	172348
Cost ($)	67162	81529	92414
Network use (B)	8309	8668	13586
Latency (ms)	24.3	25.8	249
Execution time (ms)	808	995	1001

(b) One percentage of decrease in input for GA Sensitivity analysis

Genetic algorithm	**Edge**	**Fog**	**Cloud**
Cloud energy (J)	2705161	2712235	2720227
Router energy (J)	173845	174028	172583
Cost ($)	58353	68384	79714
Network use (B)	9857	11029	17340
Latency (ms)	21.3	23.8	249
Execution time (ms)	736	825	954

Table 4.7: Sensitivity analysis for Modified FPA

(a) One percentage of increase in input for FPA Sensitivity analysis

FPA	**Edge**	**Fog**	**Cloud**
Cloud energy (J)	2715807	2717151	2723475
Router energy (J)	174061	173649	172052
Cost ($)	73448	75352	83116
Network use (B)	8308	8803	13539
Latency (ms)	24	25.6	248
Execution time (ms)	764	806	965

(b) One percentage of decrease in input for FPA Sensitivity analysis

FPA	**Edge**	**Fog**	**Cloud**
Cloud energy (J)	2704068	2706462	2717702
Router energy (J)	173819	174005	172664
Cost ($)	56806	60199	76135
Network use (B)	10181	10948	16789
Latency (ms)	21.4	25	249
Execution time (ms)	788	790	840

4.5 Summary

This chapter intends to conclude that edge computing surpasses fog computing, emphasizing the notable efficiency in resource provisioning achieved through the application of metaheuristic techniques. The examination of these findings underscores the advantages inherent in edge computing paradigms, demonstrating the effectiveness of advanced optimization strategies for enhancing resource allocation within evolving computing architectures. However, in real time scenarios, there may be around 30 hops between the IoT device and the destination server making fog computing and cloud computing very distinct in real-world deployments. In such scenarios, it is evident that the deployment of the microservices in the edge/fog layers would be beneficial. However, since we only simulate a few hops, there will not be a significant difference between edge computing and fog computing in the simulator.

Existing systems in edge and fog-based medical applications grapple with several challenges, notably concerning data privacy and efficiency. Traditional centralized approaches, when applied in these distributed environments, often lead to latency issues and pose privacy risks as sensitive medical data is transmitted to a central server. These concerns are further exacerbated by the stringent privacy regulations governing healthcare. Therefore, to address these pressing issues and harness the potential of edge and fog computing in medical applications, the adoption of federated learning is crucial. This approach facilitates collaborative model training while ensuring data remains localized, offering an effective solution to privacy and efficiency challenges and ultimately advancing the accuracy and dependability of machine learning models in these decentralized healthcare environments. The next chapter presents Resource provisioning based on Federated learning.

Chapter 5

Resource Provisioning based on Federated Learning

5.1 Introduction

The concept of intelligent healthcare involves utilizing AI to learn and analyze patient data. However, it can be challenging to find large and diverse datasets to train machine learning models in individual medical centers. This means that traditional centralized AI methods require sensitive data to be moved from medical facilities to data centers, which not only increases the demand for communication resources and energy, but also violates privacy. This has become a significant obstacle in promoting scientific collaboration between trans-national clinical medical research centers. A distributed AI technique known as federated learning has emerged that enables the cooperative training of ML models without the sharing of patient data. Federated learning may prove to be an advantageous method for facilitating IoT based intelligent applications [184–186].

Due to the necessity for real time processing, low latency, and privacy considerations, Edge/Fog computing is becoming more and more significant for medical applications [187]. Edge computing can limit the quantity of data that must be transferred to centralized servers or the cloud by processing and analyzing the data closer to the data's source, thereby reducing network traffic and delays [47]. Additionally, Edge computing can help address privacy concerns by keeping sensitive data within a local network and limiting access to authorized users only. In medical applications, where time and accuracy are critical, and privacy is essential, Edge computing has become a necessity [68], [188].

Real time ECG abnormality detection is one of the applications in medicine that has several advantages for patient care. First and foremost, it allows healthcare providers to quickly identify and respond to cardiac abnormalities, potentially saving lives. Early detection and treatment of cardiac abnormalities can prevent more serious and costly health issues down the road. Moreover, real time ECG anomaly detection can help reduce healthcare costs and improve patient outcomes. By continuously monitoring ECG signals in real-time, the system can immediately detect anomalies and alert healthcare providers, who can take action to diagnose and treat the patient. Another benefit is that real-time ECG anomaly detection can improve the accuracy of diagnoses. In some cases, anomalies may be missed or misinterpreted when relying on visual inspections alone. With automated detection, the system can analyze the ECG signals with greater precision, reducing the risk of errors and false negatives. Additionally, real time monitoring can help identify potential issues before they become acute, reducing the likelihood of hospitalizations and emergency room visits. Overall, real-time ECG anomaly detection has the potential to improve patient care, increase accuracy, and reduce healthcare costs, making it a valuable tool in healthcare. Since the healthcare issues related to ECG anomaly detection in microservice-based IoT systems are not sufficiently addressed by existing research on edge/fog/cloud federated learning approaches, we were motivated to do this study.

5.2 Proposed Model

Federated Learning is a distributed privacy-preserving machine learning paradigm in which a central server connects with various end devices, including smartphones, laptops, and security cameras, with limited computation and storage availability. Hence the clients avoid sharing the data with the server. Clients receive the server's most recent global model for each communication round, and a small percentage of clients use stochastic gradient descent (SGD) to update it throughout several rounds. The new global model is then obtained by aggregating these updated parameters on the central server. Most of the server's cloud deployments need enormous storage and computing capacity. In the proposed system, the edge/fog devices use a methodology named FedAvg to launch the federated learning module. This selection is based on the findings from the literature survey conducted in Section 2.6, which highlight FedAvg's effectiveness specifically for edge applications. Federated averaging is a communication-efficient approach for distributed training with multiple clients. Compared to traditional training and learning, FedAvg considerably lowers the communication cost between servers and clients by involving many local SGD updates and one aggregate by the server in each communication cycle [189]. The FedAvg module used in this work is depicted in Figure 5.1.

To initiate the process, an overall model is downloaded from the central server and is trained with local data over several epochs. The outcomes are the local updates. These local model updates collected from the end devices are aggregated by the FegAvg algorithm to generate the global model, which is continued until the required performance is achieved. The proposed method considers three scenarios where the FedAvg is deployed in different layers: edge, fog, and cloud.

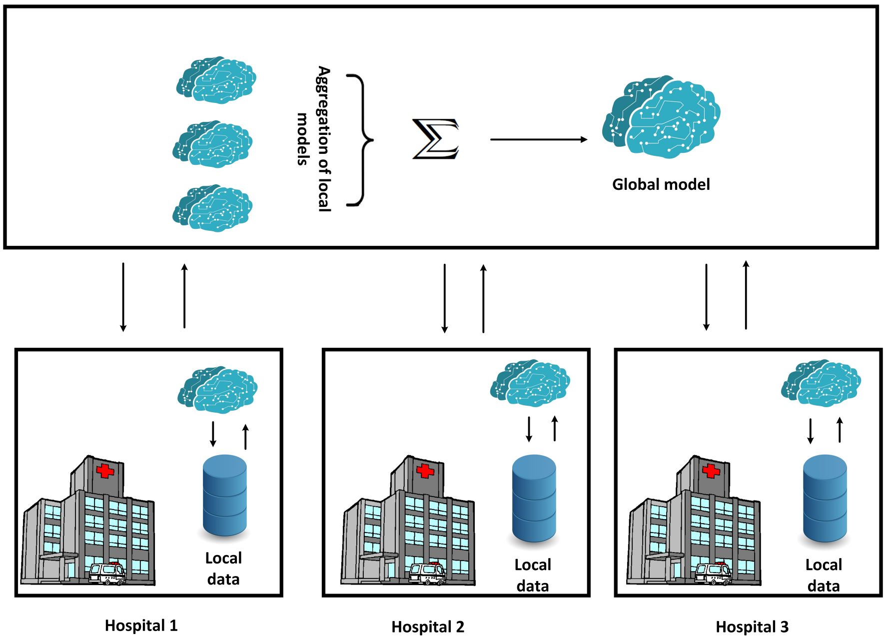


Figure 5.1: Structure of FedAvg for FedSDM

The proposed approach also uses autoencoders, a particular type of neural network used for training and testing the data to detect anomalies in the ECG readings. An autoencoder consists of three components: encoder, code, and decoder. The encoder compresses the input and produces the code, which is later used by the decoder to reconstruct the input. An encoding technique, a decoding technique, and a loss function to compare the output with the objective are required when building an autoencoder. Autoencoders can only compress data meaningfully similar to what they have been trained on. Although the autoencoder's output will not be a perfect replica of its input, it will be a similar degraded representation. The encoder and decoder are both fully linked feedforward neural networks. Figure 5.2 depicts the flowchart of the proposed system's autoencoder module.

The architecture of the autoencoder is presented in Figure 5.3 (a). The proposed approach has two layers in both the encoder and decoder, without accounting for the input and output, as shown in Figure 5.3 (b). The number of nodes per layer reduces with each subsequent encoder layer and grows back in the decoder.

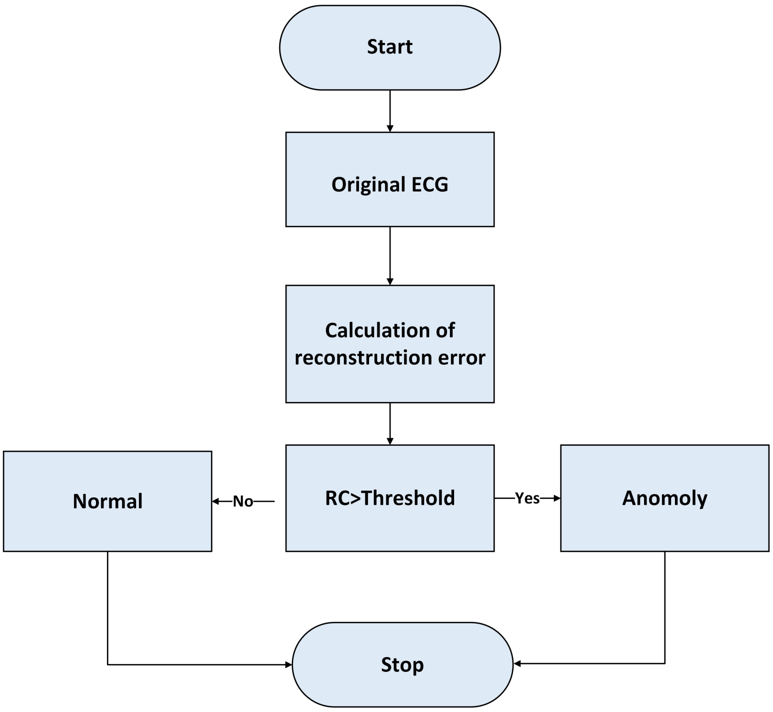
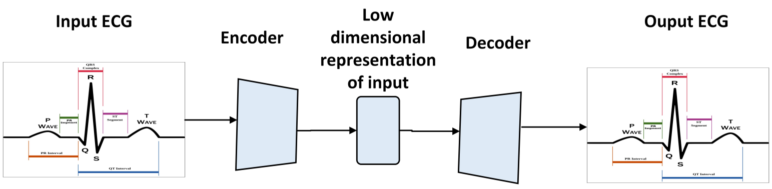
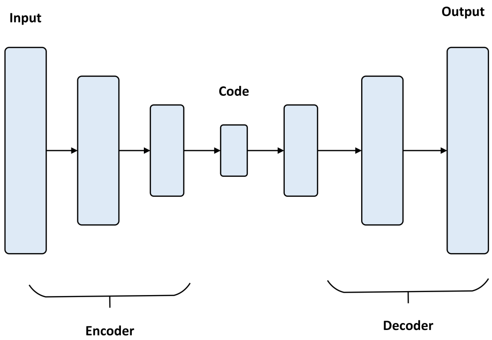


Figure 5.2: Flowchart - Autoencoder in FedSDM

In terms of layer structure, the decoder and encoder are also symmetrical. The loss function is the mean squared error in the proposed system configuration.



(a) Auto encoder architecture for FedSDM



(b) ANN for the Proposed Approach

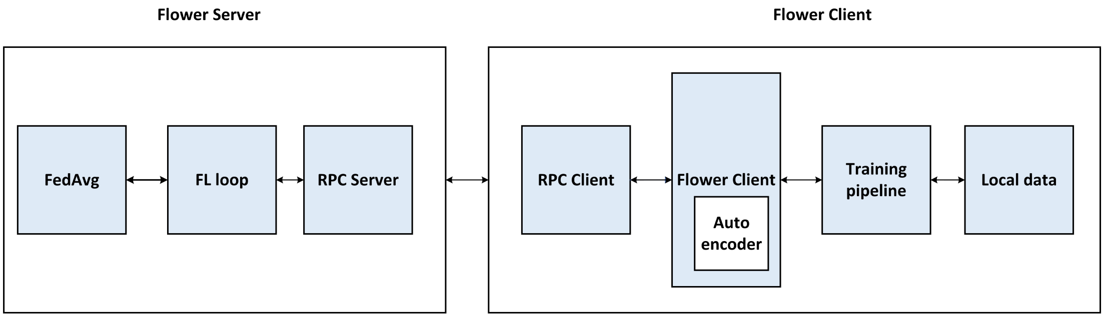
Figure 5.3: Auto encoder and ANN for FedSDM

The proposed architecture implements an encoder and a decoder using an ANN architecture. The ECG data is fed as input to the model, and the model tries to reconstruct it. The error between the original data and the reconstructed output will be called the reconstruction error. Based on this reconstruction error, the ECG data is classified as anomalous. In order to do this, the model is first trained on the standard ECG data and is tested on the complete test set. The autoencoder reconstructs the abnormal ECG when the input is provided. However, since it has been trained only on the standard ECG data, the output will have a more significant reconstruction error. The input is classified as anomalous if the reconstruction error exceeds the threshold. The proposed system uses the Keras Subclassing API to build the model, as it provides reasonable control over the model compared to Sequential API. Autoencoders are unsupervised learning models, but the proposed method trains them using the supervised method, so it is more like they are used as self-supervised.

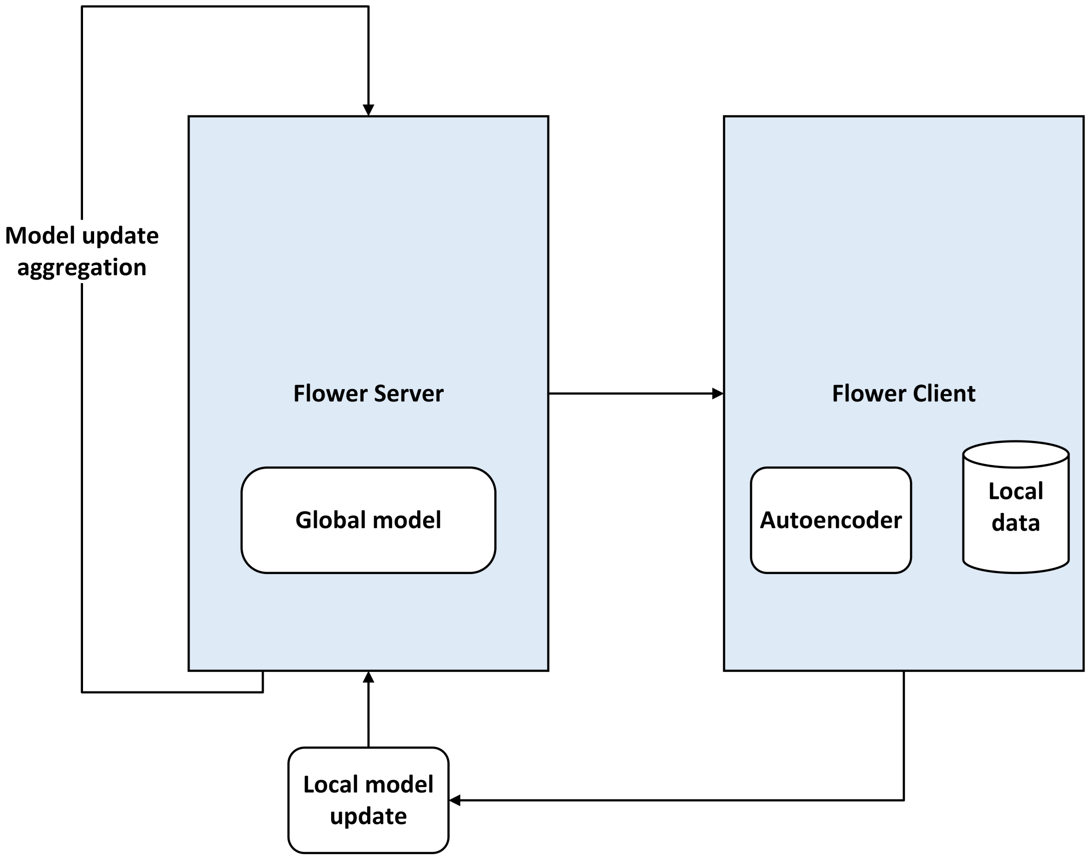
Large-scale FL experiments can be deployed and carried out using FL frameworks. The flower is a comprehensive FL framework that offer new tools to conduct large-scale FL experiments and consider highly heterogeneous FL device environments. It can run FL experiments with clients up to 15M in size. It uses only a pair of high-end GPUs. We have selected the flower framework to implement the Federated Learning module of the proposed system. The architecture of the flower framework for the proposed system is depicted in Figure 5.4 (a). As can be seen in Figure 5.4 (b), the autoencoder is added to the flower framework.

5.2.1 Proposed method - Resource Provisioning based on Federated Learning

The proposed method works as follows: ECG sensor values collected from the patient are stored in the edge device (Ex: mobile phones). The client microservice and the data preprocessing microservice, resides on the edge, do the required computation. The preprocessed ECG values are fed to the Smart Decision Making module, which checks for the anomaly. If any anomaly exists, the notification is sent to the end device. The proposed architecture compares the results of placing the FL-based decision making module at different layers, as mentioned in the previous section. In all the placement policies, the local updates from the corresponding devices/nodes are aggregated in the respective layer (edge/fog/cloud). The diagrammatic representation of the different deployment policies is presented in Figure 5.5. Each deployment has been compared for its performance in the learning efficiency and the QoS parameters, which are presented in the next section.

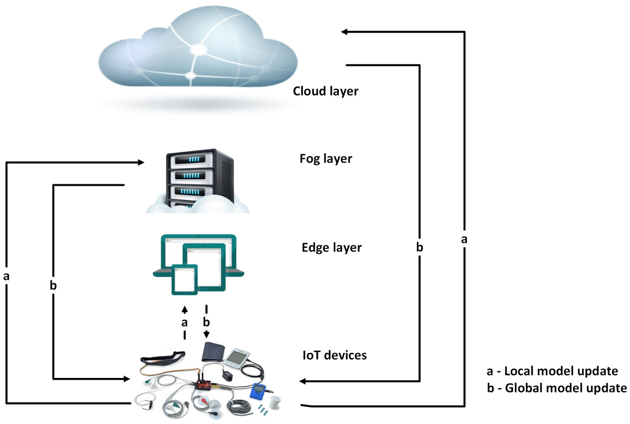


(a) Modified Flower framework for FedSDM

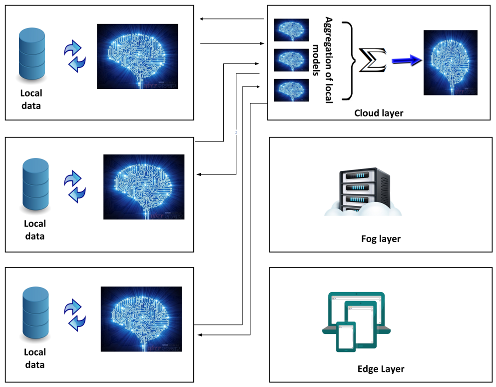


(b) Autoencoder in Flower framework for FedSDM

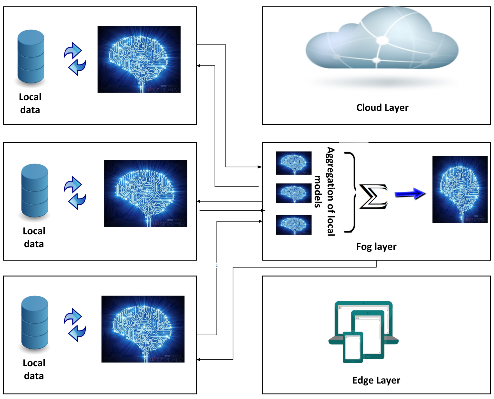
Figure 5.4: Auto encoder integrated Flower framework for FedSDM



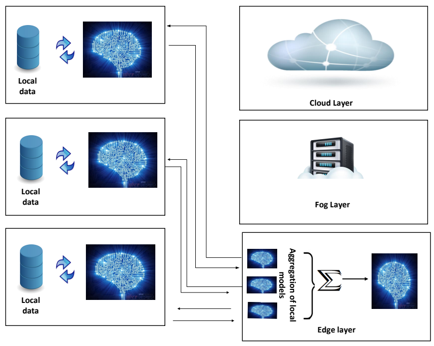
(a) Layered FL



(b) FL deployment in cloud



(c) FL deployment in fog



(d) FL deployment in edge

Figure 5.5: FL deployment in different computing layers for FedSDM

5.3 Results and analysis

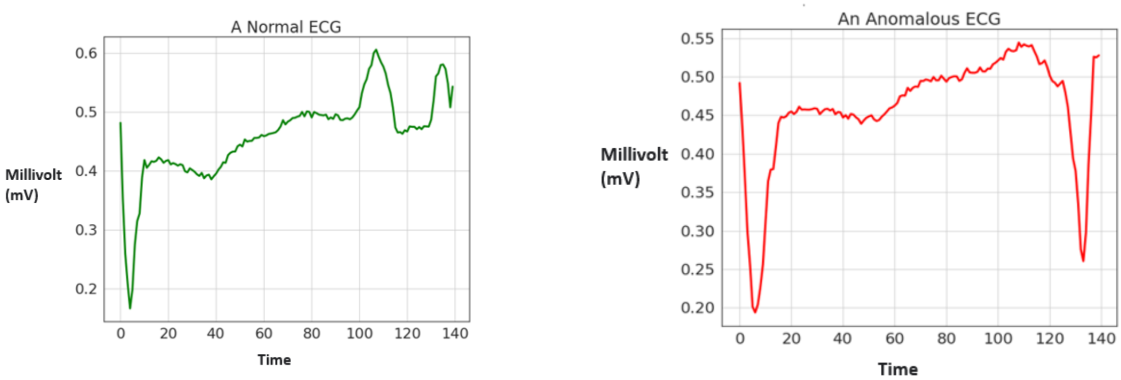
5.3.1 Data set and Evaluation Parameters in Federated Learning for Healthcare Applications

In our research on Federated Learning, a comprehensive approach was employed, incorporating specific datasets to assess the effectiveness of the proposed metaheuristic-based resource provisioning model for healthcare applications. As elucidated in Chapter 3.4.2, the utilization of a mobility dataset allowed for the simulation of dynamic device movements, mirroring real-world scenarios where devices exhibit mobility in federated learning environments. Additionally, the integration of an ECG dataset facilitated the emulation of authentic healthcare data, providing a realistic foundation for evaluating the performance of our federated learning system. Chapter 3.5 meticulously details the crucial parameters considered for evaluation, encompassing energy consumption, cost, latency, execution time, and network usage. These parameters were selected based on their critical relevance to the efficiency and effectiveness of resource provisioning in federated learning for healthcare applications. Our methodical description and utilization of diverse datasets, coupled with the meticulous examination of key evaluation parameters, contribute to the robustness and applicability of our research findings within the realm of Federated Learning for healthcare applications.

5.3.2 Simulation Environment

This section explains the simulation environment used in evaluating the proposed approach. The sensors detect the ECG of the patient and send the data to the edge/fog nodes regularly. Data is processed and analyzed on the edge/fog nodes to determine whether the patient's health status is normal or critical. The results are subsequently sent to the cloud and the patient's smartphone for storage. The edge/fog nodes' connection to the cloud server is established through the proxy server. The client module is integrated in IoT devices to get sensor data. The processing module is embedded in edge/fog nodes to process and analyze the incoming data in order to assess the patient's health status. The edge/fog node then communicates the results to the associated IoT device, which displays them. It must define values for numerous parameters in iFogSim2 when generating edge/fog devices, such as CPU length, RAM, Bandwidth, and so on.

Edge/Fog devices are the computational devices in iFogSim2. Computational gadgets, on the other hand, come in various levels. The parent node acts as a Cloud server and is placed on Level 3. The Fog nodes are connected to the Cloud server via a proxy server at Level 2. Fog nodes are closer to the user at



(a) Normal ECG (b) Anomalous ECG

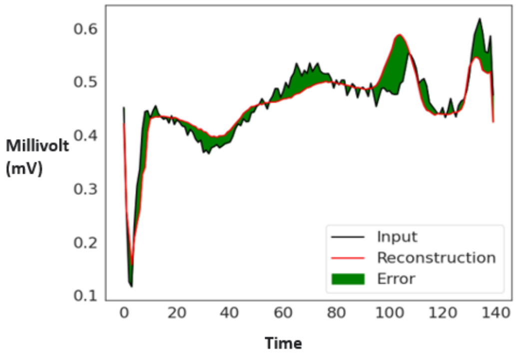
Figure 5.6: Normal ECG and anomalous ECG plot

Level 1, which is considered as Edge device, giving more frequent computational and storage capacities. Sensors and actuators are used in Level 0 IoT devices. The MicroserviceFogDevice, Sensor, and Actuator classes of iFogSim2 simulate the physical topology. The layering architecture of the iFogSim2 simulator is described with diagram in Section 4.3. The scenarios were simulated on an Intel Core i7 CPU running at 1.80 GHz and 4GB of RAM. The fractional selectivity of the input-output relationship inside a module is set to 1.0.

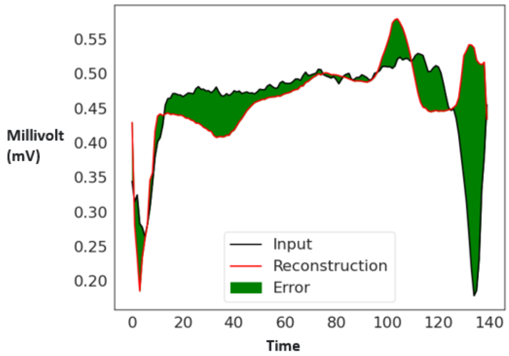
5.3.3 Analysis and Observations

This section presents the results and the observation. The model is evaluated as described in the previous sections for varying placement policies. Figure 5.6 presents the normal and abnormal ECG samples. Figure 5.7 shows the reconstructed normal and abnormal ECG plots. The reconstructed ECG helps in predicting whether the ECG is anomalous. The reconstructed one with the error beyond a threshold is considered anomalous. The error calculated from these figures helps in this classification. Figure 5.8 highlights the training and the testing loss graphically.

Figure 5.9 compares the identified performance parameters for different placement policies discussed in this work. It could be observed that the deployment of the FL module in the Edge layer reduces the Cloud energy consumption by 2% with a decrease in network use of 32%. This, in turn, reduces the cost by

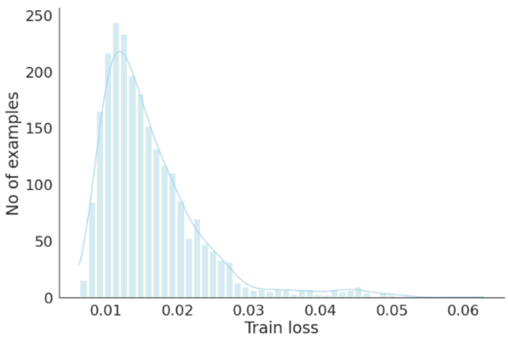


(a) Reconstructed Normal ECG

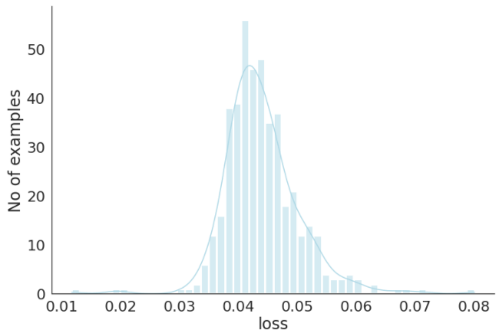


(b) Reconstructed anomalous ECG

Figure 5.7: Reconstructed normal and anomalous ECG plot

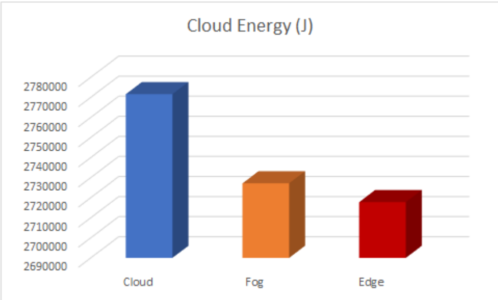


(a) Train loss

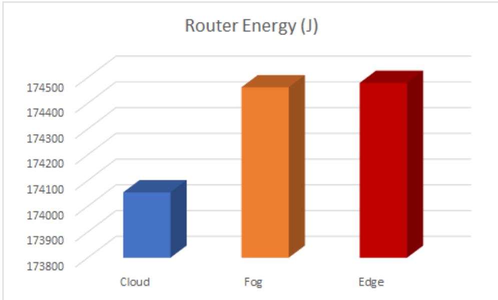


(b) Test loss

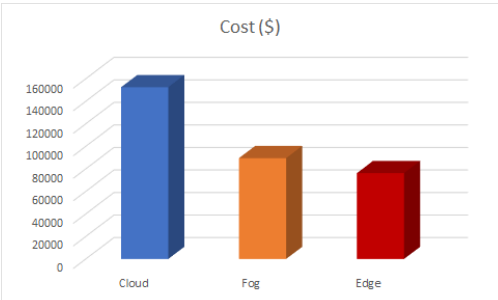
Figure 5.8: Train and test loss graphs for FedSDM



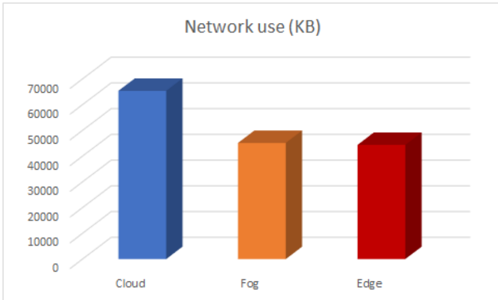
(a) Cloud Energy



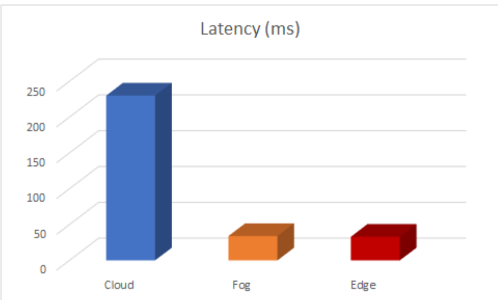
(b) Router Energy



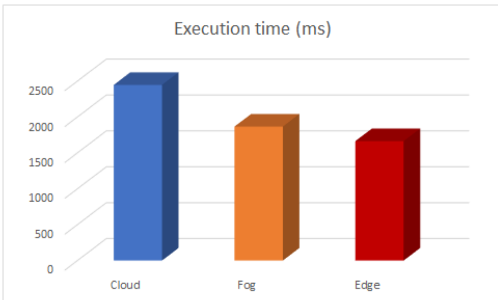
(c) Cost



(d) Network use



(e) Latency



(f) Execution time

Figure 5.9: Result of FL deployment in Edge/Fog/Cloud layers

50%, the execution time by 32%, and the latency by 86%. All the above comparisons are against the placement of the FL module in the Cloud layer. While analyzing the results of placement of the FL module in the Fog layer against the Cloud layer, Cloud energy consumption decreases by 2%, network use by 31%, cost by 41%, execution time by 23%, and latency by 85%. Table 5.1 presents the above discussions in a consolidated manner for better understanding of the results. However, the router energy consumption is found to be more (i.e.) 2.3% and 2.4% for FL module deployment in the Edge, and Fog layers since more computations are performed in those layers. A similar comparison of placing the FL module in Edge and Fog yields a performance increase of 0.3%, 2%, 15%, 11%, and 3% for energy consumption, network usage, cost, execution time, and latency, respectively as presented in Table 5.2. Table 5.3 shows the number of simulations conducted and the average results for each of the parameters. In conclusion, FL module deployment in the Edge layer is superior to FL module deployment in Fog or Cloud, which adds to the fact that the integration of AI on Edge enables smart healthcare systems. This could also support real-time or advanced remote patient monitoring by immediately processing the clinical tests.

Table 5.1: Comparison of Edge and Fog FL placement against Cloud

Metric	**Edge FL placement**	**Fog FL placement**
Energy Consumption (J)	2%	2%
Network Use (KB)	32%	31%
Cost ($)	50%	41%
Latency (ms)	86%	85%
Execution Time (ms)	32%	23%

Table 5.2: Comparison of Edge FL placement against Fog

Metric	**Edge FL placement**
Energy Consumption (J)	0.3%
Network Use (KB)	2%
Cost ($)	15%
Latency (ms)	11%
Execution Time (ms)	3%

Table 5.3: Number of simulations and the mean parameter values for the implementation of Edge FL compared to Fog FL implementation

Metric	Number of Simulations					Average
	1	2	3	4	5	
Energy Consumption (J)	0.34%	0.37%	0.30%	0.25%	0.36%	0.3%
Network Use (KB)	1.6%	1.2%	1.3%	1.5%	2%	1.5%
Cost ($)	14.8%	11.3%	12.7%	13.7%	12%	12.9%
Latency (ms)	2.9%	1.5%	3.4%	3.3%	1.7%	2.5%
Execution Time (ms)	10.8%	10.1%	11.3%	11%	9.6%	10.5%

Comparative analysis

In order to have an effective conclusion, the proposed approach has been compared for its accuracy and the training loss parameters against the existing results presented in the literature [112]. Figure 5.10 and Figure 5.11 show the contrast of the parameters used for various batch sizes and epochs. These results also prove the conclusion statement in the previous sub-section.

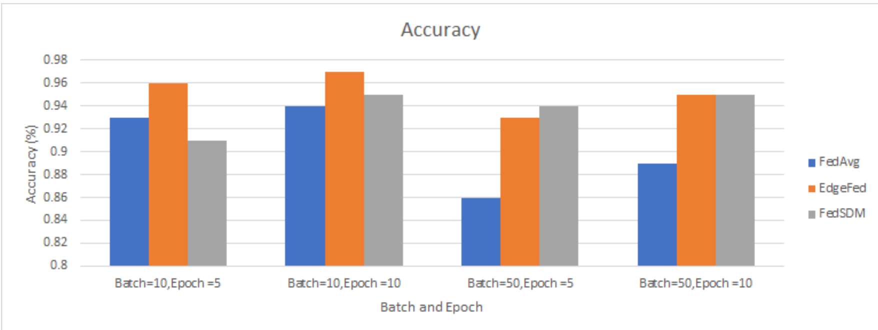


Figure 5.10: Accuracy comparison of FedSDM with FedAvg and Edgefed

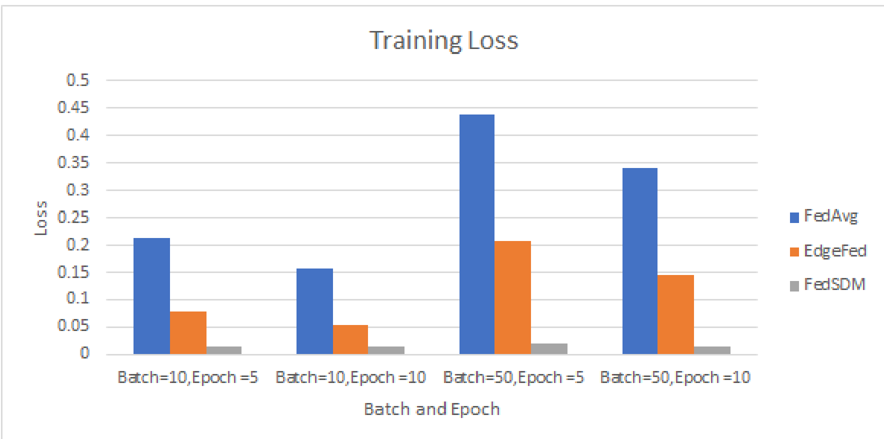


Figure 5.11: Training loss comparison of FedSDM with FedAvg and Edgefed

5.4 Summary

Due to the heterogeneous and dynamic nature of critical medical IoT applications in fog scenarios, the privacy of patients become a crucial problem. This chapter investigates the federated learning-based smart decision making module for ECG Data in microservice-based IoT medical applications. In addition, we also examine the performance of the proposed system with three different placement policies considering the deployment at edge, fog and cloud layers.

The deployment of federated learning in edge, fog, and cloud-based medical applications brings about several challenges that necessitate innovative solutions. One of the primary concerns is the preservation of patient data privacy and security. In healthcare, sensitive medical records and diagnostic information are involved, making it imperative to maintain the confidentiality of patient data. Federated learning mitigates this risk by training machine learning models directly on patient devices, but it introduces complexities in ensuring that data remains secure throughout the collaborative training process.

Moreover, the distributed nature of edge, fog, and cloud computing environments amplifies the challenge of maintaining data integrity and traceability. Medical professionals and regulatory bodies require a verifiable record of data access and usage for compliance and accountability purposes. Without a robust mechanism to track and audit data interactions, it becomes difficult to maintain trust in the federated learning system. Additionally, federated learning systems must contend with the dynamic and heterogeneous nature of edge and fog devices, which can vary significantly in terms of computational capabilities and network connectivity. Ensuring efficient model updates and synchronization across these diverse endpoints is a technical hurdle that must be addressed to achieve seamless and timely training.

In light of these challenges, the integration of blockchain technology emerges as a promising solution. Blockchain offers a secure, tamper-proof ledger that can enhance data security, privacy, and traceability in federated learning systems. It provides a transparent and immutable record of all data transactions, ensuring that patient data remains confidential, unaltered, and auditable throughout the federated learning process. Blockchain can also facilitate trust among the various stakeholders involved, including medical professionals, patients, and regulatory bodies, ultimately enabling the adoption of federated learning for more accurate medical diagnoses and treatment recommendations in edge, fog, and cloud-based healthcare applications.

Incorporating blockchain into our proposed approach for federated learning within edge, fog, and cloud-based medical applications represents a crucial step towards achieving AI-driven healthcare solutions that are not only more precise but also more secure and transparent. This integration enhances the potential for improved medical diagnoses, treatment recommendations, and overall patient care outcomes. Therefore, we are committed to integrate blockchain into our approach to meet these essential objectives, which is the focus of the next chapter.

Chapter 6

Resource Provisioning based on Blockchain Integrated Federated Learning Methods

6.1 Introduction

The integration of IoT and healthcare, commonly known as medical IoT, has witnessed remarkable growth in recent years, offering unprecedented opportunities for real-time patient monitoring and personalized healthcare. Among the numerous medical IoT applications, ECG anomaly detection plays a vital role in identifying potential cardiovascular irregularities and assisting in timely diagnosis and treatment. Accurate ECG anomaly detection is crucial for preventing life-threatening conditions and improving patient outcomes. Traditional approaches to ECG anomaly detection often involve centralizing medical data in cloud-based infrastructures, where machine learning models are trained and updated. However, this centralized paradigm raises significant concerns regarding data privacy, security, and latency. Medical data, being sensitive and highly regulated, requires stringent protection to comply with data privacy laws and maintain patient trust. Moreover, the reliance on a centralized server introduces potential points of failure, making the system vulnerable to cyber-attacks and data breaches. By keeping data decentralized and processing it locally, federated learning enhances data privacy and security, ensuring that sensitive medical information remains with the users who generate it. This decentralized approach minimizes the risk of data breaches and facilitates compliance with data protection regulations. While federated learning addresses many privacy concerns, it still faces certain limitations, especially in resource-constrained IoT environments. Communication overhead, limited computational capabilities of edge devices, and the potential for

Byzantine attacks require further enhancements for effective federated learning in medical IoT applications. In this context, blockchain technology emerges as a complementary solution to enhance the security, transparency, and efficiency of federated learning in medical IoT environments. Blockchain, as a decentralized and tamper-proof ledger, provides an immutable record of all transactions and model updates. When combined with federated learning, blockchain ensures the integrity and transparency of data and model updates, creating a distributed and trustless environment for collaborative training [116, 190–194].

The motivation behind this research is to explore the potential of blockchain-based federated learning for ECG anomaly detection in edge, fog, and cloud computing environments. By leveraging blockchain's immutability and decentralization, we aim to address the privacy and security concerns associated with medical data while enabling efficient and accurate ECG anomaly detection. This implementation seeks to empower medical IoT devices, edge, and fog nodes to collaboratively participate in the training process, leading to improved model accuracy and robustness. This research can contribute to the advancement of healthcare services by presenting a cutting-edge solution that guarantees data privacy, security, and efficient model training. The integration of blockchain and federated learning in medical IoT applications holds the potential to revolutionize patient care, enabling more personalized and timely medical interventions while upholding the highest standards of data protection and confidentiality. The motivation for conducting this study stems from the fact that existing research on Edge/Fog/Cloud Federated learning approaches does not adequately address the healthcare issues related to ECG anomaly detection in microservice-based IoT healthcare applications

6.2 Proposed Model

Based on the comprehensive literature review presented in Section 2.7, which examines existing methodologies and frameworks, highlighting the integration of IoT microservices, mobility management, and blockchain-based federated learning techniques for healthcare applications, a novel approach has been introduced. The proposed approach adopts a strategy centered around federated clients. Within this framework, every client is furnished with a pre-existing model from the federated server. This model is curated using a public data set, which serves as a foundational resource for initializing the training procedures of individual clients. By leveraging the knowledge encapsulated within this initial model, federated clients can then embark on personalized training based on their local data, ultimately contributing to the collective learning process in a distributed and privacy-preserving manner. This distributed architecture enable clients to harness the power of col-

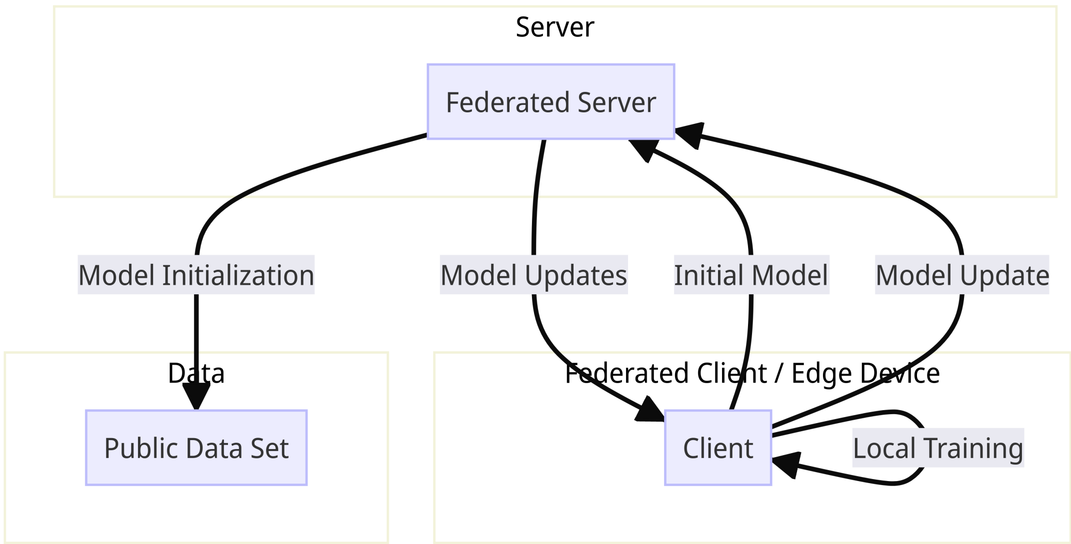


Figure 6.1: Proposed Federated learning architecture

lective data while maintaining data privacy on their local devices. Leveraging this access, each federated client autonomously undertakes model training using its own local data, tailoring the model to its unique circumstances and requirements.

Upon the local training phase's completion, the federated clients transmit their model updates back to the federated server. This communication process facilitates information exchange, allowing the federated server to consolidate the various local models into a unified and robust global model. Aggregating insights from diverse data sources enhances the overall model's accuracy and adaptability. The configuration for the suggested federated learning approach has been illustrated in Figure 6.1.

Notably, the federated server employs a mining process that involves the active participation of all federated clients and edge devices. Smart contracts play a crucial role in this stage, ensuring the mining process's transparency, security, and fairness. This collaborative mining process is vital for producing a reliable and trustworthy global model that caters to the collective needs of the federated ecosystem. To ensure tamper-proof storage and easy accessibility, the resulting global model is securely stored inside a blockchain. The immutable nature of blockchain technology ensures the integrity of the model and enable efficient retrieval whenever needed. This integration of federated learning with blockchain offer a cutting-edge solution that combines privacy, decentralization, and reliability, fostering a new paradigm for collaborative and secure machine learning in diverse applications. The configuration of the proposed blockchain based federated learning approach has been visually presented in Figure 6.2.

The described approach functions in the following manner: Patient-generated

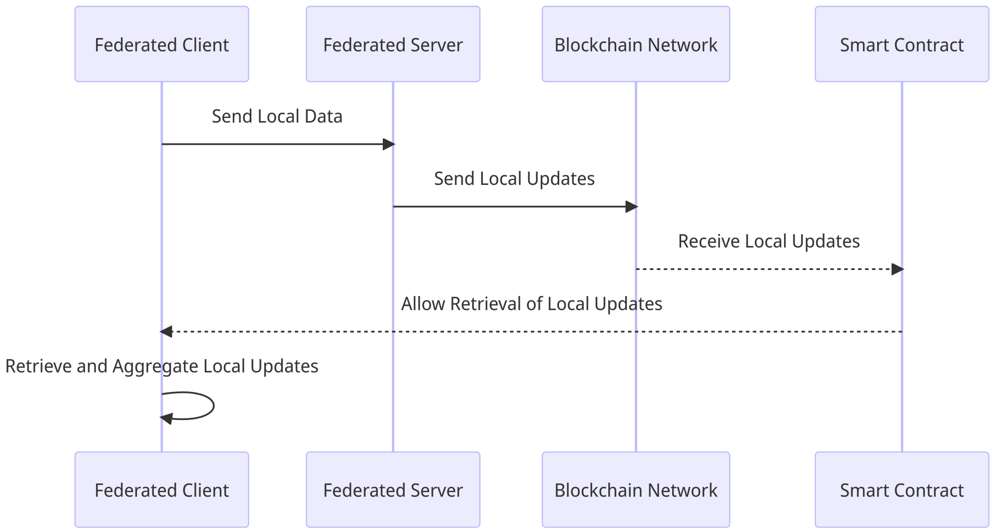


Figure 6.2: Proposed approach Sequence diagram

ECG sensor data is stored on the edge device, for instance, a mobile phone. Computation tasks are managed by the client and data preprocessing microservices, respectively, both located at the edge. Subsequently, the preprocessed ECG data is transferred to the Smart Decision Making module for anomaly analysis. In the event of anomaly detection, a notification is sent to the end device to inform the user of potential health concerns.

The presented architecture assesses and contrasts the effectiveness of various placement strategies for the decision-making module built on federated learning, as outlined in the preceding section. In every placement circumstance, be it at the edge, fog, or cloud, the individual updates from the associated devices or nodes are consolidated within their respective tiers. Following each round of federated learning, the clients and edge devices collaborate in the data extraction process and engage in the execution of smart contracts.

The key functions of smart contract are "updateGlobalModel" and "getGlobalModel." The former allows for updating the global model with a new model represented as an array, ensuring that the new model's length matches the existing global model's length to prevent invalid updates. The latter function enables clients to retrieve the latest version of the global model. This smart contract is intended to be used in conjunction with the federated learning system, where a federated server generates a new global model after each round of training, and clients interact with the smart contract using web3.js to obtain the global model for their local training, enabling collaborative machine learning while preserving data privacy on the blockchain. Consequently, the globally generated model is securely stored on the Ganache blockchain and is treated as a transaction. The integration of blockchain technology enhances the system's resilience, facilitating smooth cooperation among participants while safeguarding data privacy and preventing any unauthorized alterations to the global model. Smart contract algorithm proposed is presented in Algorithm 4.

Algorithm 4 Smart Contract for Global Model

procedure INITIALIZE
    Initialize the global model in the smart contract with zeros in the constructor.
end procedure
procedure UPDATEGLOBALMODEL(**input** newModel: array)
    if newModel.length == globalModel.length **then**
        require newModel.length == globalModel.length, "Invalid model size"
        globalModel = newModel
    end if
end procedure
procedure GETGLOBALMODEL
    return globalModel
end procedure
call UPDATEGLOBALMODEL(newModel) ▷ Passing the new model parameters as an array
call GETGLOBALMODEL ▷ To retrieve the latest version of the global model
▷ Clients use the retrieved global model for local training and updates in subsequent rounds of federated learning

The proposed method explores three different deployment scenarios for FedAvg in various layers: edge, fog, and cloud. In the edge scenario, the computation and model training takes place on the edge devices closer to the end users. In the fog scenario, the computation occurs on intermediate fog nodes, while in the cloud scenario, the central server handles the computation and aggregation of the global model. Each scenario offers unique advantages and trade-offs in terms of communication efficiency, latency, and resource utilization, allowing the system to adapt and optimize according to the application's specific requirements. Following every round of federated learning, the clients and edge devices actively engage in the mining process and execute smart contracts. As a result, the collectively generated global model is securely stored within the Ganache blockchain, where it is treated as a transaction. This decentralized and immutable ledger ensures the integrity and transparency of the model updates, enhancing the overall security and trustworthiness of the federated learning system. By leveraging blockchain technology, the process becomes more robust, enabling seamless collaboration among the participants while preserving data privacy and preventing unauthorized modifications to the global model.

Each deployment policy is assessed for its learning efficiency and QoS parameters. By comparing the results from various deployment options, the proposed method aims to determine the most effective and efficient approach for integrating federated learning in the healthcare system for ECG anomaly detection in real time. The suggested approach includes applying blockchain based FedSDM in the edge, fog, and cloud layers to assess effectiveness and cost-efficiency.

6.3 Results and Analysis

6.3.1 Blockchain-Based Federated Learning Evaluation parameters and Dataset for Healthcare IoT Microservices

Proposed research employs a comprehensive methodology by incorporating specific datasets to evaluate the effectiveness of the proposed meta-heuristic-based resource provisioning model for healthcare IoT microservices in a blockchain-based federated learning context, as detailed in Chapter 3.4.2. The integration of a mobility dataset enable the simulation of dynamic device movement patterns, emulating real-world scenarios with mobile IoT devices within the blockchain-based federated learning framework. Additionally, the incorporation of an ECG dataset facilitate the emulation of authentic healthcare data, providing a realistic foundation for assessing the performance of our system in healthcare applications within the context of blockchain-based federated learning. Chapter 3.5 meticulously outlines the key parameters considered for evaluation, encompassing energy consumption, cost, latency, execution time, and network usage, specifically tailored to the blockchain-based federated learning paradigm. These parameters were carefully selected due to their critical relevance in determining the efficiency and effectiveness of resource provisioning for healthcare IoT microservices within the blockchain-based federated learning framework. Our detailed description of the dataset utilization and the examination of key evaluation parameters contribute to the robustness and practical applicability of our research in the context of healthcare IoT microservices within the blockchain-based federated learning environment.

6.3.2 Simulation Environment

iFogSim2

This section presents the simulation environment utilized to assess the proposed approach. The sensors are responsible for detecting the patient's ECG data, which is then regularly transmitted to the Fog nodes. On the edge/fog nodes, the data undergoes processing and analysis to determine the patient's health status, whether it is normal or critical. The outcomes are subsequently transmitted to both the Cloud for storage and the patient's smartphone. To establish the connection between the Fog nodes and the Cloud server, a proxy server is employed.

To obtain sensor data, the client module is integrated into IoT devices. On the other hand, the processing module is embedded in the edge/fog nodes, enabling them to process and analyze the incoming data for the assessment of the patient's health status. Once the analysis is complete, the edge/fog node communicates the results to the associated IoT device, which displays them to the user. During the generation of Fog devices in iFogSim2, various parameters need to be defined, such as CPU length, RAM, Bandwidth, and more.

In summary, this simulation environment facilitates the evaluation of the proposed approach's performance by simulating the flow of ECG data from sensors to edge/fog nodes, cloud storage, and end-user devices. It allows for the examination of various system configurations and parameter values to assess the efficiency and effectiveness of the proposed system in processing and analyzing real-time critical healthcare data.

In the iFogSim2 simulation, computational devices are categorized into Fog devices, and they are available at various levels. The highest level, Level 3, represents the parent node, which functions as the Cloud server. At Level 2, the Fog nodes are connected to the Cloud server through a proxy server. These Fog nodes, situated at Level 1, are closer to the end-users and are considered Edge devices. They offer more frequent computational and storage capabilities. At the lowest level, Level 0, IoT devices are equipped with sensors and actuators. A thorough explanation of the aforementioned is provided in section 4.3.

In iFogSim2, the physical topology is simulated using the MicroserviceFogDevice, Sensor, and Actuator classes. The scenarios are conducted on a computer system with an Intel Core i7 CPU running at 1.80 GHz and 4GB of RAM. The fractional selectivity of the input-output relationship within a module is set to 1.0.

This configuration enables the simulation of the proposed system's behavior across different levels of computational devices, from edge to fog and cloud servers, with realistic processing capabilities and communication links. The simulation is conducted on a standard computer setup, allowing for comprehensive evaluations of performance and efficiency under various scenarios.

Ganache

Ganache is a popular personal blockchain designed specifically for Ethereum development and testing purposes. It serves as a local and private Ethereum network, enabling developers to deploy, interact, and debug their smart contracts without the need for real transactions on the main Ethereum network. One of its essential functionalities is providing a local blockchain environment, which significantly speeds up development and testing cycles. Ganache comes with predefined accounts containing test Ether, facilitating the testing of various scenarios and functionalities in smart contracts. interaction with the local blockchain through the RPC interface using JSON-RPC or web3.js, making it easy to integrate with smart contract development tools and libraries. Although Ganache is a local network, it simulates gas prices and limits, giving developers insights into how their smart contracts would perform on the main Ethereum network. Another crucial feature is the ability to take snapshots of the blockchain state and later reset it, allowing for easy testing and debugging with different initial states. Ganache also offers transaction tracing and logs, enabling detailed analysis and debugging of smart contract executions. Integrated seamlessly with the Truffle development framework, Ganache provides Ethereum developers with a convenient and efficient environment for local testing, making smart contract development a smoother and more enjoyable experience [195]

In our proposed approach, Ganache is set up with port 7545 selected for accessibility through 'localhost' or '127.0.0.1'. A smart contract was then created using Truffle, a widely used development framework that streamlines the building and deployment of smart contracts. The contract was meticulously compiled to ensure its precision and efficiency. The smart contract was successfully deployed to the Ganache blockchain, leveraging Truffle's migration scripts, making it readily available and operational on the network. The web3.js library, a powerful JavaScript framework, was utilized to facilitate interaction with the deployed smart contract. This integration with web3.js enabled seamless communication with the smart contract, empowering the applications to invoke its functions and execute transactions with ease.

6.3.3 Analysis and Observations

In this section, we present the results and observations of the proposed model, which were evaluated using different placement policies, as explained in the preceding sections. These reconstructed ECGs are crucial in predicting whether the ECG readings are anomalous. An ECG is considered anomalous if its reconstruction error surpasses a predefined threshold. The error values calculated from these figures facilitate the classification of ECG readings.

Ganache Interface includes features for account management, network configuration, blockchain data, and logs and events. Users can create, import, and fund Ethereum accounts, configure network settings, and access essential blockchain information. Figure 6.3 provides the snapshot of Ganache Interface of the proposed

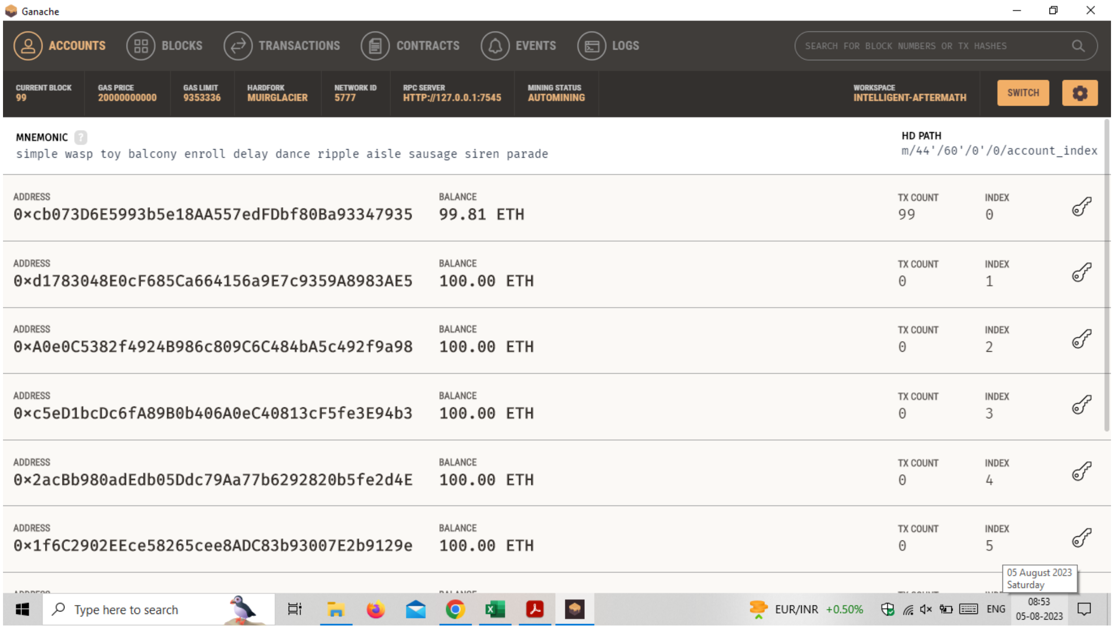


Figure 6.3: Ganache interface

approach.

Within Ganache, the smart contract deployment interface component allows to compile and deploy smart contracts. We compiled the Solidity contracts into bytecode and deployed them to the local or test Ethereum network. This interface includeed a transaction history that records the details of contract deployments, such as transaction IDs and statuses. It has interaction with deployed contracts, facilitating function calls and state inspection. Figure 6.4 provides the snapshot of smart contract deployment interface component of the proposed approach.

The Transactions page in Ganache is dedicated to displaying a comprehensive transaction history. It provides a chronological record of all transactions on the blockchain, offering insights into their senders, receivers, gas costs, and timestamps. Users can conveniently access transaction details by clicking on specific transactions, revealing critical information like transaction hashes, block numbers, and input data. This page is invaluable for tracking the flow of transactions and their associated data. Figure 6.5 provides the snapshot of transactions of the proposed approach.

The created blocks page in Ganache operates as a block explorer, offering insights into each block added to the blockchain. It provides details about individual blocks, including their block numbers, timestamps, gas used, and the transactions included within each block. Additionally, users can access information regarding the mining process, including the current miner's address. This page plays a pivotal role in helping developers monitor the blockchain's structure and understand the relationships between blocks and transactions. Figure 6.6 provides the

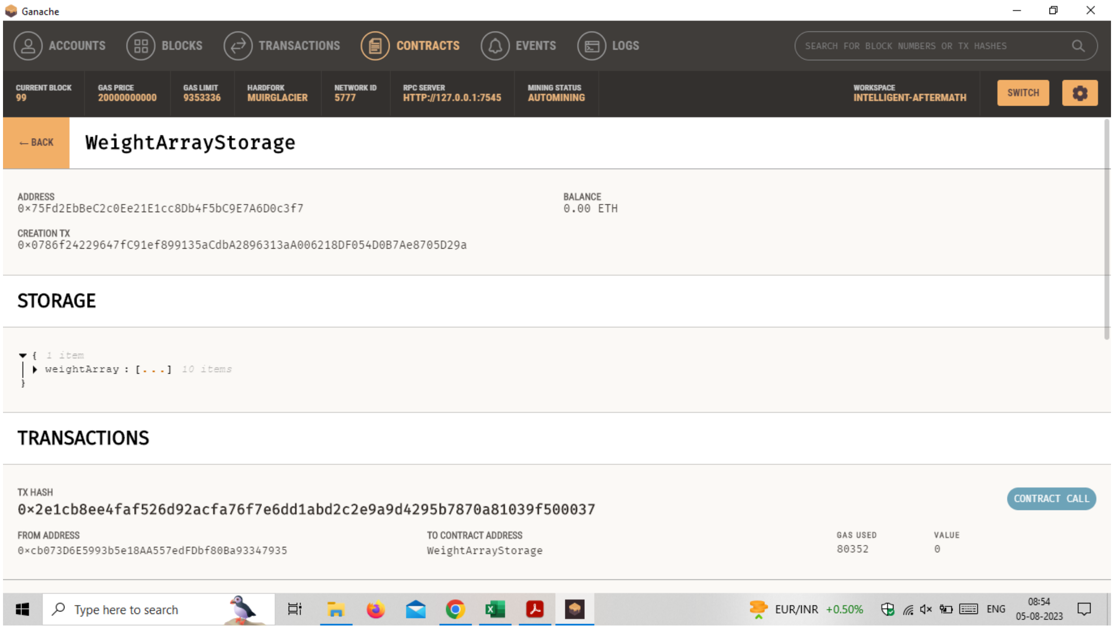


Figure 6.4: Smart contract deployment

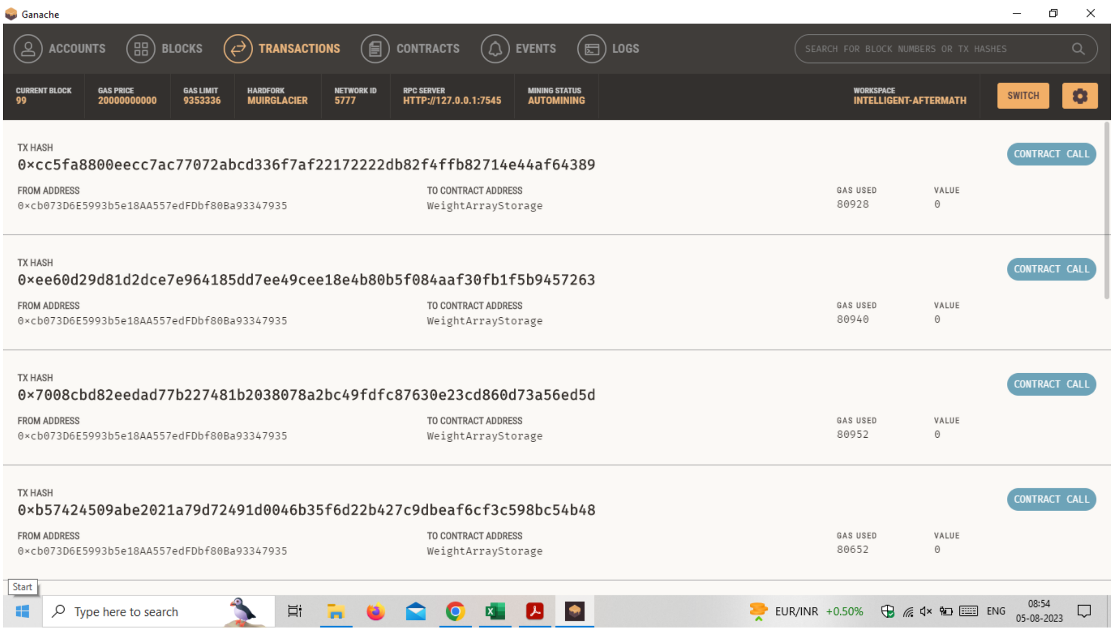


Figure 6.5: Transactions

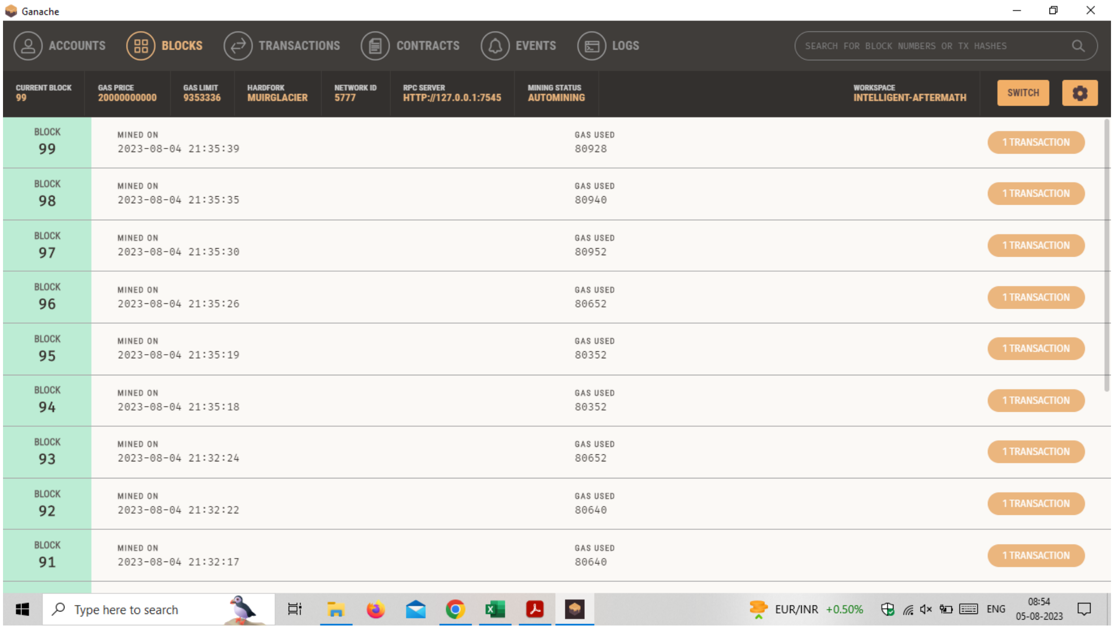


Figure 6.6: Created blocks

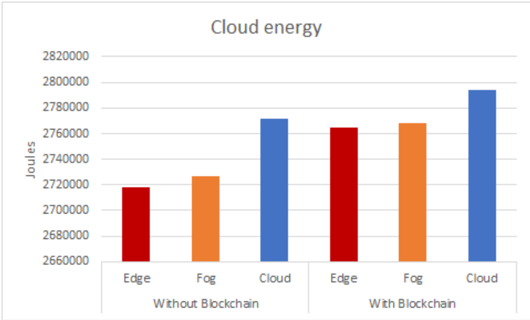
snapshot of details of the created blocks in the proposed approach.

The results of the BCFL model has been compared with the results obtained from the FL model without Blockchain which is diagrammatically presented in Figure 6.7. Figure 6.8 comprehensively compares the performance parameters for different placement policies discussed in this study. When the FL module is deployed in the Edge layer, significant improvements are observed compared to placing it in the Cloud layer. Specifically, deploying FL in the Edge layer reduces Cloud energy consumption by 1%, decreases network usage by 32%, cuts down costs by 23%, reduces execution time by 40%, and decreases latency by 80%.

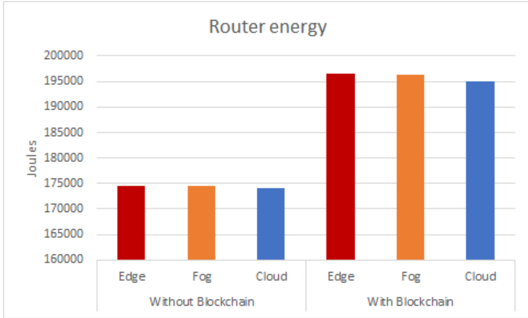
Similarly, when comparing the FL module placement in the Fog layer against the Cloud layer, significant improvements are also observed. Deploying FL in the Fog layer reduces Cloud energy consumption by 0,9%, decreases network usage by 31%, lowers costs by 20%, reduces execution time by 28%, and decreases latency by 79%.

Furthermore, a comparison between FL module placement in the Edge and Fog layers reveals that Edge deployment outperforms Fog deployment. Specifically, Edge deployment shows improvements of 0.1%, 1.1%, 3%, 16%, and 1% in terms of energy consumption, network usage, costs, execution time, and latency, respectively.

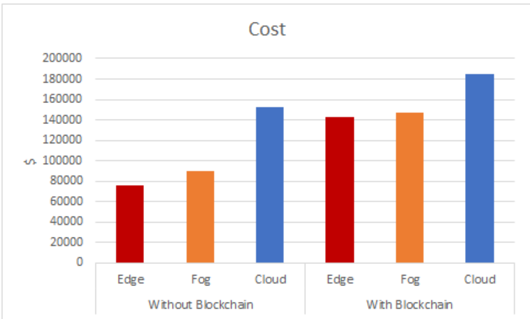
Table 6.1 and 6.2 consolidate the results and provide a clearer understanding of the comparisons made. The graph illustrating the Latency-Execution time composite for Edge BCFL, Fog BCFL, and Cloud BCFL is depicted in Figure 6.9. In conclusion, deploying the FL module in the Edge layer proves to be superior to



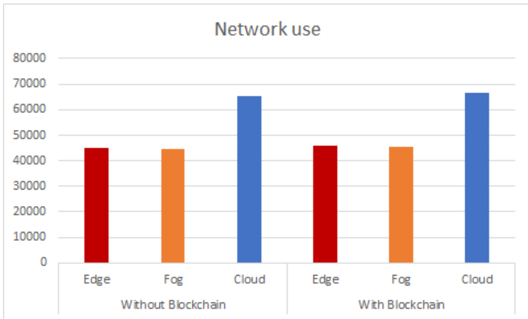
(a) Cloud Energy



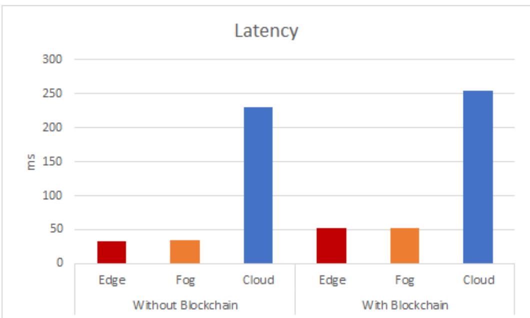
(b) Router Energy



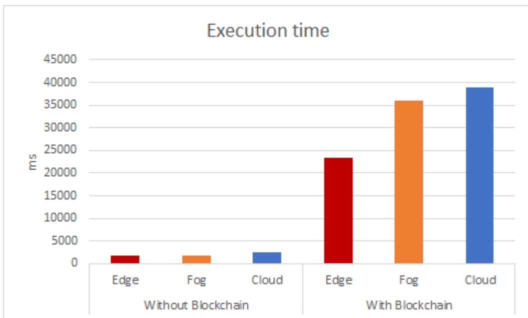
(c) Cost



(d) Network use

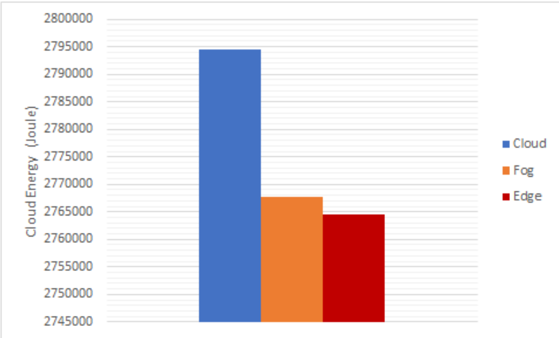


(e) Latency

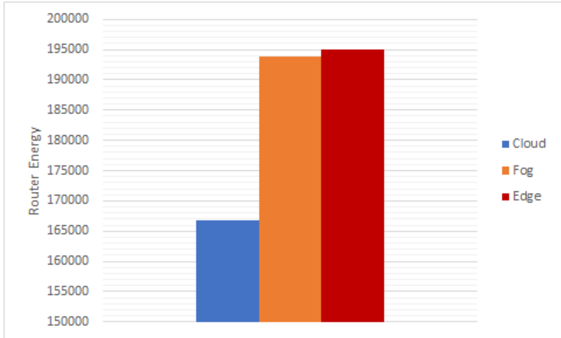


(f) Execution time

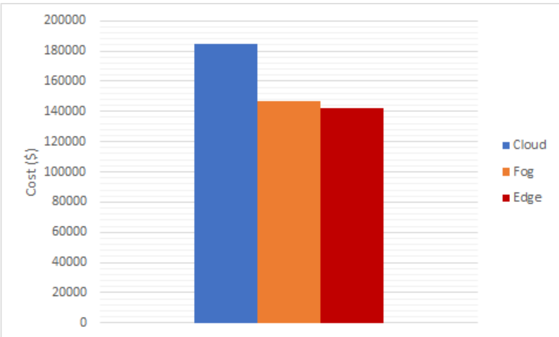
Figure 6.7: Result of BCFL deployment in Edge/Fog/Cloud layers with and without Blockchain integration



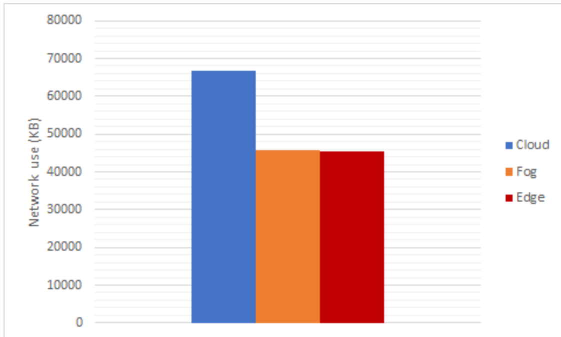
(a) Cloud Energy



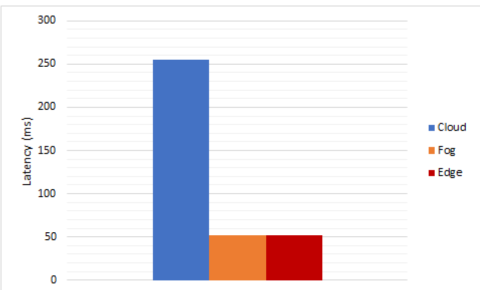
(b) Router Energy



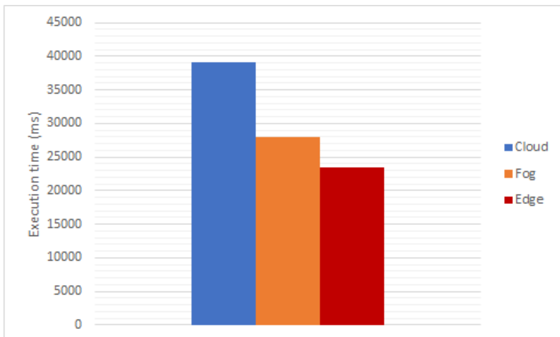
(c) Cost



(d) Network use



(e) Latency



(f) Execution time

Figure 6.8: Result of BCFL deployment in Edge/Fog/Cloud layers

both the Fog and Cloud layer deployments, supporting the integration of AI on Edge for efficient and smart healthcare systems. It enables real-time or advanced remote patient monitoring by immediately processing clinical tests.

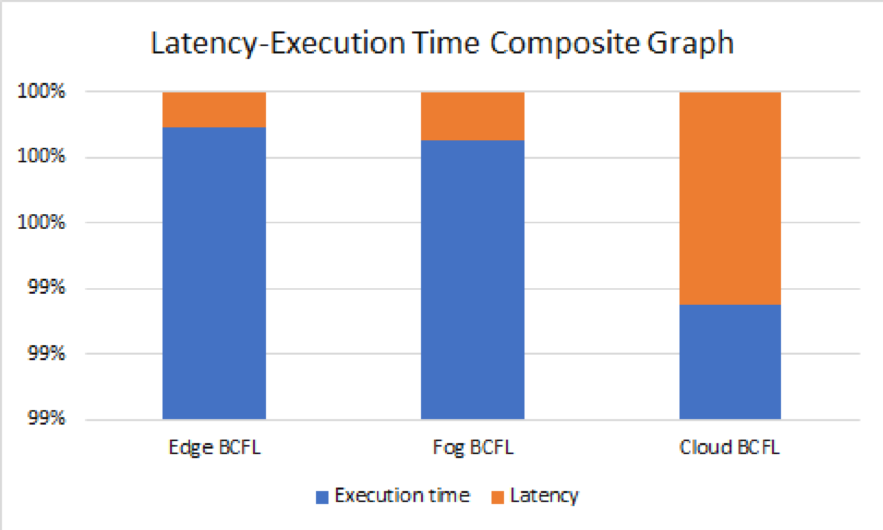


Figure 6.9: Latency-Execution Time Composite Graph

Table 6.1: Comparison of Edge and Fog FL placement against Cloud

Metric	Edge FL placement	Fog FL placement
Energy Consumption (J)	1.07%	0.95%
Network Use (KB)	32%	31%
Cost ($)	23%	20%
Latency (ms)	80%	79%
Execution Time (ms)	40%	28%

Table 6.2: Comparison of Edge FL placement against Fog

Metric	Edge FL placement
Energy Consumption (J)	0.1%
Network Use (KB)	1.1%
Cost ($)	3%
Latency (ms)	1%
Execution Time (ms)	16%

6.4 Summary

Patients' privacy is a significant concern in fog scenarios due to the diverse and dynamic nature of critical medical IoT applications. This study explores a Federated Learning-based Smart Decision Making module for ECG Data in microservice based IoT medical applications. Additionally, the system's performance is evaluated using three different placement policies, considering deployment at the edge, fog, and cloud layers. Future research will address the limitations of this work and focus on experimenting with the model's energy usage. The proposed method will be implemented, and additional aggregation techniques will be explored and deployed to enhance prediction models. Moreover, blockchain techniques will be leveraged to enhance system security in real time edge/fog/cloud scenarios.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

In conclusion, this thesis has delved into the intricate realm of resource provisioning for critical ECG medical applications, employing both the weighted sum method and heuristic methods within the context of edge, fog, and cloud computing. Throughout our exploration, we have highlighted the significance of resource allocation precision in ensuring the uninterrupted functionality and reliability of such life-critical applications. The utilization of the weighed sum method has demonstrated its efficacy in achieving optimal resource allocation by considering multiple factors such as latency, throughput, and energy consumption. This approach has allowed for a more comprehensive and balanced distribution of resources, which is crucial for real-time ECG monitoring and diagnosis. The heuristic methods, on the other hand, have offered practical solutions for resource allocation when dealing with dynamic and unpredictable workloads, enhancing the adaptability of the system to changing conditions.

Furthermore, the extension of this project to integrate federated learning and blockchain technologies showcases the commitment to advancing the capabilities of resource provisioning for ECG medical applications. Federated learning empowers distributed edge and fog nodes to train machine learning models while preserving data privacy collaboratively, ultimately enhancing the accuracy and responsiveness of ECG analysis. Simultaneously, blockchain technology adds an additional layer of security and transparency to the data exchange and resource allocation processes, ensuring the integrity of medical data and resource utilization. This holistic approach enhances the performance of critical ECG medical applications and addresses the paramount concern of data privacy and security in the healthcare domain. It represents a significant step forward in the evolution of edge, fog and cloud computing solutions for healthcare, setting the stage for even more sophisticated and robust systems in the future.

In summary, the research presented in this thesis underscores the vital role of resource provisioning methodologies in ensuring the effectiveness and reliability of ECG medical applications. Incorporating weighed sum and heuristic methods, alongside the integration of federated learning and blockchain technologies, is a testament to the commitment to advancing healthcare technology. As we move forward, it is imperative to continue exploring innovative approaches to resource provisioning and data security, with the ultimate goal of enhancing patient care and well-being through cutting-edge technology.

7.2 Summary of Research Findings

Throughout this thesis, we have systematically investigated several key research questions pertaining to the effective utilization and optimization of IoT microservices, metaheuristic scheduling techniques, Smart Decision Making modules for ECG anomaly detection, Blockchain-based Federated learning in healthcare applications, and resource provisioning solutions in fog and edge computing environments. Here's a summary of our findings and the status of each research question discussed in Chapter 1:

- **Utilization of IoT Microservices for Healthcare Applications:** We have explored effective strategies for utilizing IoT microservices in resource provisioning and mobility management within healthcare applications, highlighting their potential to enhance operational efficiency. This has been accomplished through the implementation of the strategies outlined in Section 3.4.

- **Multiobjective Optimization in Healthcare Applications:** Through the weighted sum method, we have identified and optimized key parameters crucial for healthcare applications, improving outcomes across multiple objectives. The results of this optimization are detailed in Figure 3.12.

- **Metaheuristic Scheduling Techniques for Resource Provisioning:** Modified metaheuristic scheduling techniques have been investigated to enhance resource provisioning efficiency in fog and edge devices, showcasing promising results in optimizing resource allocation. The outcomes of these investigations are presented in Tables 4.3 and 4.4.

- **Design Considerations for Early Warning Systems:** Essential design considerations for developing an early warning system for ECG anomalies using Smart Decision Making modules have been outlined, focusing on real-time detection and response. The results of these considerations, including the comparison of Edge and Fog FL placement against Cloud, and the comparison of Edge FL placement against Fog, are detailed in Tables 5.1 and 5.2, respectively.

- **Integration of Blockchain-based Federated Learning in Healthcare:** We have proposed methodologies to integrate Blockchain-based Federated learning into critical healthcare applications, emphasizing privacy-preserving methods for protecting end-user data. The outcomes of these methodologies are illustrated in Figure 6.7.

- **Placement Policy for Blockchain-based Federated Learning:** The most suitable placement policy for deploying the Blockchain-based Federated learning module across edge, fog, and cloud layers has been identified, ensuring optimal performance within distributed healthcare architectures. From the discussions in Section 6.3.3, it is evident that the edge deployment policy proved to be more effective.

- **Performance Evaluation of Resource Provisioning Solutions:** We have evaluated the performance of our resource provisioning solution under various conditions, assessing energy consumption, network use, cost, execution time, and latency. Deploying BCFL in the Edge layer has resulted in 1% reduction in Cloud energy consumption, 32% decrease in network usage, 23% cut in costs, 40% reduction in execution time, and 80% decrease in latency.

- **Challenges in Implementing IoT Microservices with Mobility Management:** Challenges associated with implementing IoT microservices with mobility management in healthcare applications have been identified, along with potential mitigation strategies. These discussions are extensively covered in Section 4.3 of the study.

- **Validation of Resource Provisioning Solution through Simulation:** The effectiveness of our proposed resource provisioning solution has not yet been validated through simulation experiments. This is primarily due to the absence of existing literature on Blockchain-based Federated learning methods in edge computing, which limits our ability to benchmark and validate our approach against established standards.

- **Impact of Blockchain-based Federated Learning on Healthcare Applications:** We have examined the impact of integrating Blockchain-based Federated learning on the performance and security of healthcare applications in distributed environments, highlighting its potential benefits and challenges. The integration of Blockchain-based Federated Learning introduces additional overheads, yet it significantly enhances security and ensures robust privacy preservation measures.

7.3 Real-World Deployment Challenges

Deploying the proposed Efficient Resource Management Framework for critical healthcare applications in integrated edge-fog-cloud environments using blockchain-based federated learning methods presents several real-world challenges. One of the primary issues is ensuring seamless interoperability between the diverse components of edge, fog, and cloud infrastructures, which often have varying capabilities and standards. Additionally, the integration of blockchain technology, while enhancing security and data integrity, introduces computational overhead that can strain resource-constrained edge devices. Federated learning methods, which rely on decentralized data processing, must contend with inconsistent data quality and connectivity issues across different network nodes. Ensuring data privacy and compliance with stringent healthcare regulations, such as HIPAA, further complicates the deployment. Moreover, achieving real-time performance and reliability in critical healthcare scenarios demands robust fault tolerance and efficient resource allocation mechanisms. Addressing these challenges requires comprehensive testing, optimization, and collaboration with healthcare stakeholders to ensure the framework's efficacy and sustainability in real-world applications.

7.4 Future Directions

This thesis addressed several challenges of resource management for critical healthcare applications in integrated edge/fog/cloud environments using Blockchain-based Federated Learning methods. However, the proposed approach can be further refined by addressing several key issues that require additional investigation. An overview of these future research directions is detailed in the following sections and depicted in Figure 7.1.

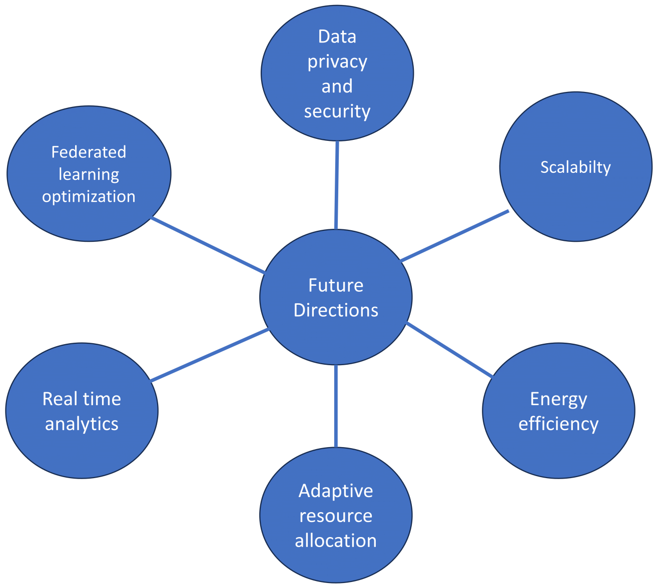


Figure 7.1: Future Directions

7.4.1 Enhanced Data Privacy and Security

The integration of Blockchain technology in Federated Learning offers a promising approach to ensuring data privacy and security. However, there are still challenges related to the computational overhead and energy consumption associated with Blockchain operations. Future research could focus on developing lightweight Blockchain protocols that can be efficiently deployed on resource-constrained devices within the edge/fog/cloud continuum. Additionally, exploring advanced encryption techniques and secure multi-party computation methods can further enhance the privacy and security of sensitive healthcare data.

7.4.2 Scalability and Interoperability

As the number of connected devices and the volume of data in healthcare applications continue to grow, ensuring the scalability and interoperability of the proposed framework becomes crucial. Future work could investigate scalable Federated Learning algorithms that can handle a large number of participants and diverse data sources. Moreover, developing standardized protocols and interfaces can facilitate seamless interoperability between different healthcare systems and platforms, enabling broader adoption of the framework.

7.4.3 Adaptive Resource Allocation

Healthcare applications often have dynamic and unpredictable resource demands. Future research could focus on adaptive resource allocation strategies that can respond in real-time to changing workloads and network conditions. Machine Learning-based predictive models can be employed to forecast resource usage patterns and optimize the allocation of computing resources across the Edge, Fog, and Cloud layers. Additionally, integrating Quality of Service (QoS) metrics can ensure that critical healthcare applications meet stringent performance and reliability requirements.

7.4.4 Energy Efficiency

Energy efficiency is a critical concern in the deployment of resource management frameworks in integrated edge/fog/cloud environments. Future studies could explore energy-aware scheduling algorithms that minimize energy consumption while maintaining high performance. Techniques such as DVFS and energy-efficient networking protocols can be incorporated to achieve energy savings. Furthermore, research into renewable energy sources and energy harvesting technologies can contribute to the sustainability of the proposed framework.

7.4.5 Real-Time Analytics and Decision Support

The ability to perform real-time analytics and provide decision support is essential for critical healthcare applications. Future research could investigate the integration of real-time data processing frameworks and advanced analytics platforms within the proposed resource management framework. This includes developing efficient data aggregation and processing pipelines that can handle streaming data from medical devices and sensors. Additionally, incorporating decision support systems that leverage AI and ML algorithms can assist healthcare professionals in making timely and informed decisions.

7.4.6 Federated Learning Optimization

Federated Learning, while promising, faces challenges such as communication overhead and model convergence. Future work could explore optimization techniques to improve the efficiency of Federated Learning in resource-constrained environments. This includes developing communication-efficient protocols, model compression techniques, and asynchronous training methods. Moreover, personalized Federated Learning approaches can be investigated to tailor models to the specific needs and preferences of individual healthcare providers and patients.

7.4.7 Integration with Emerging Technologies

The proposed framework can benefit from the integration of emerging technologies such as 5G, Internet of Medical Things, and AI. Future research could examine how these technologies can enhance the performance, reliability, and functionality of the resource management framework. For instance, 5G networks can provide ultra-low latency and high bandwidth connectivity, enabling real-time data transfer and processing. Similarly, IoHT devices can offer rich data sources for training Federated Learning models, while AI can enhance predictive analytics and decision support capabilities.

7.4.8 Usability and User Experience

Ensuring the usability and positive user experience of the proposed framework is critical for its adoption in healthcare settings. Future studies could focus on designing user-friendly interfaces and tools that simplify the deployment, management, and monitoring of the resource management framework. Conducting usability testing with healthcare professionals can provide valuable insights into the design and functionality of the framework, leading to improvements that align with the needs and workflows of end-users.

7.4.9 Compliance and Regulatory Considerations

Healthcare applications must comply with stringent regulatory requirements and standards to ensure the safety and privacy of patient data. Future research could investigate the legal and regulatory implications of deploying the proposed framework in different regions and healthcare systems. This includes ensuring compliance with regulations such as the Health Insurance Portability and Accountability Act in the United States and the General Data Protection Regulation in Europe. Developing frameworks and guidelines that address these compliance requirements can facilitate the safe and lawful adoption of the proposed approach.

7.4.10 Enhancing Resilience and Efficiency in Federated Learning and Blockchain Integration

Future work can focus on developing lightweight, optimized algorithms and frameworks to ensure edge devices efficiently handle the computational load imposed by Federated Learning and blockchain processes. Additionally, implementing federated client timeout mechanisms and asynchronous local updates can serve as contingency plans for handling node failures during the Federated Learning process, ensuring robustness and continuity even if some nodes become unresponsive. Strategies for handling federated clients that fail and later rejoin may include storing intermediate updates on the blockchain, allowing the federated server to aggregate these updates once the client reconnects. Fault-tolerant mechanisms such as checkpointing and state synchronization can also ensure that rejoining clients seamlessly continue contributing to the federated learning process without disrupting the overall system.

7.4.11 Final Remarks

In summary, while the proposed Efficient Resource Management Framework for Critical Healthcare Applications in Integrated Edge/Fog/Cloud Environments using Blockchain-based Federated Learning Methods presents a robust solution, further research in these areas will be essential to address existing challenges and harness the full potential of this innovative approach.

Appendix A

Dataset

A.1 Mobility dataset

A.1.1 random_usersLocation-melbCBD_1.csv

The "random_usersLocation-melbCBD_1.csv" dataset within the iFogSim framework contains geospatial information about randomly generated user locations within the Central Business District (CBD) of Melbourne. It includes attributes such as latitude and longitude coordinates, indicating the approximate locations of users within the CBD area. This dataset is valuable for simulating user interactions and mobility patterns in edge computing scenarios, providing essential spatial data for evaluating edge resource allocation and task scheduling strategies [178]. Due to space limitations, below is a condensed sample from the "random_usersLocation-melbCBD_1.csv" dataset:

Latitude	Longitude
-37.81349283	144.9523705
-37.81349283	144.9523705
-37.81349283	144.9523705
-37.81349283	144.9523705
-37.81349283	144.9523705
-37.81349283	144.9523705
-37.81484742	144.9537732
-37.81607696	144.9550465
-37.81571029	144.9566347
-37.81478954	144.9575238
-37.81574041	144.9587857
-37.81628541	144.9597297
-37.81628541	144.9597297
-37.81628541	144.9597297
-37.81628541	144.9597297
-37.81781084	144.9587007
-37.81781084	144.9587007

A.1.2 edgeResources-melbCBD.csv

The "edgeResources-melbCBD.csv" dataset contains spatial information about edge resources located in the Central Business District (CBD) of Melbourne. It includes attributes such as unique identifiers (ID), latitude, longitude coordinates, block information, level, parent relationship, state, and additional details about each resource. This dataset is valuable for simulating edge computing scenarios within urban environments, providing essential geographical and logistical data for deploying and managing edge resources effectively [178]. Due to space constraints, a limited sample from the "edgeResources-melbCBD.csv" dataset is provided below:

ID	Block	Level	Parent	Details
0	0	0	-1	DataCenter
1	1	1	0	Block1 Proxy
2	2	1	0	Block2 Proxy
3	3	1	0	Block3 Proxy
4	4	1	0	Block4 Proxy
5	5	1	0	Block5 Proxy
6	6	1	0	Block6 Proxy
7	7	1	0	Block7 Proxy
8	8	1	0	Block8 Proxy
9	9	1	0	Block9 Proxy
10	10	1	0	Block10 Proxy
11	11	1	0	Block11 Proxy
12	12	1	0	Block12 Proxy
13	12	2	12	Spring and Flinders ucell Optus North West corner Spring and Flinders St MELBOURNE
14	1	2	1	Optus Minicell - Lon_Spencer Corner Spencer and Lonsdale St MELBOURNE
15	11	2	11	136 Exhibition St MELBOURNE
16	9	2	9	Federation Square North -V 164A Flinders Street MELBOURNE
17	3	2	3	KING ST (3144 REPLACEMENT) -V 530 Collins Street MELBOURNE
18	4	2	4	Empire Apartments 402-408 La Trobe Street MELBOURNE
19	10	2	10	CMTS Site 287-293 Exhibition St MELBOURNE
20	1	2	1	CGU Bldg 485 La Trobe Street MELBOURNE
21	6	2	6	360 Collins St Collins Wales Building MELBOURNE
22	3	2	3	Stock Exchange Bldg 530 Collins Street MELBOURNE
23	2	2	2	625 Lt Collins St MELBOURNE
24	12	2	12	ANZ Bank Tower 55 Collins Street MELBOURNE
25	2	2	2	Bourke Place 600 Bourke Street MELBOURNE
26	2	2	2	Marland House 570 Bourke Street MELBOURNE
27	3	2	3	Rialto Towers 525 Collins Street MELBOURNE

A.2 ECG dataset

A.2.1 ecg.csv

This dataset contains electrocardiogram (ECG) readings of patients, with each row representing a complete ECG recording. Each recording consists of 140 data points, with columns 0 to 139 containing the ECG data points represented as floating-point numbers. Additionally, there is a label column indicating whether the ECG reading is classified as normal or abnormal, with categorical values of either 0 or 1 [179]. Given the extensive nature of the dataset, a restricted sample from the "ecg.csv" dataset is presented below:

Column A	Column B	Column C	Column D	Column E	Column EK
-1.1008778	-3.9968398	-4.2858426	1.1196209	-0.17456252	1
-0.56708802	-2.5934502	-3.8742297	0.90422673	-0.55638598	1
0.49047253	-1.9144071	-3.6163638	1.403011	-0.67499495	1
0.80023202	-0.87425189	-2.3847613	1.6143924	-0.98324201	1
-1.5076736	-3.57455	-4.4780109	1.4933655	-0.80992101	1
-0.297161	-2.7666349	-4.1021848	1.2881654	-0.63651154	1
0.44676853	-1.5073974	-3.1874679	0.96121472	-0.83883747	1
0.087630577	-1.7534903	-3.3044731	-0.64868286	-0.75435259	1
-0.83228111	-1.7003675	-2.2573013	1.6792986	-0.76728083	1
0.084430128	-3.1903071	-4.686175	1.6267069	-0.76484136	1
-0.007819138	-2.3367567	-3.527643	1.470211	-1.0828626	1
-1.0743015	-3.2593996	-4.1259793	1.1316334	-0.4515052	1
4.0581274	2.0878442	0.4231153	-1.0382589	-0.79944254	1
-0.76160326	-2.9212433	-3.8943153	1.1621124	-0.63460819	1
-0.18649962	-2.6824878	-4.0168823	2.1514756	-0.71594341	1
0.80393944	-1.1069099	-2.8541073	0.61324167	-0.56830862	1
-0.92021269	-2.4495095	-3.1920276	-0.37083214	-0.22780434	1
2.7446026	-0.10192395	-2.8516809	1.5902704	-0.62244661	1
2.4028692	2.036719	0.34090237	1.9628195	-1.5850003	1
-1.3622631	-3.3530235	-3.9756571	-0.35122193	-0.47888403	1
1.9354139	-0.54876161	-2.3433933	1.8711056	-0.79419029	1
-1.862747	-3.2036327	-3.6341531	-0.302293	-0.20626303	1
-0.99857184	-2.8897248	-3.3877792	1.5209621	-0.8781486	1
-0.28834122	-2.2725454	-3.6660708	1.7227841	-0.94959214	1
0.53974315	0.66577827	-1.2755868	-0.70528326	-0.91614426	1
-1.1812944	-2.3030416	-2.8690724	-0.28587388	-0.5096429	1
1.5319341	0.92006083	-0.49019514	1.8501208	-1.2343389	1
-3.5278988	-5.1176214	-4.6348854	0.80512981	-0.38265832	1
-0.77586665	-3.4141412	-4.0897041	0.44012228	-0.49072884	1
-0.03594174	-0.27819632	-0.38375438	-0.24737942	0.1036615	1
-1.3132285	-1.9628833	-0.84626258	-2.4051707	0.93777282	1
-2.0956639	-4.182035	-4.3009498	0.82778817	-0.3609917	1
-0.65713662	-1.4969218	-2.0119899	1.587308	-0.85828141	1
1.0066612	-0.7129771	-2.1090848	0.7774028	-1.35122	1
0.30348816	-1.7968885	-3.596764	1.0289664	-0.76894357	1
-0.83354058	-1.4633151	-2.2679727	-1.3197323	-0.46750944	1

-0.869182	-2.2065865	-2.9895709	1.8521256	-0.85471294	1
0.1940386	-2.6269766	-4.0946853	1.2951468	-0.87573421	1
1.381994	0.36621607	-2.4019983	-0.15810707	-0.072694063	1
0.8601815	-0.1418624	-1.9673779	0.59271243	-1.1037867	1
-0.50162131	-1.0885841	-1.9397493	2.5894742	-0.59484332	1
-0.53334196	-1.8556681	-2.2140382	1.1994292	-0.81267242	1

References

[1] The Economic Times, "India's population expected to rise for next three decades before starting to decline: Un projections," *The Economic Times*, 2022.

[2] Wikipedia contributors, "Demographics of india," 2023. Accessed: January 19, 2024.

[3] C. Lahariya, "Health & wellness centers to strengthen primary health care in india: Concept, progress and ways forward," *The Indian Journal of Pediatrics*, vol. 87, no. 11, pp. 916–929, 2020.

[4] B. A. Azeem, *Customer shopping behaviour in organised retailing scenario*. PhD thesis, Doctoral dissertation]. https://doi. org/10.13140/RG. 2.2. 14015.36006, 2012.

[5] L. V. Gangolli, R. Duggal, A. Shukla, *et al.*, *Review of healthcare in India*. Centre for enquiry into health and allied themes Mumbai, 2005.

[6] J. A. Razzak and A. L. Kellermann, "Emergency medical care in developing countries: is it worthwhile?," *Bulletin of the World Health Organization*, vol. 80, no. 11, pp. 900–905, 2002.

[7] S. Keenan and J. C. Riesberg, "Prolonged field care: beyond the "golden hour"," *Wilderness & Environmental Medicine*, vol. 28, no. 2, pp. S135–S139, 2017.

[8] World Health Organization, "Cardiovascular diseases," Year. Accessed: August 9, 2024.

[9] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu, "A survey on ecg analysis," *Biomedical Signal Processing and Control*, vol. 43, pp. 216–235, 2018.

[10] S. Banerjee, T. Hemphill, and P. Longstreet, "Wearable devices and healthcare: Data sharing and privacy," *The Information Society*, vol. 34, no. 1, pp. 49–57, 2018.

[11] Innominds, "Emergency Healthcare for the Elderly at Your Fingertips with Remote Health Monitoring Systems." https://www.innominds.com/blog/emergency-healthcare-for-the-elderly-at-your-fingertips-with-remote-health-monitoring-systems, 2024.

[12] G. Aceto, V. Persico, and A. Pescapé, “Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0,” *Journal of Industrial Information Integration*, vol. 18, p. 100129, 2020.

[13] Insider Intelligence, “Wearables: 2023.” https://www.insiderintelligence.com, 2023.

[14] A. Marakhimov and J. Joo, “Consumer adaptation and infusion of wearable devices for healthcare,” *Computers in Human Behavior*, vol. 76, pp. 135–148, 2017.

[15] A. Names, “Heart attack deaths at home delay treatment: Aiims study,” *The Indian Express*, Month Year.

[16] India Today, “Apple Watch saves life of a 36-year-old user suffering from heart condition.” https://www.indiatoday.in/technology/news/story/apple-watch-saves-life-of-a-36-year-old-user-suffering-from-heart-condition-2345300-2023-03-11, March 2023.

[17] NDTV, “Apple Watch Dials 911, Saves Woman’s Life After She Collapsed in Hotel Room.” https://www.ndtv.com/feature/apple-watch-dials-911-saves-womans-life-after-she-collapsed-in-hotel-room-4014270, 2023.

[18] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, “Fog computing: a comprehensive architectural survey,” *IEEE access*, vol. 8, pp. 69105–69133, 2020.

[19] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, “A survey on the edge computing for the internet of things,” *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[20] M. Chernyshev, S. Zeadally, and Z. Baig, “Healthcare data breaches: Implications for digital forensic readiness,” *Journal of medical systems*, vol. 43, pp. 1–12, 2019.

[21] Hindustan Times, “Report flags host of security lapses at AIIMS.” https://www.hindustantimes.com/cities/delhi-news/report-flags-host-of-security-lapses-at-aiims-101670359251255.html, 2024.

[22] HIPAA Journal, “Healthcare Data Breach Statistics.” https://www.hipaajournal.com/healthcare-data-breach-statistics/, YYYY.

[23] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, “Blockchain for healthcare data management: opportunities, challenges, and future recommendations,” *Neural Computing and Applications*, pp. 1–16, 2021.

[24] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, “Federated learning for healthcare: Systematic review and architecture proposal,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[25] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 73–78, IEEE, 2015.

[26] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," *Cloud computing: Principles and paradigms*, pp. 1–41, 2011.

[27] P. Prakash, K. Darshaun, P. Yaazhlene, M. V. Ganesh, and B. Vasudha, "Fog computing: issues, challenges and future directions," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 6, p. 3669, 2017.

[28] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric internet of things services," in *2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1–2, IEEE, 2015.

[29] Yahoo Finance, "Edge computing market size to surpass $18 billion," 2022. Accessed: January 19, 2024.

[30] E. J. Qaisar, "Introduction to cloud computing for developers," in *Proc. of Information Technology Professional Conference (TCF Pro IT), IEEE TCF 2012*, 2012.

[31] Skillmine Technologies, "The future of connected: A deep dive into iot technologies," 2022. Accessed: January 19, 2024.

[32] S. S. Gill, R. C. Arya, G. S. Wander, and R. Buyya, "Fog-based smart healthcare as a big data and cloud service for heart patients using iot," in *International Conference on Intelligent Data Communication Technologies and Internet of Things*, pp. 1376–1383, Springer, 2018.

[33] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in iot-enabled healthcare solutions," in *Proceedings of the 19th international conference on distributed computing and networking*, pp. 1–10, 2018.

[34] S. T. Selvi, C. Valliyammai, and V. N. Dhatchayani, "Resource allocation issues and challenges in cloud computing," in *2014 International Conference on Recent Trends in Information Technology*, pp. 1–6, IEEE, 2014.

[35] S. Padmavathi, P. Soniha, N. Soundarya, and S. Srimathi, "Dynamic resource provisioning and monitoring for cloud computing," in *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–6, IEEE, 2017.

[36] M. Faraji Mehmandar, S. Jabbehdari, and H. Haj Seyyed Javadi, "A dynamic fog service provisioning approach for iot applications," *International Journal of Communication Systems*, vol. 33, no. 14, p. e4541, 2020.

[37] I. Lee, "Big data: Dimensions, evolution, impacts, and challenges," *Business Horizons*, vol. 60, no. 3, pp. 293–303, 2017.

[38] M. Abdel-Basset, N. Moustafa, R. Mohamed, O. M. Elkomy, and M. Abouhawwash, "Multi-objective task scheduling approach for fog computing," *IEEE Access*, vol. 9, pp. 126988–127009, 2021.

[39] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected internet of things," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404–413, 2016.

[40] M. Shakil, A. Fuad Yousif Mohammed, R. Arul, A. K. Bashir, and J. K. Choi, "A novel dynamic framework to detect ddos in sdn using metaheuristic clustering," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3622, 2022.

[41] W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 343–356, 2017.

[42] V. Porkodi, A. R. Singh, A. R. W. Sait, K. Shankar, E. Yang, C. Seo, and G. P. Joshi, "Resource provisioning for cyber–physical–social system in cloud-fog-edge computing using optimal flower pollination algorithm," *IEEE Access*, vol. 8, pp. 105311–105319, 2020.

[43] B. El Khalyly, A. Belangour, M. Banane, and A. Erraissi, "A comparative study of microservices-based iot platforms," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 7, pp. 389–398, 2020.

[44] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software (JSS), ISSN: 0164-1212, Elsevier Press, Amsterdam, The Netherlands*, 2022.

[45] C. Puliafito, D. M. Gonçalves, M. M. Lopes, L. L. Martins, E. Madeira, E. Mingozzi, O. Rana, and L. F. Bittencourt, "Mobfogsim: Simulation of mobility and migration for fog computing," *Simulation Modelling Practice and Theory*, vol. 101, p. 102062, 2020.

[46] M. Shinu and M. Supriya, "Performance comparison of vm allocation and selection policies in an integrated fog-cloud environment," in *International Conference on Ubiquitous Communications and Network Computing*, pp. 169–184, Springer, 2021.

[47] H. Pydi and G. N. Iyer, "Analytical review and study on load balancing in edge computing platform," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 180–187, IEEE, 2020.

[48] B. Zhou and R. Buyya, "Augmentation techniques for mobile cloud computing: A taxonomy, survey, and future directions," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–38, 2018.

[49] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[50] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.

[51] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020.

[52] G. Xu, "Iot-assisted ecg monitoring framework with secure data transmission for health care applications," *IEEE Access*, vol. 8, pp. 74586–74594, 2020.

[53] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[54] H. Wang, T. Liu, B. Kim, C.-W. Lin, S. Shiraishi, J. Xie, and Z. Han, "Architectural design alternatives based on cloud/edge/fog computing for connected vehicles," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2349–2377, 2020.

[55] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B. Yoon, "A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology," *Future Generation Computer Systems*, vol. 129, pp. 380–388, 2022.

[56] N. Wang, J. Zhou, G. Dai, J. Huang, and Y. Xie, "Energy-efficient intelligent ecg monitoring for wearable devices," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 5, pp. 1112–1121, 2019.

[57] D. Agarwal, "Cloud computing- a brief introduction," Jan 2019.

[58] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva, and M. A. Astorga-Vargas, "Tracking the evolution of the internet of things concept across different application domains," *Sensors*, vol. 17, no. 6, p. 1379, 2017.

[59] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *2012 10th international conference on frontiers of information technology*, pp. 257–260, IEEE, 2012.

[60] S. M. Babu, A. J. Lakshmi, and B. T. Rao, "A study on cloud based internet of things: Cloudiot," in *2015 global conference on communication technologies (GCCT)*, pp. 60–65, IEEE, 2015.

[61] C. F. C. Solutions, "Unleash the power of the internet of things," *Cisco Systems Inc*, 2015.

[62] O. Bibani, C. Mouradian, S. Yangui, R. H. Glitho, W. Gaaloul, N. B. Hadj-Alouane, M. Morrow, and P. Polakos, "A demo of iot healthcare application provisioning in hybrid cloud/fog environment," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 472–475, IEEE, 2016.

[63] M. Deepika and R. Srinivasa, "Eamc: using fog computing and rrp for exigency alert service,"

[64] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[65] M. Mishra, S. K. Roy, A. Mukherjee, D. De, S. K. Ghosh, and R. Buyya, "An energy-aware multi-sensor geo-fog paradigm for mission critical applications," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–19, 2019.

[66] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya, "Mobi-iost: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications," *IEEE Transactions on Network Science and Engineering*, 2019.

[67] D. H. Brahmbhatt and M. R. Cowie, "Remote management of heart failure: an overview of telemonitoring technologies," *Cardiac failure review*, vol. 5, no. 2, p. 86, 2019.

[68] M. Prabhu and A. Hanumanthaiah, "Edge computing-enabled healthcare framework to provide telehealth services," in *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pp. 349–353, IEEE, 2022.

[69] H. Nashaat, E. Ahmed, and R. Rizk, "Iot application placement algorithm based on multi-dimensional qoe prioritization model in fog computing environment," *IEEE Access*, vol. 8, p. 111253–111264, 2020.

[70] A. Kumari, V. Kumar, M. Y. Abbasi, S. Kumari, P. Chaudhary, and C. Chen, "Csef: Cloud-based secure and efficient framework for smart medical system using ecc," *IEEE Access*, vol. 8, pp. 107838–107852, 2020.

[71] F. T. Zohora, M. R. R. Khan, M. F. R. Bhuiyan, and A. K. Das, "Enhancing the capabilities of iot based fog and cloud infrastructures for time sensitive events," in *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pp. 224–230, IEEE, 2017.

[72] M. Hindia, T. Rahman, H. Ojukwu, E. Hanafi, and A. Fattouh, "Enabling remote health-caring utilizing iot concept over lte-femtocell networks," *PloS one*, vol. 11, no. 5, 2016.

[73] A. George, H. Dhanasekaran, J. P. Chittiappa, L. A. Challagundla, S. S. Nikkam, and O. Abuzaghleh, "Internet of things in health care using fog computing," in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–6, IEEE, 2018.

[74] A. A. Mutlag, M. K. Abd Ghani, N. a. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare iot systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[75] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Application management in fog computing environments: A taxonomy, review and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–43, 2020.

[76] G. K. Walia, M. Kumar, and S. S. Gill, "Ai-empowered fog/edge resource management for iot applications: A comprehensive review, research challenges and future perspectives," *IEEE Communications Surveys & Tutorials*, 2023.

[77] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–37, 2019.

[78] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management, and evaluation of fog computing systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2494–2516, 2020.

[79] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, "Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 14, p. e5674, 2022.

[80] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, "Resource provisioning in edge computing for latency-sensitive applications," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11088–11099, 2021.

[81] S. Hu, W. Shi, and G. Li, "Cec: A containerized edge computing framework for dynamic resource provisioning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3840–3854, 2023.

[82] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 968–980, 2021.

[83] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware metaheuristic algorithm for industrial-internet-of-things task scheduling problems in fog computing applications," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12638–12649, 2020.

[84] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, "Resource provisioning in edge/fog computing: A comprehensive and systematic review," *Journal of Systems Architecture*, p. 102362, 2021.

[85] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, "Resource provisioning for iot services in the fog computing environment: An autonomic approach," *Computer Communications*, vol. 161, pp. 109–131, 2020.

[86] R. Panwar and M. Supriya, "Dynamic resource provisioning for service-based cloud applications: A bayesian learning approach," *Journal of Parallel and Distributed Computing*, 2022.

[87] D. R. Vemula, M. K. Morampudi, S. Maurya, A. Abdul, M. M. Hussain, and I. Kavati, "Enhanced resource provisioning and migrating virtual machines in heterogeneous cloud data center," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2022.

[88] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, "Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing," *IEEE Access*, vol. 11, pp. 20635–20646, 2023.

[89] S. K. Mishra, D. Puthal, J. J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, 2018.

[90] P. Hosseinioun, M. Kheirabadi, S. R. K. Tabbakh, and R. Ghaemi, "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm," *Journal of Parallel and Distributed Computing*, vol. 143, pp. 88–96, 2020.

[91] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "Fogplan: A lightweight qos-aware dynamic fog service

provisioning framework," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5080–5096, 2019.

[92] P. G. V. Naranjo, E. Baccarelli, and M. Scarpiniti, "Design and energy-efficient resource management of virtualized networked fog architectures for the real-time support of iot applications," *The journal of Supercomputing*, vol. 74, no. 6, pp. 2470–2507, 2018.

[93] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1441–1456, 2022.

[94] M. Adhikari, S. N. Srirama, and T. Amgoth, "A comprehensive survey on nature-inspired algorithms and their applications in edge computing: Challenges and future directions," *Software: Practice and Experience*, vol. 52, no. 4, pp. 1004–1034, 2022.

[95] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, "An improved multi-objective genetic algorithm with heuristic initialization for service placement and load distribution in edge computing," *Computer Networks*, vol. 194, p. 108146, 2021.

[96] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in internet of underwater things networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1775–1789, 2021.

[97] H. Nashaat, E. Ahmed, and R. Rizk, "Iot application placement algorithm based on multi-dimensional qoe prioritization model in fog computing environment," *IEEE Access*, vol. 8, pp. 111253–111264, 2020.

[98] A. Kumari, V. Kumar, M. Y. Abbasi, S. Kumari, P. Chaudhary, and C.-M. Chen, "Csef: cloud-based secure and efficient framework for smart medical system using ecc," *IEEE Access*, vol. 8, pp. 107838–107852, 2020.

[99] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, "Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments," *Future Generation Computer Systems*, vol. 104, pp. 187–200, 2020.

[100] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3710, 2022.

[101] A. Asghar, A. Abbas, H. A. Khattak, and S. U. Khan, "Fog based architecture and load balancing methodology for health monitoring systems," *IEEE Access*, vol. 9, pp. 96189–96200, 2021.

[102] K. Jayasena and B. Thisarasinghe, "Optimized task scheduling on fog computing environment using meta heuristic algorithms," in *2019 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 53–58, IEEE, 2019.

[103] Y. Qiu, H. Zhang, and K. Long, "Computation offloading and wireless resource management for healthcare monitoring in fog-computing based internet of medical things," *IEEE Internet of Things Journal*, 2021.

[104] M. Abdel-Basset, G. Manogaran, A. Gamal, and V. Chang, "A novel intelligent medical decision support model based on soft computing and iot," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4160–4170, 2019.

[105] H. B. Hassen, W. Dghais, and B. Hamdi, "An e-health system for monitoring elderly health based on internet of things and fog computing," *Health information science and systems*, vol. 7, no. 1, pp. 1–9, 2019.

[106] T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing approach for mobility support in internet-of-things systems," *IEEE Access*, vol. 6, pp. 36064–36082, 2018.

[107] A. Bajaj and O. P. Sangwan, "A systematic literature review of test case prioritization using genetic algorithms," *IEEE Access*, vol. 7, pp. 126355–126375, 2019.

[108] P. Yu and Y. Liu, "Federated object detection: Optimizing object detection model with federated learning," in *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, pp. 1–6, 2019.

[109] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[110] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[111] E. Sannara, F. Portet, P. Lalanda, and V. German, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, 2021.

[112] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "Edgefed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.

[113] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[114] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[115] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, 2021.

[116] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, 2021.

[117] R. Saha, S. Misra, and P. K. Deb, "Fogfl: Fog-assisted federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, 2020.

[118] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10782–10793, 2020.

[119] C. W. Zaw, S. R. Pandey, K. Kim, and C. S. Hong, "Energy-aware resource management for federated learning in multi-access edge computing systems," *IEEE Access*, vol. 9, pp. 34938–34950, 2021.

[120] T. Hiessl, S. R. Lakani, J. Kemnitz, D. Schall, and S. Schulte, "Cohort-based federated learning services for industrial collaboration on the edge," *Journal of Parallel and Distributed Computing*, vol. 167, pp. 64–76, 2022.

[121] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019.

[122] A. Abusitta, G. H. de Carvalho, O. A. Wahab, T. Halabi, B. C. Fung, and S. Al Mamoori, "Deep learning-enabled anomaly detection for iot systems," *Internet of Things*, vol. 21, p. 100656, 2023.

[123] A. Chatterjee and B. S. Ahmed, "Iot anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, p. 100568, 2022.

[124] T. Andrysiak, "Machine learning techniques applied to data analysis and anomaly detection in ecg signals," *Applied Artificial Intelligence*, vol. 30, no. 6, pp. 610–634, 2016.

[125] M. Gu, Y. Zhang, Y. Wen, G. Ai, H. Zhang, P. Wang, and G. Wang, "A lightweight convolutional neural network hardware implementation for wearable heart rate anomaly detection," *Computers in Biology and Medicine*, p. 106623, 2023.

[126] M. Nawaz and J. Ahmed, "Cloud-based healthcare framework for real-time anomaly detection and classification of 1-d ecg signals," *Plos one*, vol. 17, no. 12, p. e0279305, 2022.

[127] Z. Ji, J. Gong, and J. Feng, "A novel deep learning approach for anomaly detection of time series data," *Scientific Programming*, vol. 2021, 2021.

[128] C. Cambra Baseca, S. Sendra, J. Lloret, and J. Tomas, "A smart decision system for digital farming," *Agronomy*, vol. 9, no. 5, p. 216, 2019.

[129] N. Kaur and S. K. Sood, "Cognitive decision making in smart industry," *Computers in industry*, vol. 74, pp. 151–161, 2015.

[130] S. A. A. Bokhari and S. Myeong, "Use of artificial intelligence in smart cities for smart decision-making: A social innovation perspective," *Sustainability*, vol. 14, no. 2, p. 620, 2022.

[131] M. W. Moreira, J. J. Rodrigues, V. Korotaev, J. Al-Muhtadi, and N. Kumar, "A comprehensive review on smart decision support systems for health care," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3536–3545, 2019.

[132] S. Zhou, R. Zhang, D. Chen, and X. Zhu, "A novel framework for bringing smart big data to proactive decision making in healthcare," *Health Informatics Journal*, vol. 27, no. 2, p. 14604582211024698, 2021.

[133] M. T. Quasim, A. Shaikh, M. Shuaib, A. Sulaiman, S. Alam, and Y. Asiri, "Smart healthcare management evaluation using fuzzy decision making method," 2021.

[134] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[135] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[136] J. Passerat-Palmbach, T. Farnan, R. Miller, M. S. Gross, H. L. Flannery, and B. Gleim, "A blockchain-orchestrated federated learning architecture for healthcare consortia," *arXiv preprint arXiv:1910.12603*, 2019.

[137] R. Kumar, A. A. Khan, J. Kumar, A. Zakria, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, and W. Wang, "Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging," *IEEE Sensors Journal*, pp. 1–1, 2021.

[138] B. Yuan, S. Ge, and W. Xing, "A federated learning framework for healthcare iot devices," *arXiv preprint arXiv:2005.05083*, 2020.

[139] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, 2021.

[140] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[141] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, p. e2024789118, 2021.

[142] C. Briggs, Z. Fan, and P. Andras, "A review of privacy-preserving federated learning for the internet-of-things," *Federated Learning Systems: Towards Next-Generation AI*, pp. 21–50, 2021.

[143] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2021.

[144] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023.

[145] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35–44, 2020.

[146] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2021.

[147] S. Xuan, M. Jin, X. Li, Z. Yao, W. Yang, and D. Man, "Dam-se: a blockchain-based optimized solution for the counterattacks in the internet of federated learning systems," *Security and Communication Networks*, vol. 2021, pp. 1–14, 2021.

[148] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system," *arXiv preprint arXiv:1906.10893*, pp. 2327–4662, 2019.

[149] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[150] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[151] S. Aich, N. K. Sinai, S. Kumar, M. Ali, Y. R. Choi, M.-I. Joo, and H.-C. Kim, "Protecting personal healthcare record using blockchain federated learning technologies," in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, pp. 109–112, 2021.

[152] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[153] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[154] R. Myrzashova, S. H. Alsamhi, A. V. Shvetsov, A. Hawbani, and X. Wei, "Blockchain meets federated learning in healthcare: A systematic review with challenges and opportunities," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[155] J. Passerat-Palmbach, T. Farnan, M. McCoy, J. D. Harris, S. T. Manion, H. L. Flannery, and B. Gleim, "Blockchain-orchestrated machine learning for privacy preserving federated learning in electronic health data," in *2020 IEEE International Conference on Blockchain (Blockchain)*, pp. 550–555, 2020.

[156] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 151–159, IEEE, 2019.

[157] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[158] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[159] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[160] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, 2020.

[161] V. Drungilas, E. Vaičiukynas, M. Jurgelaitis, R. Butkienė, and L. Čeponienė, "Towards blockchain-based federated machine learning: Smart contract for model inference," *Applied Sciences*, vol. 11, no. 3, p. 1010, 2021.

[162] M. A. Rahman, M. S. Hossain, M. S. Islam, N. A. Alrajeh, and G. Muhammad, “Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach,” *Ieee Access*, vol. 8, pp. 205071–205087, 2020.

[163] M. Ejaz, T. Kumar, I. Kovacevic, M. Ylianttila, and E. Harjula, “Health-blockedge: Blockchain-edge framework for reliable low-latency digital healthcare applications,” *Sensors*, vol. 21, no. 7, p. 2502, 2021.

[164] A. Benayache, A. Bilami, S. Barkat, P. Lorenz, and H. Taleb, “Msm: A microservice middleware for smart wsn-based iot application,” *Journal of Network and Computer Applications*, vol. 144, pp. 138–154, 2019.

[165] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, “A survey on security issues in services communication of microservices-enabled fog applications,” *Concurrency and Computation: Practice and Experience*, vol. 31, no. 22, p. e4436, 2019.

[166] X. Zhao and C. Huang, “Microservice based computational offloading framework and cost efficient task scheduling algorithm in heterogeneous fog cloud network,” *IEEE Access*, vol. 8, pp. 56680–56694, 2020.

[167] M. Abdullah, W. Iqbal, A. Mahmood, F. Bukhari, and A. Erradi, “Predictive autoscaling of microservices hosted in fog microdata center,” *IEEE Systems Journal*, vol. 15, no. 1, pp. 1275–1286, 2020.

[168] S. Pallewatta, V. Kostakos, and R. Buyya, “Microservices-based iot applications scheduling in edge and fog computing: A taxonomy and future directions,” 2022.

[169] L. N. T. Thanh, N. N. Phien, H. K. Vo, H. H. Luong, T. D. Anh, K. N. H. Tuan, H. X. Son, *et al.*, “Ioht-mba: an internet of healthcare things (ioht) platform based on microservice and brokerless architecture,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.

[170] I. Lera, C. Guerrero, and C. Juiz, “Yafs: A simulator for iot scenarios in fog computing,” *IEEE Access*, vol. 7, pp. 91745–91758, 2019.

[171] T. Qayyum, A. W. Malik, M. A. K. Khattak, O. Khalid, and S. U. Khan, “Fognetsim++: A toolkit for modeling and simulation of distributed fog environment,” *IEEE Access*, vol. 6, pp. 63570–63583, 2018.

[172] J. Das, A. Mukherjee, S. K. Ghosh, and R. Buyya, “Spatio-fog: A green and timeliness-oriented fog computing model for geospatial query resolution,” *Simulation Modelling Practice and Theory*, vol. 100, p. 102043, 2020.

[173] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, “Multi-objective optimization for resource allocation in vehicular cloud computing networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25536–25545, 2021.

[174] A. Mohammadzadeh, M. Masdari, and F. S. Gharehchopogh, "Energy and cost-aware workflow scheduling in cloud computing data centers using a multi-objective optimization algorithm," *Journal of Network and Systems Management*, vol. 29, pp. 1–34, 2021.

[175] M. Hosseinzadeh, M. Y. Ghafour, H. K. Hama, B. Vo, and A. Khoshnevis, "Multi-objective task and workflow scheduling approaches in cloud computing: a comprehensive review," *Journal of Grid Computing*, vol. 18, pp. 327–356, 2020.

[176] A. Mohammadzadeh and M. Masdari, "Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–21, 2021.

[177] K. Velusamy, D. Venkitaramanan, S. K. Vasudevan, P. Periasamy, and B. Arumugam, "Internet of things in cloud," *Journal of Engineering and Applied Sciences*, vol. 8, no. 9, pp. 304–13, 2013.

[178] P. Lai, Q. He, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *International Conference on Service-Oriented Computing*, pp. 230–245, Springer, 2018.

[179] "kaggle.com/code/devavratatripathy/ecg-anomaly-detection-using-autoencoders." https://www.kaggle.com.

[180] M. Supriya, K. Sangeeta, and G. K. Patra, "Trustworthy cloud service provider selection using multi criteria decision making methods," *Engineering Letters*, vol. 24, no. 1, pp. 1–10, 2016. Cited By :32.

[181] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, "Ehopes: Data-centered fog platform for smart living," in *2015 International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 308–313, IEEE, 2015.

[182] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE Access*, vol. 5, pp. 9882–9910, 2017.

[183] R. Paesani, G. Paolone, P. Di Felice, D. Iachetti, and M. Marinelli, "ifogsim simulations on iot computational alternatives," *Engineering Proceedings*, vol. 31, no. 1, p. 44, 2022.

[184] D. G. Nair, C. Aswartha Narayana, K. Jaideep Reddy, and J. J. Nair, "Exploring svm for federated machine learning applications," in *Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2022*, pp. 295–305, Springer, 2022.

[185] V. P. Pillai and R. K. Megalingam, "System partitioning with virtualization for federated and distributed machine learning on critical iot edge systems," in *Congress on Intelligent Systems: Proceedings of CIS 2021, Volume 2*, pp. 443–453, Springer, 2022.

[186] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing: A systematic survey," *Sensors*, vol. 22, no. 2, 2022.

[187] D. Jose and J. Swaminathan, "A model in healthcare cloud for securing the data using fog computing," in *ICT Infrastructure and Computing: Proceedings of ICT4SD 2022*, pp. 441–448, Springer, 2022.

[188] P. D. Bharathi, V. A. Narayanan, and P. B. Sivakumar, "Fog computing enabled air quality monitoring and prediction leveraging deep learning in iot," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–22, 2022.

[189] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[190] L.-p. Jin and J. Dong, "Intelligent health vessel abc-de: an electrocardiogram cloud computing service," *IEEE transactions on cloud computing*, vol. 8, no. 3, pp. 861–874, 2018.

[191] W.-Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, and A. A. Abd El-Latif, "Secure and optimized load balancing for multitier iot and edge-cloud computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8119–8132, 2020.

[192] O. Cheikhrouhou, R. Mahmud, R. Zouari, M. Ibrahim, A. Zaguia, and T. N. Gia, "One-dimensional cnn approach for ecg arrhythmia analysis in fog-cloud environments," *IEEE Access*, vol. 9, pp. 103513–103523, 2021.

[193] W. Y. B. Lim, S. Garg, Z. Xiong, D. Niyato, C. Leung, C. Miao, and M. Guizani, "Dynamic contract design for federated learning in smart healthcare applications," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16853–16862, 2020.

[194] A. Lakhan, M. A. Mohammed, J. Nedoma, R. Martinek, P. Tiwari, A. Vidyarthi, A. Alkhayyat, and W. Wang, "Federated-learning based privacy preservation and fraud-enabled blockchain iomt system for healthcare," *IEEE journal of biomedical and health informatics*, vol. 27, no. 2, pp. 664–672, 2022.

[195] W.-M. Lee and W.-M. Lee, "Testing smart contracts using ganache," *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pp. 147–167, 2019.

List of Publications

Papers Published

1. S. M. R, M. Supriya, "Performance Comparison of VM Allocation and Selection Policies in an Integrated Fog-Cloud Environment," in Ubiquitous Communications and Network Computing: 4th EAI International Conference, UBICNET 2021, Virtual Event, March 2021, Proceedings 2021, pp. 169-184, Springer International Publishing.

2. S. M. Rajagopal, M. Supriya and R. Buyya, "Resource Provisioning Using Meta-Heuristic Methods for IoT Microservices With Mobility Management," in IEEE Access, vol. 11, pp. 60915-60938, 2023, doi: 10.1109/ACCESS.2023.3281348. Impact factor - 3.557, Percentile - 90th percentile (Q1)

3. S. M. Rajagopal, M. Supriya and R. Buyya, "FedSDM: Federated learning based smart decision making module for ECG data in IoT integrated Edge–Fog–Cloud computing environments," Internet of Things, vol. 22, pp. 100784, 2023, doi: 10.1016/j.iot.2023.100784. Impact factor - 5.711, Percentile - 97th percentile (Q1)

4. S. M. Rajagopal, M. Supriya and R. Buyya, "Blockchain Integrated Federated Learning in Edge-Fog-Cloud Systems for IoT based Healthcare Applications: A Survey". Federated Learning: Principles, Paradigms, and Applications published by Apple Academic Press, Exclusive co-publishing with CRC Press, Taylor & Francis Publisher, USA (2024), (Accepted and yet to be published).

Papers Communicated

1. S. M. Rajagopal, M. Supriya and R. Buyya, "Leveraging Blockchain and Federated Learning in Edge-Fog-Cloud Computing Environments for Intelligent Decision-Making with ECG Data in IoT" in Journal of Network and Computer Applications, Elsevier (Under review).