PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
University Echahid Hamma Lakhdar Of El Oued
Faculty of exact sciences
DEPARTMENT OF COMPUTER SCIENCES

A Thesis

Presented for the award of the grade of

DOCTORATE

In : **computer science**

Speciality : **artificial intelligence**

By :

Kenioua Laid

Theme :

IoT approach for smart cities based on edge computing

Thesis defended on Feb 26, 2025 at El Oued in Front of the Jury Composed of:

Mr. **NAOUI** Mohamed Anouar	Associate Professor	University of El Oued	President
Mr. **LEJDEL** Brahim	Full Professor	University of El Oued	Supervisor
Mr. **NEDIOUI** M.Abdelhamid	Associate Professor	University of El Oued	Co-Supervisor
Mr. **Meftah** M. Charaf Eddine	Full Professor	University of El Oued	Examiner
Mr. **GASMI** Mohamed	Associate Professor	University of Tebessa	Examiner
Mr. **MERZOUG** Soltan	Associate Professor	University of Tebessa	Examiner

Academic Year 2024/2025

I would like to dedicate this work to ... my parents: May they find here the testimony of my deep gratitude and acknowledgment To my brothers and my sisters, my grandparents and my family who give love and liveliness. To all those who have helped me - directly or indirectly - and those who shared with me the emotional moments during the accomplishment of this work and who warmly supported and encouraged throughout my ýourney. To all my friends who have always encouraged me, and to whom I wish more success. Thanks!
Laid Kenioua

Acknowledgements

First of all, I thank God, who helped me and condemn it for my presence to provide me with this opportunity and gave me the ability to go forward successfully. I would like to express my sincere gratitude to Professor Brahim Lejdel to obtain the continuous support for doctoral study and relevant research, his patience, motives and tremendous knowledge. His guidance helped me all the time research and write this thesis. I couldn‘t imagine a better consultant and teacher to study. We would like to thank the members of the jury, who have done us the honor of accepting to be reviewers and for kindly devoting some of their time to reviewing our work. We would like to thank all the people with whom we had the joy of collaborating, directly or indirectly, on our project. They were all able to share their knowledge and skills with us. Finally, I express my gratitude to all those who ha ve contributed in one way or another to the development of this work. O Allah, send your blessings on your noble messenger, his family, and companions,and bless us in our life.

Abstract

Edge computing has emerged as a transformative technology that processes data close to its source, rather than relying on remote centralized data centers or the cloud. This approach significantly reduces latency, enhances processing speed, and enables real-time decision-making, making it ideal for meeting the dynamic needs of smart cities.

In urban environments, edge computing plays a critical role by supporting various applications such as smart traffic management, energy consumption monitoring, and public safety through smart cameras and sensors. Edge computing contributes to creating more efficient, responsive, and sustainable urban systems in real time, by enabling faster data analysis and reducing the burden on central networks, thus contributing to the development of smart cities. In this thesis, we use edge computing techniques and methods to support the development and operation of smart cities in various applications including supporting the work of self-driving cars, healthcare applications, and supporting data security in edge computing units. This integration between edge computing and smart cities allows data to be collected and analyzed close to the processing unit, allowing applications to make decisions in real time and improve their performance.

Keywords: Edge Computing, Smart Cities, IOT , Big Data.

ملخص

لقد برزت الحوسبة الحافة كتكنولوجيا تحويلية تعمل على معالجة البيانات بالقرب من مصدرها، .بدال ًمن االعتماد على مراكز البيانات المركزية البعيدة أو السحابة يقلل هذا النهج بشكل كبير من زمن الوصول، ويعزز سرعة المعالجة، ويمكّن من اتخاذ القرارات في الوقت الفعلي، مما يجعله .مثاليًا لتلبية االحتياجات الديناميكية للمدن الذكية في البيئات الحضرية، تلعب الحوسبة الحافة دورًا حاسمًا من خالل دعم تطبيقات مختلفة مثل إدارة حركة المرور الذكية ومراقبة استهالك .الطاقة والسالمة العامة من خالل الكاميرات وأجهزة االستشعار الذكية تساهم الحوسبة الحافة في إنشاء أنظمة حضرية أكثر كفاءة واستجابة واستدامة في الوقت الفعلي، من خالل تمكين تحليل البيانات بشكل أسرع وتقليل العبء على الشبكات المركزية، وبالتالي المساهمة في تطوير المدن .الذكية في هذه ألطروحة، نستخدم تقنيات وأساليب الحوسبة الحافة لدعم تطوير وتشغيل المدن الذكية في تطبيقات مختلفة بما في ذلك دعم عمل السيارات ذاتية القيادة وتطبيقات الرعاية .الصحية ودعم أمن البيانات في وحدات الحوسبة الحافة يسمح هذا التكامل بين الحوسبة الحافة والمدن الذكية بجمع البيانات وتحليلها بالقرب من وحدة المعالجة، مما يسمح للتطبيقات باتخاذ .القرارات في الوقت الفعلي وتحسين أدائها.

الكلمات المفتاحية: حوسبة الحافة، المدن الذكية، إنترنت ألشياء، البيانات الضخمة.

Résumé

L’Edge Computing est devenu une technologie transformatrice qui traite les données à proximité de leur source, plutôt que de s’appuyer sur des centres de données centraux distants ou sur le cloud. Cette approche réduit considérablement la latence, améliore la vitesse de traitement et permet une prise de décision en temps réel, ce qui la rend idéale pour répondre aux besoins dynamiques des villes intelligentes. Dans les environnements urbains, l’informatique de pointe joue un rôle crucial en prenant en charge diverses applications telles que la gestion intelligente du trafic, la surveillance de la consommation d’énergie et la sécurité publique grâce à des caméras et des capteurs intelligents. L’Edge Computing contribue à créer des systèmes urbains plus efficaces, plus réactifs, en temps réel et plus durables, en permettant une analyse plus rapide des données et en réduisant la charge sur les réseaux centraux, contribuant ainsi au développement de villes intelligentes.Dans cette thèse, nous utilisons des technologies et des méthodes informatiques de pointe pour soutenir le développement et l’exploitation de villes intelligentes dans diverses applications, notamment le fonctionnement de voitures autonomes, les applications de soins de santé, ainsi que la sécurité des données dans les unités informatiques de pointe. Cette intégration entre l’edge computing et les villes intelligentes permet de collecter et d’analyser les données au plus près de l’unité de traitement, permettant ainsi aux applications de prendre des décisions en temps réel et d’améliorer leurs performances.

Mots-clés: Informatique en périphérie,Villes intelligentes,Internet des objets ,mégadonnées .

Contents

List of Figures

List of Tables

List Of Publications

International Jornals

1. Laid Kenioua, Brahim Lejdel, Sultan Alamri, Qusai Ramadan, A password-based authentication approach for edge computing architectures, Egyptian Informatics Journal, Volume 28,2024,100543, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2024.100543.

2. Laid Kenioua, Brahim Lejdel, Mohamed Abdelhamid Nedioui, "A Comprehensive Approach to Combat GPS Spoofing and Ensure Security Positioning in Autonomous Vehicles", The International Arab Journal of Information Technology (IAJIT) , Volume 21, Number 04, pp. 627 - 635, July 2024, doi: 10.34028/iajit/21/4/7.

International Conferences

1. L. Kenioua, B. Lejdel and N. M. Abdelhamid, "Empowering Healthcare: Edge Computing and ML for Early Heart Attack Detection And Prevention," 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), EL OUED, Algeria, 2024, pp. 1-8, doi: 10.1109/PAIS62114.2024.10541283. keywords: Support vector machines;Machine learning algorithms;Computational modeling;Medical services;Predictive models;Prediction algorithms;Real-time systems;Edge computing;healthcare;cardiac arrest;real-time;Cloud computing,

2. Abdelhamid, K., Ammar, T.B., Laid, K. (2022). Artificial Intelligent in Upstream Oil and Gas Industry: A Review of Applications, Challenges and Perspectives. In: Lejdel, B., Clementini, E., Alarabi, L. (eds) Artificial Intelligence and Its Applications. AIAP 2021. Lecture Notes in Networks and Systems, vol 413. Springer, Cham. https://doi.org/10.1007/978-3-030-96311-8.24

3. M. Kara et al., "An Authenticated Method for a Secure Changing Password," 2024 5th International Conference in Electronic Engineering, Information Technology and Education (EEITE), Chania, Greece, 2024, pp. 1-6, doi: 10.1109/EEITE61750.2024.10654422.

keywords: Design methodology;Education;Authentication;Passwords;Proposals;Password;Authent
system;P2P Networks,

4. laid kenioua, brahim lejdel,"Revolutionizing Renewable Energy Grids: The Power of Edge Computing and AI Integration",first international conference for renewable energies their applications(1st ICERA-2023)

International Conferences

1. laid kenioua, brahim lejdal ,"smart cities and pollution: an advanced model in edge computing and artificial intelligence to mesure the impact of pesticides on the atmisphere and humain health", 1st national seminar: biotechnology, health and agro-environment.University Of El Oued 14-15/03/2022.

2. laid kenioua,kamal maaloul, brahim lejdel ,"using edge computing for improving healthcare in smart cities". the 1st national conference on new educational technologies and informatics, university of 08 mail1945, guelma, Algeria.NCNET23, 03-04/10/2023.

General Introduction

Context

With the rapid advancements in technology and communication, numerous modern innovations have emerged to enhance the quality of life and streamline daily operations. Among these cutting-edge technologies, Edge Computing has gained prominence as a revolutionary solution for data processing and service delivery, offering increased efficiency and effectiveness. Edge Computing is based on a simple yet powerful concept: bringing data processing closer to the source, rather than relying on distant, centralized data centers. This approach has played a crucial role in enabling a wide range of modern applications, especially in the development of smart cities, improving their services, and providing faster response times and better overall system performance for Internet of Things (IoT)-based infrastructures.

Edge Computing can be defined as a computing model that processes and stores data near the point of generation or at the 'edge' instead of relying on distant cloud data centers. This model reduces latency, improves processing speed, and minimizes the volume of data that needs to be transferred across the network. While cloud computing has been dominant for a decade, the shift to edge computing is gaining momentum due to its ability to process data in real-time or near real-time, which is essential for many applications.

Edge Computing plays a central role in the development of smart cities, which use technology to enhance the efficiency of everyday operations and services, creating a sustainable and convenient environment for residents. With the proliferation of IoT, encompassing connected devices and smart sensors spread throughout urban areas, vast amounts of data are generated continuously, from smart traffic signals, energy consumption meters, to security cameras and environmental sensors.

Here, the effectiveness of Edge Computing becomes apparent; it allows data to be processed close to the point where it is collected, enabling the rapid analysis of information and real-time decision-making. For example, computers at smart traffic signals can

analyze vehicle flow and make decisions on signal changes to ensure smooth traffic flow and prevent congestion without the need to send data to distant data centers and wait for a response.

The primary advantage of Edge Computing lies in its ability to reduce latency and improve efficiency, enabling users and businesses to process data more quickly and effectively, which, in turn, cuts down on data transfer costs and reliance on network infrastructure. This advantage makes it particularly valuable for applications requiring fast responses, such as smart healthcare systems that monitor patient conditions in real-time, or autonomous vehicles that need to process data from multiple sensors to make quick and safe decisions.

By processing data locally, Edge Computing also helps to ease the burden on cloud data centers and enhances security and privacy, as sensitive data does not need to leave the location where it was generated. This reduces the security risks associated with transferring data over public networks. Additionally, this method helps to reduce costs and lower dependence on network infrastructure, especially in areas where networks are unreliable or costly.

With the continued expansion of Edge Computing, we can expect a significant transformation in how cities and businesses manage data. Edge Computing can be utilized to enhance the management of smart energy grids, providing more efficient energy distribution by monitoring and analyzing energy consumption in homes and businesses. It can also be used to improve public transportation services by gathering data on passenger movements and routes to optimize transit systems more effectively.

Moreover, Edge Computing plays a vital role in the healthcare sector, where connected medical devices can process and analyze patient data on the spot, enabling doctors to diagnose medical conditions faster and make quick treatment decisions.

Edge Computing stands as one of the future technological pillars that will reshape the way data is handled and services are delivered, thus contributing to the development of smart cities and improving the quality of life. Thanks to its ability to offer fast and efficient processing, cost reduction, and enhanced security, Edge Computing presents an ideal model for addressing the technical challenges posed by the digital revolution. As IoT technologies continue to evolve, the demand for edge computing solutions will grow, making it an indispensable part of the infrastructure for future smart cities.

Problematic

Despite its advantages, edge computing in smart cities comes with a host of challenges that must be addressed to ensure successful implementation. Security and privacy issues, high setup costs, scalability concerns, and the complexity of managing a decentralized network are all critical factors that can affect the feasibility of deploying edge solutions. Furthermore, the need for interoperability, network reliability, and energy efficiency adds layers of complexity to an already intricate system. For smart cities to truly benefit from edge computing, these problems must be carefully managed through strategic planning, robust infrastructure development, and the establishment of clear industry standards. Only by overcoming these challenges can edge computing fulfill its potential to revolutionize urban environments and create smarter, more efficient cities. Therefore, these questions need to be answered:

1. What are the business visions of edge computing to support smart cities by focusing on how edge computing can fundamentally change work at different levels?
2. What are the most important areas that edge computing can apply in smart cities?
3. How can we leverage the integrated and scalable edge computing frameworks and benefits for smart cities?

Objectives

The main objective of this thesis is:

- Reviewing edge computing applications in smart cities and exploring the opportunities and challenges involved in using this technology in smart cities. In addition, investigating the general requirements for designing and implementing edge computing applications and services in smart cities.
- Presenting the most important and best frameworks that allow for the analysis of big data using edge computing to manage smart cities, in addition to presenting a study of massive edge computing in smart cities.
- Evaluate the security between edge computing units and IoT devices and provide a proposal to increase security to ensure more reliability and productivity for our strategy.

Contributions

In our thesis, we propose three contributions to solve several problems using edge computing in smart cities.

Contribution 01:

This work presents an engineering model based on hybrid cooperation for secure positioning measurements in autonomous vehicles. The proposed approach enhances inter-vehicle cooperation, offering a highly reliable solution that resists GPS spoofing, signal jamming, and fraudulent attacks. Additionally, it ensures secure communication between different vehicles, further strengthening the overall security framework.

Contribution 02 :

Enhancing security in edge computing environments requires the implementation of strong encryption, secure authentication protocols, and regular software updates. Initially, a robust yet lightweight mutual authentication method was designed specifically for low-cost devices that efficiently utilize the edge computing framework. In the proposed approach, following the registration phase, mutual authentication between edge nodes is achieved in two steps, enabling seamless verification between edge nodes and users.

Contribution 03 :

One of the most important features of cloud-based prediction software is real-time work. The challenge of working in an edge computing environment is that it provides better results in remote areas or in network pressure areas, where with the provision of lightweight machine learning models with the optimal choice of machine learning algorithms, do we obtain more efficient results than their counterparts that work in the cloud?

Thesis Plan

This thesis is divided into Four main chapters.

- Chapter 1: provides an overview of edge computing and smart cities, its characteristics, challenges, and applications, as well as the role of edge computing in smart cities. This chapter begins with a brief introduction to edge computing, followed by a definition of the terms associated with it in the smart city environment. Then, we examine some studies related to edge computing techniques to solve some problems in smart cities.

- Chapter 2: presents an engineering model architecture based on hybrid collaboration for secure positioning measurements in autonomous vehicles. The proposed approach enhances the collaboration between vehicles, providing a highly reliable solution that is resistant to GPS spoofing, signal jamming, and fraudulent attacks. In addition, it ensures secure communication between different vehicles, further enhancing the overall security framework.

- Chapter 3: To enhance security with encryption and secure authentication protocols in edge computing environments, a robust yet lightweight mutual authentication method is introduced, well-suited for low-resource devices operating within edge computing. Following the registration phase, the authentication process can be completed in just two steps, enabling the edge node and the user to seamlessly authenticate each other.

- Chapter 4:In the field of healthcare, a machine learning model was created to predict angina pectoris, working through IoT sensors and through an edge computing environment in remote areas. To ensure real-time operation, a method for mitigating the model and its operation on weak devices was presented. A comparison was also made between three machine learning algorithms and the quality of the results obtained was demonstrated.

Chapitre 1

Theoretical Concepts, Definitions, and Related Research

In recent years, the tremendous scientific development of information technology has led to the generation of large amounts of data and the emergence of what has become called big data. Which in turn led to the development of techniques and methods of storage, treatment, and use. And since traditional data analysis methods may not be able to deal with a huge amount of data, cloud computing has emerged. In recent years, the Internet of Things revolution has become the future of information technology actors such as home control, health care, industry, logistics, agriculture, and in Service field. With progress, smart cities have emerged, in which the latest technologies and modern technologies are combined. Some obstacles appeared in the use of cloud computing, as its work is limited in some areas, and some applications depend on real-time work, bandwidth, and lack of access to offline mode. Where information can be shared quickly, safely, and without delay, especially in the areas of services, health care, games and commercial services. It relies on low latency, improved reliability, faster response time, and real-time work. It also combines the work of the cloud and advanced infrastructure.

1.1 Introduction

The last decade has seen significant growth in Internet activities and advancements in information and communication technologies, leading to an explosion of data known as big data. This has resulted in new methods for collecting, processing, storing, and computing data, along with the rise of cloud computing. Additionally, the development of the Internet of Things (IoT) and electronic industries has introduced innovative solutions, such as smart cities, which use IoT devices to improve quality of life in areas like transportation, health services, and security.

Smart cities integrate technology to enhance daily life, manage natural and man-made disasters, and improve decision-making processes through data gathered from sensors and other devices, with artificial intelligence playing a crucial role. However, the vast amount of data and the need for real-time processing pose challenges, particularly in cloud-based systems, which can experience bandwidth limitations and slower response times.

To address these challenges, new computing paradigms such as fog computing and edge computing have emerged. These approaches move data processing closer to the user or data source, improving response times and reducing network load. Despite progress, further research is needed on edge computing frameworks and big data analysis in smart cities to optimize their functionality and future potential.

1.2 Definitions

In this chapter, we define the most important concepts and terms on which this study is based, to help the reader better understand the context of the research.

1.2.1 smart city

There are many definitions of a smart city, for example, IBM defines a smart city as a city that makes optimal use of all interconnected information available today to better understand and control its operations and optimize the use of limited resources. In general, a smart city is an urban area in which the latest information and communication technologies (ICT) are used to create, disseminate, and promote development within the city, meet urban and technical challenges, create sustainable and technically feasible infrastructure, and promote economic growth while improving the quality of life for citizens through the use of smart technologies and data analysis The intelligence of any city is determined using a set of characteristics, among which are :

- Technology Infrastructure.
- Environmental initiatives that optimize resource utilization, promote modern and efficient transportation, and enhance parking facilities.
- Integrated and effective health care, Modern and advanced means of communication.
- Confident and progressive city plans.

The advanced information and communication technologies used in cities alone are not sufficient to create a smart city. There is also a need for data analysts and advanced smart systems to evaluate the process, follow up on changes, and make decisions To achieve the goal of improving quality of life, the ICT framework integrates real-time data

from networked devices and IoT devices. Citizens can also engage and interact with smart city ecosystems through mobile devices, connected vehicles, smart buildings, and mobile devices It is important to reduce expenditures, increase sustainability, simplify variables such as distribution systems and waste collection, reduce energy consumption as well as reduce traffic congestion and improve air quality and health services, by connecting devices to data and city infrastructure

1.2.2 Edge Computing

Cloud computing has greatly transformed the way we understand and deal with technology and has become an important part of our lives, work and studies since its inception around 2005 And software service providers (SaaS), such as Google Apps, social networking programs, Facebook, Twitter, and others, have become widespread in our daily lives. Moreover, scalable infrastructures as well as processing engines developed to support cloud service greatly influence the way business is run, providing a modern look and feel for business management, Google File System, MapReduce, Apache Hadoop, Apache Spark ..and others The Internet of Things (IoT) technology appeared in 1999 to manage the supply chain and then the concept of making computer information without the aid of human intervention was widely adapted to other fields such as health care, home, environment and transportation Now with the Internet of Things, the goal has shifted beyond the cloud, as there will be great quality of data generated by IoT devices in our daily lives, and a lot of applications will also be deployed on the edge to consume this data. By the new millennium, data generated by people, machines, and things will reach 500 zettabytes, according to Cisco Global Cloud Index estimates, however, IP traffic to the global data center will only reach 10.4 zettabytes by then With increasing sophistication, 45percent of the data generated using IoT will be stored, processed, analyzed and acted upon near the network or at the network edge There will be more than 50 billion things connected to the Internet by 2022, as predicted by Cisco Internet Business Solutions Group Some IoT applications may require very short response time and work in real time, some may contain private data and private encryption, and some may allow us a large amount of data which may be a significant load on networks. This is why cloud computing is not efficient enough to support these applications. And think of a new approach

1.2.2.1 Définition

Edge computing is a distributed information technology infrastructure in which data is processed and stored in the vicinity of the network, as close to the original information source as possible, often in the form of local devices, mobile devices, or handheld

devices that are distributed across different locations With the increasing popularity of the Internet of Things (IoT), edge computing has become increasingly important.

1.2.2.2 How Edge Computing Works

The basic work of edge computing is to capture information from a data source and process it as soon as possible, and the process of capturing data is done by sensors, tools and computing devices to collect data and send it to edge or cloud servers depending on the type of task, the amount of data and the expected results. This data may feed into machine learning systems or models, provide automation, or give a view of the current state of a device, system, site, product, etc Currently, most big data computing is done via the cloud or data centers, and with the shift of enterprises to the edge computing model with the widespread use and spread of Internet of Things (IoT) devices, the need to deploy edge servers on a large scale, gateway devices and other equipment that It reduces computing time and distance, and thus connects the entire enterprise and city infrastructure, which may also include deploying smaller edge data centers in scattered and remote secondary locations, or using cloud containers that can be moved easily across the cloud and systems Not only are local advanced data centers the only way to process data, Internet of Things (IoT) devices can also process data at their level, and data can be sent to a smartphone, edge server, or storage device to handle computing operations. We can form an advanced network through a variety of techniques. These include mobile computing that operates via wireless devices ; Infrastructure cloud computing that uses clouds and other means of storage to place data in different locations ; and so-called cloudlets that act as micro-data centers. The edge framework delivers the flexibility, data agility, and scalability required for a growing range of IoT use cases and real-time scalability. For example, a sensor might provide a real-time alert about the storage temperature of a vaccine and whether it has been kept at the desired temperature throughout the transportation process, track the status of vehicles during an organized transportation process, and provide real-time insights into congestion and routing within cities. Motion sensors can also include artificial intelligence algorithms that detect when an earthquake strikes to provide early warning that allows businesses and homes to shut down gas supplies and other systems that may lead to a fire or explosion, track elderly people, etc the primary reason for the growth of edge computing is quality and efficiency. where all that collected data is processed somewhere. as the volume of data produced from IoT devices increases, more and more processing is done on the edge. over time, today's connected devices are getting smarter, enabling the ability to program "Edge AI" - artificial intelligence at the edge - a growing trend in edge intelligence

1.2.2.3 Edge Devices

In an edge work environment, the systems and devices that capture and transmit data act as sophisticated and intelligent devices. This can include standalone devices as well as gateways that connect to devices and interconnect and interact with other systems within the network. However, the primary role and greatest interest revolves around IoT sensors and devices on the edge. These systems can include a range of different sensors, including sensing light, sound, motion, humidity, magnetic fields, gravity, electric fields, and chemicals. It can process data using applications or via internal computing capabilities, which often include batteries or are connected to a power source [1]. Edge and IoT devices can transact through a variety of communication protocols, including : *Bluetooth Low Energy (BLE) *RFID *Wi-Fi *Zigbee *Z-Wave *Cellular (including 5G) *NFC *Ethernet . Edge IoT devices send data via an Open Systems Interconnection (OSI) framework that suggests Approved devices and standards. These systems also connect to Internet and cloud protocols such as AMQP, MQTT, CoAP, and HTTP. The business mostly relies on a specialized edge device, a smart gateway, to route and transmit data and manage communications within the network. Microprocessor operations used in IoT devices are constantly growing and evolving. Not only are some chips able to accommodate internal processing and integrated artificial intelligence and machine learning functions, but they have become more capable, smart, and energy efficient. Some can wake up on demand and work only during a certain time. 5G chips give faster and more powerful communications capabilities which also enable the ability to change the Internet of Things and the edge by transferring data between devices. This enables IoT devices and systems and edge frameworks to advance and gain new capabilities [2].

1.2.2.4 Edge Connection Gateway

To make the most of IoT devices, efficient and modern methods and mechanisms are required to connect the edge with the cloud and data centers. Therefore, an edge gateway serves this purpose. After IoT devices collect various data and perform processing at the local level, either on the device itself or on a separate remote device such as a smartphone or at the cloud level, the gateway manages the flow of data between the end network and the cloud or the data center. By using traditional encryption capabilities or machine learning capabilities, it can only transmit necessary or important data, thus improving bandwidth and lowering transmission costs. The edge gateway also interacts with IoT edge devices in a consistent direction, being told when to turn on and off or how to adapt to different conditions. When data is needed, a device or multiple device connectivity test can be performed. This allows us to perform analytics and machine learning at the

edge, being able to isolate devices, manage traffic patterns more effectively, and share and connect various gateways to other gateways, thus creating a larger network and a larger and more modular area coverage from the various connected devices. As a result, the IoT framework can operate in a very fluid and dynamic way [3].

1.2.2.5 Edge computing importance

Computing tasks differ from one type to another depending on the type of data, location, and type of work, and a structure that fits one type of computing task does not necessarily fit all types of computing tasks. Edge computing has emerged as a viable and important paradigm that supports distributed computing to spread computing and storage resources close to a data source (virtually in the same physical location). It enshrines the view of decentralization in the performance of tasks and computing. The concept of decentralization can be difficult to implement, as it requires high levels of monitoring and control that can be avoided by moving away from the traditional centralized computing model. Edge computing has become relevant because it provides an effective solution to the emerging network problems associated with the transmission of huge amounts of data produced by enterprises and devices. The huge amount of data is no longer a concern only. It is also a matter of time ; Where applications rely on processing and responses in real time are increasingly time sensitive. Nowadays, self-driving cars are an example of that. which rely on intelligent traffic control signals. Vehicles and traffic controllers will need to produce, analyze, and share data in real time. Consider the huge numbers of self-driving vehicles, and the range of potential problems becomes clearer. This requires a fast and responsive network. Edge and fog computing addresses three major network limitations : bandwidth, response time, and congestion, or reliability [4].

- Bandwidth : The concept of bandwidth expresses the ability to transfer a connection, or more generally, the amount of data that a network can carry over time, and it is an important factor when determining the quality and speed of a network or Internet connection. Bandwidth is measured as the amount of data that can be transferred from one point to another within the network. in a specified period of time. Normally, bandwidth is expressed as bit rate and is measured in bits per second (bits per second). All networks have limited bandwidth, this means that there is a limited limit to the amount of data or the number of devices that can deliver data across the network. The network bandwidth can be increased to accommodate more devices and data, but the cost of the process is usually expensive and expensive [5].
- Latency : Latency is the time required to send data between two points on a network,

often measured as the latency between the user's "client" device and the data center. This measurement helps developers understand how quickly data moves and users load an application or a web page. Although the connection is made at the ultra-fast speed of light, large physical distances associated with network congestion or outages can delay network traffic. This leads to the delay of any analyzes and decision-making processes, which leads to a decrease in the ability of the system to respond in real time. This may lead to human losses in cases of self-driving cars [5].

- Congestion : Despite the development of the Internet and its large and fundamental role in the exchange of data between users for general purposes and its contribution to most daily computing tasks such as file exchange or basic flow, the volume of data contained and flowing in tens of billions of devices constitutes a great burden on the Internet, which It causes many anomalies such as congestion and forcing data retransmissions that take a long time. In other cases, Sometimes network congestion is the result of temporary conditions, or a sign of deeper chronic problems such as pending repairs or misconfigurations that require more significant solutions. A network outage can exacerbate congestion and even disconnect some Internet users completely, rendering IoT devices useless during outages [5].

1.2.3 Big data

Big data is a term used to describe large amounts of data in disparate formats that flow into different organizational systems at high speed [6, 7]. This data requires the use of special tools to analyze it and extract insights from it that can give businesses a competitive advantage ,volume of data is not the only feature of big data [8], There are many advantages of big data, including :

— Volume expresses the vast amounts of information generated every second from social media [6], cell phones, cars, credit cards, M2M sensors, photos, videos, etc.

— Variety. Big data is created with multiple types of data and information [9]. Compared to traditional data such as personal information, phone numbers, etc., the new trend of data is in the form of photos, videos, audio recordings and many more, which makes a large amount of the resulting data completely unorganized and needs to be reorganized, arranged and filtered. [9].

— Veracity means the degree of reliability that the data has to offer. Since a major part of the data is unstructured and irrelevant, Big Data needs to find an alternate way to filter them or to translate them out, as the data is crucial in business developments [9].

— Value is the major issue that we need to concentrate on. It is not just the amount of data that we store or process. It is actually the amount of valuable, reliable and trustworthy data that needs to be stored, processed, and analyzed to find insights [6].
— Velocity plays a major role compared to the others ; there is no point in investing so much to end up waiting for the data. Therefore, the major aspect of Big Data is to provide data on demand and at a faster pace [6].

Big data solutions give us control over large amounts of data, for example storage, backups, analysis, and visualization. Big data systems also allow us to efficiently manage and organize a complex data infrastructure. In addition, big data solutions enable the use of advanced capabilities in smart cities. Exploiting Internet of Things (IoT) technology, smart sensors, smart transportation, security monitoring, building monitoring, etc. The smaller level where big data is used by individuals in different applications, such as smart map programs that guide users, for example, through transportation networks and different places. On another level, big data also gives smart buildings the ability to monitor and regulate their various functions. Smart buildings can manage and control heating or cooling and automatically regulate lighting in specific rooms. It also enables it to monitor the condition of its repair so that it can refer to preventive maintenance before serious or serious problems occur, and to monitor the security and health status of individuals and other functions. [10] On the other hand, at the higher levels that are still under study, big data gives the opportunity to improve the various systems of the urban city such as health, transportation, infrastructure and development. Through which smart city systems integrate these different measures to provide new concepts for the ways human users interact and interact with their environment Influence them and ways to deal with how urban systems are used in the real world [11].

1.2.4 Cloud Computing

Cloud computing has become one of the following buzzwords in the information technology industry, where, with the large volume of data and the complexity of data processing and computing processes, the concept of cloud [12]emerged, where instead of keeping files, data and various programs on the computer or on the local storage device and processing Data locally, cloud technologies have allowed us to transfer files, data, computing programs, and artificial intelligence models to peers linked with some of them available on the Internet with huge storage and processing capabilities [13].Cloud computing is basically the provision of on-demand powerful computer system resources , especially large data and information storage (cloud storage), computing power and data processing, wi-

thout active and continuous monitoring by the user. Mostly large clouds contain several functions distributed on multiple sites and peers, to give different services including resource management and providing tools and applications such as data storage, servers, databases, networks and programs, each site is a data center. Cloud computing depends on sharing resources to achieve the best performance and result and reduce costs for the user and for organizations that use big data. [13] . It is also known as a type of computing in which actions associated with information technologies are provided "as a service", allowing users to access and interact with these services over the Internet ("in the cloud"). They do not have to know or control the technologies behind them . There are types of cloud computing services, including public computing and private computing, where Public Cloud - A public cloud can be accessed by any subscriber with an internet connection and access to the cloud space.. As for A private cloud is established for a specific group or organization and limits access to just that group. There is also a mixed type, which combines the two options, public services and private services, but With different services. [14]

1.2.4.1 Types of Cloud Computing

In cloud computing, there are three main types of services, and there are fundamental differences between them in terms of what can be provided as a service, storage, processing and management of programs and resources, and it is also possible to interact with each other to create a broad and comprehensive model of cloud computing : Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). [14]

— Infrastructure as a Service (IaaS) It is the most popular cloud computing service model because it provides the basic infrastructure for virtual servers, networks, operating systems, and data storage engines. This gives many companies the flexibility, reliability, and scalability they demand from the cloud, eliminating the need for on-premises hardware. In short, managing these services, such as updates and patches, is the provider's responsibility and is ideal for small businesses looking for cost-effective IT solutions to support their business growth. IaaS is a fully outsourced paid service that can be used as a public, private, or hybrid infrastructure. [13, 14]

— Platform as a Service (PaaS) The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers,

operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. [15]

— Software as a Service (SaaS) Software as a Service provides finished products operated and managed by service providers. Most often, when people talk about software as a service, they refer to end-user applications. With SaaS offerings, you dont have to worry about how your services will be maintained or how your underlying infrastructure will be managed. You need to think about how to use this particular software. A common example of a SaaS application is web-based email. This allows you to send and receive e-mail without having to manage the enhancements of your e-mail product or maintain the server or operating system on which your e-mail program is running . [15]

1.2.5 Internet of Things (IoT)

The Internet of Things (IoT) is a system of interconnected computing devices, mechanical and digital machines (or a group of objects) with sensors that have the ability to process with a set of technologies and mechanisms that allow data to be transmitted with various devices over the network without human intervention [14,16]. IoT systems consist of web-enabled smart devices that collect, transmit, and process data collected from the environment using embedded systems such as processors, sensors, and communication hardware. IoT devices share sensor data collected by connecting to IoT gateways or other edge devices. The data is sent to the cloud for analysis or analyzed locally. These devices may communicate with other related devices and respond to information received from each other. Devices perform most of their work without human intervention, but humans can interact with the device to set up the device, provide instructions, access data, and more [17].

1.2.6 Distributed computing

This is the field of computer science that studies distributed systems. A distributed system is a system in which components reside on different network computers, communicate with each other, and coordinate their actions by exchanging messages from any system. Components interact to achieve a common goal. Three key characteristics of a distributed system are component concurrency, lack of global clock, and failure of independent components. [18] This addresses the important issue that a system component failure does not mean a system-wide failure. Examples of distributed systems range from SOA-based systems to large-scale multiplayer online games and peer-to-peer applications.

Distributed computing also refers to using distributed systems to solve computational problems. In distributed computing, the problem is divided into many tasks, each solved by one or more computers that communicate with each other via message passing [19].

1.2.7 Mobile Cloud Computing (MCC)

It combines cloud and mobile computing to provide mobile users, network operators, and cloud computing providers with a wealth of computing resources. [20] The ultimate goal of MCC is to deliver rich mobile applications that can be run on a variety of mobile devices with a rich user experience. MCC offers business opportunities for mobile network operators and cloud providers. Broadly speaking, MCC "utilizes the integrated and resilient resources of diverse cloud and network technologies to deliver unlimited functionality, storage, mobility, Ethernet or Internet channels. It can be defined as "a comprehensive mobile computing technology that serves a variety of mobile devices anytime, anywhere." Follow prepaid principles regardless of heterogeneous environment or platform [21]. The Center for Hard Challenge distinguishes its research focus from close interaction, building, and integration of physical systems and CVS. PS is made up of smart devices and mobile devices, and CVS mainly consists of cloud-based virtual resources and services. Recent developments in augmented reality (AR) 1 have shown some potential applications for MCC [22].

1.3 Application Domain Review :

1.3.1 Smart home :

The concept of automation is the ability to schedule events for devices connected to a local network or the Internet through timed or triggered programs. The concept of automation is applied everywhere to reduce human intervention and improve energy efficiency and productivity, from large industries to small offices, and depends mainly on the methods of connecting the various devices and technologies used in them. [23] IoT products, as well as smart device products such as smart TV, smart lighting, and robots, have become available and varied in the market today. Which in turn led to major changes in life within the city due to the modern technologies use d. In a smart home environment, devices are connected and can be controlled by a smartphone or smart device. Which necessarily leads to the production of a large amount of data that is transmitted over the network. Constraints of real-time response time, high bandwidth issue, and cloud-level processing burden make edge computing ideal for building a smart home. Data is ana-

lyzed and processed locally as it is generated. With Edge Gate running in an advanced operating system (edgeOS), devices can be easily connected and managed. Data is processed locally and there by releasing the burdens of Internet bandwidth and high data privacy and security [24]. ferraris et AI [25] presented the smart home, demonstrating the infrastructure of the various connectivity networks, hardware and software components needed to create a smart home. It also showed concepts about modeling the different habits of the population and providing appropriate services. Park et al. [26] I studied the extent of acceptance of the concept of the smart home and coexistence with the idea, especially with the progress and availability of Internet of things devices, the compatibility of operating systems and different frameworks for their work, and the increased use of wireless and interconnection networks, where the study showed experimentally that enjoyment, compatibility, interconnection, control and cost can stimulate Smart home acceptance. taiwo et al. [27] He pointed out the importance of controlling the smart home to determine user behavior, predict demand, and schedule connected Internet of Things devices. Whereas, smart control is able to learn user behavior and operate automatically and provide smarter and more flexible performance in the smart home environment, On the other hand, the issues of security and privacy are of paramount importance in accepting the idea of a smart home. Yu et al. [28] It showed the correlation and acceptance of the idea of a smart home with the strength of security and privacy granted. qashlan et al. [29] They explained the importance of privacy and data protection for the operation of smart energy systems and their design for the use and production of energy in local environments through interviews that show individuals' attitudes, needs and expectations towards smart energy systems. And to identify the problems that affect smart energy in homes. Sisavath et al. [30] The study showed how configurable systems can be added to enforce security policy to reduce security and safety risks that affect people's behavior. Also in the field of smart home automation Dahlgren et al. [31] To reduce the energy consumption of the smart home automation system. An energy management controller has been proposed to optimize and rationalize consumption. Gozuoglu et al. [32] It showed the need for a culture of consumption and interaction between the user and the smart home in order to exploit its resources and rationalize the use of the smart home automation process. In [33] Babangida et AI to connect communication between different devices and we use Blue tooth 4.0 and it is controlled through various tablets and mobile phones remotely, the shapes or defect here is the inability to control the devices from a long range. 81] piyar et AI The same method was also proposed, that is, the connection based on Blue tooth 4.0 technology, with the possibility of controlling it by phones running on Android, but not being able to control it via the Internet. tooth 4.0, but it can only be contacted from a nearby place. To solve the previous forms, a new model of automation

was proposed in [34] Amoran et AI based on Blue tooth 4.0 and Ethernet, and a mobile application running on the Android system was used to control the various applications and devices. It leads to a deadlock when multiple Android phones try to access the same IoT devices. In [35] stolojescu et AI here an integrated intelligent data acquisition and power management system has been proposed where a web page has been created to display important data as well with the help of SMS, GPRS and Email messages. And to control the various wired and wireless devices, the ARM controller was relied on, and the system is based on the IEEE 1516 protocol, which depends on determining the electronic data sheet for the power adapter for all the adapters used, and here a smart, low-cost and flexible system was presented. [**?**] khunchai et AI to communicate between devices A model based on IoT devices has been adopted to connect and communicate over the Internet through an application installed on a smartphone based on the Android system. The user controls various sensors such as humidity, temperature, smoke, gas, etc., integrated with the mini server raspberry PI. In the event of a notification, the system used is notified via smartphone. In [36] choi et AI to control various smart home devices and also maintain its security, a power management and smart home automation model has been proposed that is built on an Intel Galileo board connected to various sensors and Connected via Internet.

1.3.2 Healthcare :

Mohamed Abdel-Basset et al. [37] created a system that leverages computers, the Internet of Things (IoT), and diverse data sources to diagnose and monitor heart failure patients. Sensors on the patient's body transmit data via Bluetooth to phones, which then send it to a cloud database. By employing IoT and neutrosophic methods, patients are grouped based on shared symptoms. Their research emphasizes cost-effective identification, data analysis, and monitoring of heart conditions. In a study by Li C et al. [38] a robust health monitoring system designed for remote medical applications was introduced. This system involved the regular collection of patients' physical measurements, including blood glucose, blood pressure, blood clotting rate, pulse rate, and blood oxygen levels. These measurements were transmitted in real-time to remote applications. The system utilized data storage and retrieval to analyze the stored medical data, facilitating accurate disease diagnosis and minimizing medical errors. Research on edge caching at Base Stations (BS) to reduce transmission costs through device-to-device (D2D) communications was conducted in studies [39] and [40]. These studies introduced cache replacement strategies utilizing Q-learning to lower transmission costs and enhance network efficiency. Wang et al. [41] explored resource allocation for TDMA/OFDMA-based Mobile Edge Computing

Offloading (MECO) systems. They developed an algorithm that calculates local energy consumption to prioritize and download data to mobile phones, thereby reducing power consumption. To reduce computational complexity in cloud computing, a framework based on Deep Reinforcement Learning (DRL) was proposed in [42]. The authors developed a model using a Markov Decision Process (MDP) to manage limited cloud computation capacity and lower the computation threshold. Soraia Oueida et al. [43] introduced a Petri net-based Resource Preservation Net (RPN) framework for Emergency Department (ED) systems, integrating cloud and edge computing. This framework aimed to model non-consumable resources and optimize key performance metrics such as patient length of stay, resource utilization rate, and average patient waiting time. Ali Hassan Sodhro et al. [44] created the window-based Rate Control Algorithm (w-RCA) to enhance medical Quality of Service (m-QoS) in mobile edge computing-based healthcare, outperforming existing methods in optimizing QoS for remote healthcare systems. Sodhro Fortino [45] proposed energy-efficient transmission power management and media transmission protocols for patient vital sign signals and smart healthcare applications, focusing on energy efficiency, security, and quality of service optimization. Yasuhiro Nagai et al. [46] introduced several techniques for IoT-enabled cardiac monitoring and battery-efficient healthcare applications. However, their study did not focus on Quality of Service (QoS) prediction models for remote healthcare. Overall, these studies emphasize the integration of technologies like IoT, edge computing, and data analysis to enhance healthcare monitoring, diagnosis, and resource optimization. However, additional research is necessary to develop comprehensive Quality of Service (QoS) prediction models in the medical field.

1.3.3 Security And AUthentification

To ensure the validity of nodes in edge computing (EC) systems and facilitate authentication between IoT devices and edge nodes, various techniques have been proposed. These methods employ different cryptographic approaches, such as elliptic curves, hash functions, and Captcha [47–49]. In the study by Guo et al. [50], a distributed authentication method leveraging edge computing and blockchain technology was introduced [51]. The system is structured into three layers : the Physical network, the Blockchain edge, and the Blockchain network. The Blockchain network layer is designed to store authentication data using a consensus mechanism and a signature scheme [52]. Additionally, Malhi et al. [53]. proposed a message authentication technique for inter-vehicle communication that utilizes a bloom filter. This approach assumes that each vehicle is equipped with an on-board unit capable of storage and computation.

Zhang et al. [54], introduced a privacy-preserving authentication model that integrates

5G and edge computing systems. They employed a device-to-device communication technique for vehicle interactions. The protocol is divided into two phases : the first involves authenticating and selecting an edge computing vehicle using a mathematical strategy, while the second ensures mutual authentication between the edge computing vehicle and other vehicles. This setup allows for data exchange within a group of vehicles, although the authentication process still necessitates a relatively large message size.

Jiang et al. [55]. proposed an authentication scheme called anonymous batch authentication, which relies on a certificate revocation list (CRL). When a device receives a message, it first checks the CRL before verifying the certificate and signature. In Another study [56] the authors introduced a message authentication scheme for vehicular ad hoc networks (VANETs) that can easily detect defective vehicles. However, this technique does not support batch verification, making it impractical for real-world VANET systems.

Wang et al. [57] introduced a mutual authentication and key agreement scheme for edge computing that utilizes blockchain technology. This scheme supports key management and ensures conditional anonymity and data integrity [58] ;, though it requires a relatively large amount of data to be exchanged. He et al. proposed an authentication and key agreement protocol in [59], which is susceptible to ephemeral secret key leakage. Kumar et al. [60], presented an anonymous authentication and key agreement technique using symmetric encryption to achieve identity anonymity [61, 62]. However, this method requires the neighborhood area network gateway to maintain numerous keys for different home area network gateways.

Jia et al. [63] proposed an identity-based anonymous authenticated key agreement approach for mobile edge computing, which eliminates the need for a trusted third party during authentication. However, this scheme does not address key management for nodes. Liao et al. [64] introduced a physical layer authentication protocol based on deep learning to enhance the security of mobile edge computing and detect spoofing attacks in wireless networks. This protocol leverages channel state information and employs three gradient descent algorithms to speed up deep neural network training, thereby reducing computation and energy consumption costs.

Yang et al. [65] designed an identity-based authenticated key agreement technique for mobile devices using elliptic curve cryptography. However, Yoon et al. [66] demonstrated that [65]. this technique is susceptible to impersonation attacks and does not offer perfect security. Tsai and Lo [67] presented an identity-based authentication system for distributed mobile cloud computing services, where both users and providers register with a trusted third party that generates long-term secret keys. Although this scheme employs bilinear pairings, which are time-consuming, the computations are handled by service providers with powerful computing capabilities. Nevertheless, Jiang et al. [68] showed that this

technique is vulnerable to service provider impersonation attacks and does not achieve mutual authentication.

Kaur et al. [69] introduced a mutual authentication scheme for mobile edge computing systems that utilizes elliptic curve cryptography and concatenation hash functions. They claim that their technique is resilient against various attacks, including impersonation, replay, and man-in-the-middle attacks. However, the scheme still requires improvements in terms of message size.

Cheng et al. [70] proposed a mutual authentication technique that leverages blockchain and elliptic curve cryptography to provide decentralization, anonymity, and mobility benefits. Sarier et al. [71] introduced a privacy-preserving biometric authentication scheme for multimodal biometrics in mobile edge computing systems. They focused on mitigating hill-climbing attacks by incorporating a non-colluding edge server to detect such attacks.

1.3.4 Autonomous Vehicles

The study by Cui et al. [72] offers an extensive review of the current research on safety failures and security threats in Autonomous Vehicles (AVs), as well as an analysis of the existing countermeasures for these issues. In their paper, Ashish Nanda and colleagues [73] discuss the communication layers of self-driving vehicles, highlighting their features and potential security vulnerabilities. They also provide a brief overview of recent research advancements and future challenges in this area. Xu et al. [74] examine how ultrasonic sensors, commonly used for obstacle detection in AVs, can be compromised through spoofing or jamming attacks. These attacks can cause the vehicles to either stop unexpectedly or fail to stop when necessary, leading to accidents. The authors propose two defense mechanisms : physical shift authentication and multiple sensor consistency checks, to enhance the security of ultrasonic sensors and AVs.

Dutta et al. [75] emphasize the importance of safety and security in automotive systems, especially against electronic attacks or sensor failures that could disrupt vehicle operations. They introduce the concept of "Security for Safety's sake" and propose a system-level security solution that integrates a modified Kalman filter with a chi-square detector to identify and counteract sensor attacks while preserving integrity constraints. The effectiveness of this solution is demonstrated through a case study involving a vehicle tracking scenario where the slave vehicle is equipped with an adaptive cruise control unit.

In their work, Dasgupta et al. [76] propose a framework for analyzing both safety and security issues in autonomous vehicles, integrating the ISO 26262 and SAE J3061 standards with a combined safety and security (S-S) method. This framework is demonstrated on a typical autonomous vehicle model, showing how it can identify and address

vulnerabilities affecting vehicle safety and security. Almeaibed et al. [77] discuss how digital transformation and digital twins can enhance intelligent and autonomous systems in smart manufacturing transportation, while also presenting new challenges and solutions for safety and security functions. They propose a standard framework for vehicular digital twins that supports data collection, processing, and analytics. This approach is illustrated with a case study of a vehicle follower model under a radar sensor attack, and the authors suggest future research directions related to digital twins in the autonomous vehicle industry.

In [78], study, Dasgupta et al. propose a model to estimate the uncertainty of autonomous vehicles and assess cyber risks, aiding decision-makers in enhancing physical design and minimizing attack surfaces. The model employs neutrosophic sets to address multi-criteria decision-making problems with conflicting criteria and alternatives. It integrates MCDM, AHP, MABAC, and PROMETHEE II with SVNSs. A case study involving ten risks in self-driving vehicles demonstrates the model's feasibility. The results suggest that this model is more consistent and reliable than existing methods for handling and representing uncertainty and incomplete risk information using neutrosophic sets.

He et al. [75] propose a UML-based cybersecurity framework for Connected and Autonomous Vehicles (CAVs), adhering to the UK CAV cybersecurity principles. This framework is used to categorize potential vulnerabilities in CAV systems. The study also introduces a new CAV communication cyber-attack dataset (CAV-KDD), derived from the KDD99 dataset. The paper compares and analyzes two classification models, Decision Tree and Naive Bayes, trained on the CAV-KDD dataset to detect various types of communication-based attacks. The findings indicate that the Decision Tree model is faster and more suitable for detecting CAV communication attacks.

The research by Gu et al. [79] focuses on Vehicular Ad-hoc Networks (VANETs), which offer numerous benefits to intelligent transportation systems but also face significant challenges in resource allocation, security, and privacy. The study reviews various resource allocation schemes in VANETs, including bandwidth, power, computing, and storage. It also examines the security and privacy issues related to messages, vehicles, and the network, and discusses methods to protect them from malicious attacks and tracking. The work identifies and explores open issues and future research directions in resource allocation and VANET security and privacy.

In study, Habib et al. [80], introduce VeRA, a security risk analysis method designed for evaluating attack risks in antiviruses and connected and autonomous vehicles (CDVs). VeRA considers human capabilities and the level of vehicle automation in its analysis. It employs a simplified process with fewer factors than the SAE J3061 standard, reducing analysis time without compromising accuracy. VeRA also includes a mathematical model

to assess risk by factoring in attack probability, severity, and human control. The study demonstrates VeRA's effectiveness through a case study on a general autonomous vehicle model, comparing it to other methods. The results indicate that VeRA can identify critical attacks as accurately as other methods while also evaluating the potential for human control over vehicle automation. Additionally, VeRA is reported to save approximately 43persent of analysis time compared to the benchmark.

Chapter 2

Strengthening GPS Security in Autonomous Vehicles .

The development of autonomous vehicles signifies a pivotal moment in transportation, showcasing significant advancements in artificial intelligence, edge computing, sophisticated sensing technologies, and control systems. These vehicles are equipped with advanced sensors and AI, allowing them to operate independently of human drivers. They utilise various technologies, including cameras, radar, and GPS, to gather detailed information about their environment and make immediate decisions. However, autonomous vehicles encounter unique challenges, particularly their reliance on accurate and dependable location data for safe driving and timely decision-making. A primary issue is that central positioning systems can be susceptible to attacks, including security breaches, spoofing, and signal jamming, which may compromise vehicle control. The ability to accurately assess a vehicle's location is crucial for enhancing driving accuracy, optimising decision-making, improving mobility in diverse environments, and facilitating effective communication between vehicles for superior collective performance. This research presents an engineering model that employs mixed collaboration for secure positioning of autonomous vehicles. In this model, a leader vehicle determines its position through satellite systems, while other vehicles in the group rely on the leader's location for their own positioning, ensuring resilience against GPS spoofing, signal jamming, and other fraudulent activities. Additionally, to guarantee secure communication between vehicles, a robust encryption system is integrated to facilitate message transmission within the proposed framework, enhancing reliability. The employed technique is lightweight and effective, utilising only a single multiplication and a single exponential operation. Furthermore, the analysis of network traffic and the complexity of various algorithms has been conducted to validate the proposed framework's efficiency and effectiveness.

2.1 Introduction

Technological advancements have led to the rise of self-driving vehicles, which are considered revolutionary in transportation systems, monitoring, disaster management, and search operations. These vehicles provide solutions and assurances for safer and more efficient transportation through the use of smart systems. With the progress in computing processes, autonomous vehicles now rely on edge computing, with each vehicle connected to an edge unit [87] . Self-driving vehicles are not operated directly from the vehicle itself but are monitored by an administrator from a remote wireless station. The administrator intervenes only in dangerous, critical, or emergency situations to prevent disasters.

In research, monitoring, and disaster management, the importance of teamwork among self-driving vehicles is emphasised. These vehicles form a mobile sensor network, enabling them to communicate and coordinate while performing tasks. Airborne Communication Networks (ACNs) offer rapid emergency communication and precise surveillance services [88], This has garnered significant interest from researchers due to the network's highly dynamic topology, which enhances the vehicles' capabilities in challenging tasks across vast and remote areas.

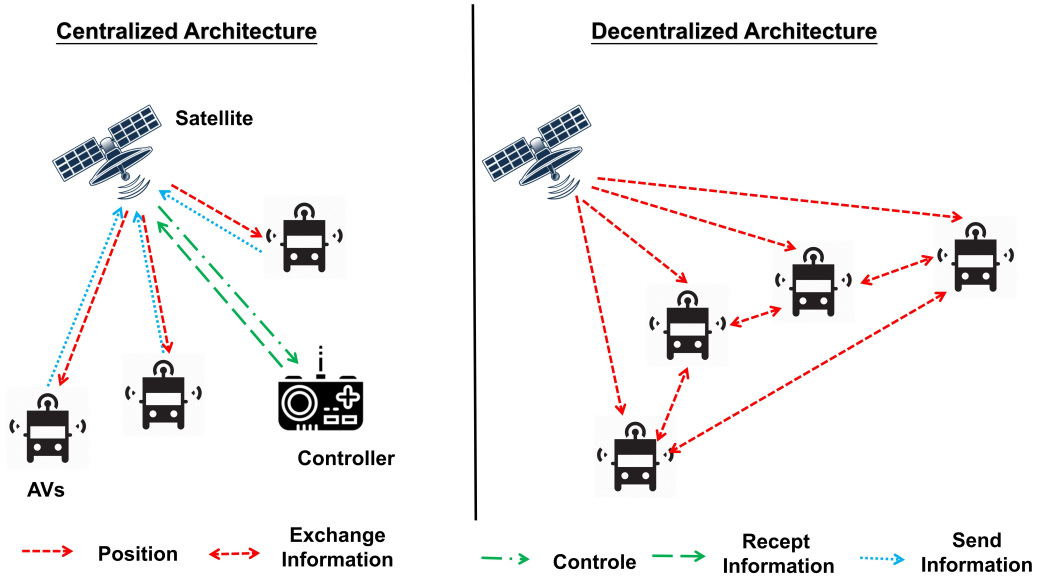

Figure 2.1: Centralized and decentralised architectures

Self-driving cars can be operated by radio waves and multiple controls over short distances and restricted geographic areas, but not in open, remote areas, or even in remote areas hundreds of kilometres from the primary monitoring and control station, where they are not functional. Radio waves travel over great distances, but satellite technology is used to carry out the work [89]. This technology is distinguished by the strength of the signal

and the absence of interruptions. Data gathered from the various technologies mounted on the vehicle, including the compass, speedometer, radar, and satellite positioning devices (GPS), is used to determine the locations of self-driving cars on the ground.

To coordinate and manage teamwork among self-driving vehicles in terms of communication and task planning, two methods are used (Figure 2.1). The first is the central method, where a single entity controls the group. This method's main advantage is its speed and minimal network traffic. However, it has significant drawbacks, primarily the risk of controller failure. The second method is decentralised, relieving on communication between the vehicles. In this approach, learning is fundamental at various levels to achieve clear objectives such as information exchange, task sharing, and collective decision-making. This method allows the system to adapt to its environment

In our proposed model, we integrate both central and decentralised methods. We designate a leader for the group, who determines their location using GPS satellites. The rest of the team members are identified based on the information sent by the leader. All team members are connected via radio waves to a special communication vehicle positioned centrally for large distances.

2.2 Security Technology in AVs: Addressing Security Risks and Spoofing Attacks

The technology behind Autonomous Vehicles (AVs) is rapidly evolving within the Internet of Things (IoT) landscape. AVs are increasingly being used across various sectors, including public, private, recreational, commercial, and military applications. [90] However, as the use of AVs grows, so do the associated security risks. Sensitive data such as photos, videos, GPS locations, and other private information are at risk of being targeted by hackers. The capability to remotely control AVs introduces potential for electronic attacks, such as denial of service and unauthorized takeovers. If these security concerns are not properly addressed, it could lead to the exploitation of widely used commercial vehicles.

This study tackles these challenges by concentrating on two key areas. The first area involves establishing secure communication between vehicles. While this topic won't be discussed in detail here, as it typically involves encryption techniques for peer-to-peer information exchange, it remains a vital component of vehicle security. The second, and highly significant, area focuses on mitigating spoofing attacks (see Figure 2.2). In this context, we will review various studies and contributions from the existing literature.

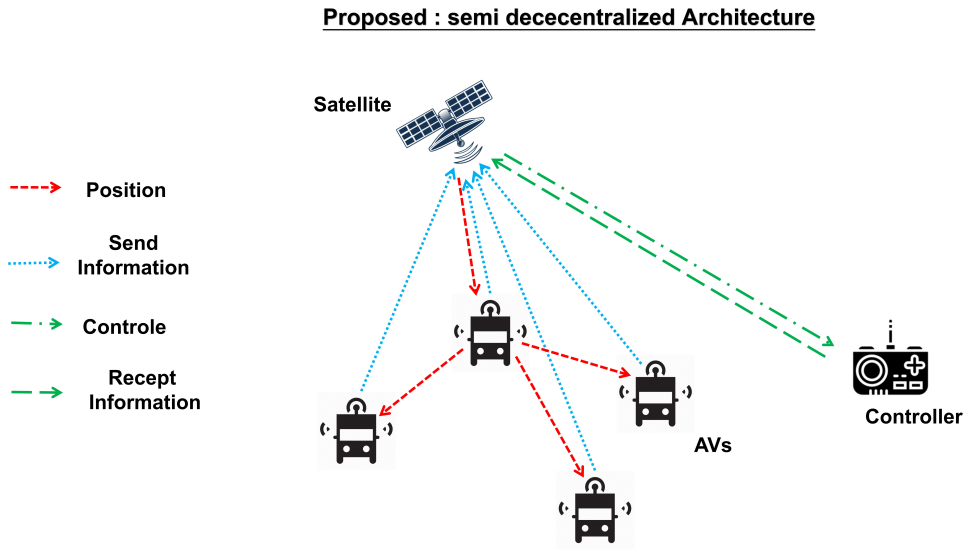

Figure 2.2: Proposed semi-decentralized architecture for measurement of position

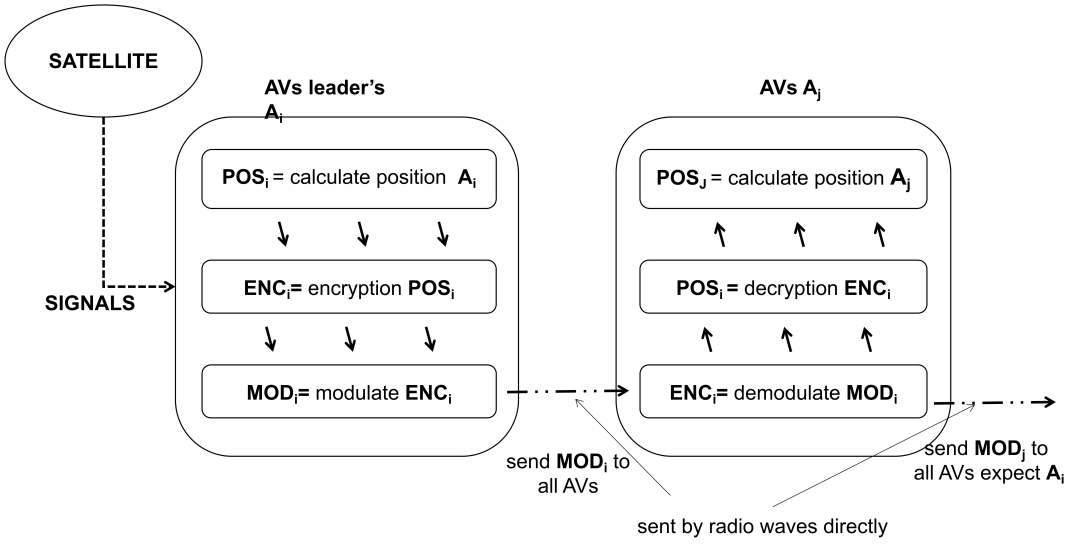

Figure 2.3: The general phases in our approach

2.2.1 Securing data exchange in AVs

There are several positioning systems available, including two Global Navigation Satellite Systems (GNSS): GPS from the United States and GLONASS from Russia. Additionally, the European Galileo system and the Chinese Beidou-2 system are in the process of becoming global. Generally, the designs for all GNSS variants are quite similar. However,

the weakest point is the GPS receiver used in autonomous vehicles (AVs), which rely on unencrypted civilian GPS signals. These civilian GPS signals are not designed for safety and security systems.

Unlike military GPS signals, which keep their codes secret, civilian GPS signals are neither encrypted nor authenticated. These signals are broadcast using well-known spread symbols, making it difficult for AV equipment receivers to distinguish between genuine and spoofed signals. There have been several successful attempts to manipulate the trajectory of autonomous vehicles by spoofing civilian GPS signals. The authors in [84] examined the vulnerabilities of GNSS used by AVs and proposed a prediction-based strategy to detect spoofing attacks. GNSS, which provides Positioning, Navigation, and Timing (PNT) services, is vulnerable to spoofing due to the lack of encryption and the use of open-access codes.

The study employs a recurrent neural network model known as the long short-term memory (LSTM) model. Alheeti, K. M. A et al. [91] emphasized the critical role of detecting 3D objects for the safe and responsible operation of autonomous vehicles (AVs). AVs utilize LiDAR sensors to swiftly and accurately detect objects under various conditions. However, these sensors are vulnerable to spoofing attacks through laser satirizing, which can deceive AVs by providing false information. To counter such attacks, the paper suggests a machine learning model (specifically, decision trees) to detect LiDAR spoofing attacks. In [92], the authors introduced a system to attack FMCW mmWave radar used in automotive applications. By employing a single rogue radar, it can simultaneously spoof distance and velocity measurements, creating coherent phantom measurements.

A proof-of-concept hardware-based system was developed, successfully demonstrating two real-world scenarios. The study also discusses countermeasures to mitigate the described attack. Paper [83] introduced a deep reinforcement learning (RL)-based method for detecting turn-by-turn spoofing attacks in autonomous vehicles (AVs) using low-cost in-vehicle sensor data. This method aims to ensure a resilient Positioning, Navigation, and Timing (PNT) system for AV navigation, even when GNSS signals are compromised. Utilizing the Honda Research Institute Driving Dataset, the deep RL model achieved high accuracy (99.99 percent), recall , precision (93.44 percent), and F1 score (96.61 percent). These results highlight the RL model's effectiveness in detecting turn-by-turn spoofing attacks.

2.3 Proposed System

Determining a secure method for measuring position is a complex challenge. Our approach, illustrated in Figure ??, involves using the GPS position of a designated team member, known as the leader. Before transmission, the leader encrypts their position and shares it with all team members via a wireless network. Each receiving vehicle then forwards the leader's position to other connected vehicles, excluding the sender. Upon receiving the information, the recipient vehicle decrypts it to obtain the leader's position. Using the leader's position and additional data such as radio signal propagation time, each vehicle calculates its own current position. A consensus process is used to select a new leader for a specified time interval (e.g., ten seconds). This interval, denoted as Tc, depends on factors like connection quality, communication speed, and task importance. The leader only needs to be connected to at least one other vehicle to transmit their position. If the leader fails to send a position signal, an additional short period is granted, after which a new leader is selected by the team. If a vehicle loses contact with others, it resorts to GPS for determining its location while attempting to re-establish communication with the team periodically.

Algorithm 1 Position algorithm

Require: Tc: consensus time, Pr: reception period
Ensure: position

```
1: function Pos
2:     per1 ← 0
3:     while per1 < Tc do
4:         increment p1
5:     end while
6:     consensus()
7:     per1 ← 0                                          ▷ per1 : period 1
8:     if I am the leader then
9:         each Pr of time
10:        Pn ← calculate position(GPS info)
11:        En ← encrypt(Pn)
12:        Md ← modulate(En)
13:        send(Md) to all connected vehicles
14:    else                                              ▷ I am not the leader
15:        each Pr of time
16:        receive(Md)   ▷ from the leader or from other connected vehicle
17:        send(Md) to all connected vehicles, except the sender
18:        En ← demodulate(Md)
19:        Pn_leader ← decrypt(En)
20:        Pn_my ← calculate position(Pn_leader and AF)   ▷ AF: additional
   information such as signal propagation
21:    end if
22:    return position
23: end function
```

To prevent external parties from participating in the consensus process, we use a two-step technique. The first step is to create a list of encrypted values, in which each vehicle generates a random temporary number and a corresponding temporary secret key. where each vehicle V_i generates a random temporary number v_i and a random temporary secret key sk_i. The v_i value is then encrypted using the corresponding sk_i according to Equation 2.1.

$$c1_i = v_i \times sk_i \mod p \tag{2.1}$$

Equation 2.1 involves a known prime value p, and the condition is set such that $0 < v_i < p$. This condition is put in place to ensure the correct retrieval of each v_i later. Because if v_i is greater than pn, then the initial value of v_i cannot be recalculated. Equation 2.1 shows at the same time lightweight and hard encryption because both values v_i and sk_i are unknown. The purpose of encrypting the generated v_i is to prevent any

malicious AVs from influencing the outcome of the selection algorithm. To enhance the robustness of the process because each vehicle has the ability to modify the secret key sk_i or manipulate the selection value, we introduced another parameter $c2_i$ computed by Equation 2.2. Besides $c1_i$, each AVs must share $c2_i$.

$$c2_i = sk_i^{sk_i} \mod pn \tag{2.2}$$

Once the encrypted list is created i.e., all $c1_i$ and $c2_i$ are broadcast, autonomous vehicles can start the second step, known as the selection process. In this step, each vehicle V_i have to share its secret key sk_i with the other vehicles. Using these shared keys, each vehicle can calculate all the v_i values by decrypting the $c1_i$ values using Equation 2.3.

$$v_i = c1_i \times sk_i^{-1} \mod pn \tag{2.3}$$

where sk_i^{-1} denotes the multiplicative inverse of sk_i which means $sk_i \times sk_i^{-1} \mod pn = 1$.

After getting all v_i values, the vehicle calculates v based on the following equation:

$$v = (\sum_{i=1}^{N} v_i) \mod N + 1 \tag{2.4}$$

where N denotes the vehicles total number. The last operation at this stage is to identify the leader. Upon computing v the sum of all v_i $(modN + 1)$, the operation is very simple; the leader is the vehicle whose v_i is the closest to v.

2.4 Securing AVs Communication

In Figure 2.4, it is clear that vehicle 3 is within the attacker's control zone, which indicates the area impacted by the jamming signal used to spoof GPS. However, the attacker is unaware that vehicle 3 does not depend on GPS for its positioning; instead, it calculates its location based on the leader's position, which is outside the jamming zone. As a result, vehicle 3 remains unaffected by the interference. It's also important to note that the attacker cannot control the entire area where the autonomous vehicle (AV) team operates. To lessen the effects of a wide jamming range, team members should maintain a safe distance from each other, ensuring that the two most distant vehicles are beyond the attacker's range. This strategy ensures that at least one vehicle is outside the jamming zone. When some team members fall under the attacker's control, the rest of the team will

adjust their positions based on the vehicle that is safe. To strengthen signal reliability during radio transmission modulation and guard against attacks, the leader's position communicated from vehicle 3 to vehicle 4 is encrypted. This encryption protects against interference as well as distance reduction and enlargement attacks from both external and internal threats.

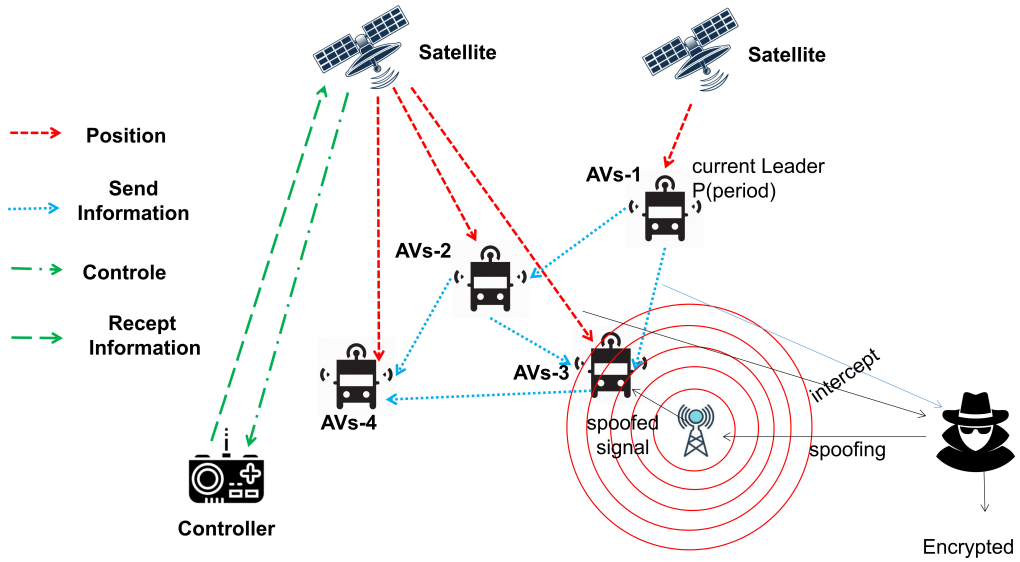

Figure 2.4: Secure position in our proposal

2.5 Secure Data Collection and Processing

To harness modern technologies such as cloud computing and the Internet of Things (IoT), data collection can be facilitated through the use of autonomous vehicles (AVs). Each vehicle can compute specific information values within its designated area, which are then encrypted using an additive scheme [93] before being sent to the cloud for storage. Upon receiving the encrypted data, the cloud server can perform operations on it using homomorphic encryption. In this context, the additive scheme enables the cloud to execute addition operations on data from different vehicles, regardless of their collection times or locations. The additive scheme follows the verification equation:

$$Dec(Enc(v) + Enc(v')) = v + v' \tag{2.5}$$

By utilizing homomorphic addition, the cloud can perform computations on encrypted corporate data without ever accessing the actual values, thereby preserving the privacy of the data owner using the cloud. The owner can later decrypt the data using their secret

key to retrieve the content. The technique outlined in reference [94] aims to obscure the data by randomly dividing it into two components.v_1 and v_2, such that $v = v_1 + v_2$. This fragmentation makes it challenging for an attacker to determine either v_1 or v_2 individually. Consequently, the encryption equation proposed can be represented as follows:

$$c = (v_1 + v_2 \times sk + r \times p) \mod pk \tag{2.6}$$

In this scheme, the symbol sk represents the secret key, r is a randomly generated number, p denotes the prime number, and Pk represents the public key.This approach guarantees the security of information during its transmission to the cloud, thwarting hacking attempts and protecting it from being accessed by the cloud service provider. Furthermore, the encryption scheme allows for additional operations on the encrypted data without needing to decrypt it. This technique is considered a lightweight solution compared to other methods discussed in the literature [93], making it well-suited for the UAV (Unmanned Aerial Vehicle) environment.

2.6 Preferences

In this section, we will discuss two key points: network traffic and algorithm complexity.

2.6.1 Network Traffic

Network traffic refers to the exchange of data packets between devices on a network. It encompasses all digital communications transmitted across the network, including file sharing and other activities. Monitoring and analyzing network traffic is crucial for enhancing network performance by detecting bottlenecks, areas of congestion, and opportunities for improvement. Optimizing network performance allows both businesses and individuals to achieve faster, more reliable data transmission, reduce delays, and improve overall user experience. [95]

Additionally, analyzing network traffic is essential for ensuring security and identifying potential threats. By tracking unusual patterns or suspicious activities, security experts can quickly detect possible cyberattacks, malware, or unauthorized access attempts, enabling swift responses to protect the network. Network traffic analysis is vital to maintaining a stable, secure, and efficient network infrastructure [92]. A deep understanding of data flow aids in better decision-making, improved performance, stronger security measures, and an overall better user experience. To illustrate how data transmission works within a network, consider a group of autonomous vehicles (AVs) equipped with sensors

and communication systems [96]. These vehicles communicate with each other ($V2V$) and a central traffic management system ($V2V$), sharing sensor data, location information, and receiving commands from the central system.

To calculate the amount of data transmitted in our approach, we consider several factors: each vehicle sends sensor data (such as LiDAR and camera data) at intervals of Ts (send-time), site updates are transmitted every P (period), and the size of each site update package is sus (site-update-size).

We put n denotes the number of vehicles, with regard to determining the location, a consensus algorithm is executed to determine the leader, and $n \times (n-1)$ messages will be sent. After that, the leader sends his location to all vehicles, through which the locations of the vehicle paths are determined. Therefore, other $n-1$ messages will be sent from the leader to the rest of the vehicles. Thus, the total traffic can be shown by Equation 2.7.

$$sus = 2 \times (n \times (n-1)) + (n-1) \tag{2.7}$$

To further improve the effectiveness of the proposed system, an additional step can be introduced after the leader transmits its location data. Once each vehicle receives the leader's location, it can broadcast this information to ensure that all elements receive it. This step is necessary because the leader's message may not reach all vehicles due to obstacles or distance limitations. Consequently, $(n-1)$ additional messages will be sent, and the resulting network traffic will be as follows:

$$sus = 2 \times ((n+1)) \times (n-1)) \tag{2.8}$$

2.6.2 Algorithms complexity

Algorithm complexity involves analyzing how an algorithm's efficiency varies with the growth of input size. It measures the time and resources required by the algorithm to solve a problem, providing insights into its scalability and performance [79]. Understanding algorithm complexity is crucial as it helps in selecting the most efficient algorithm for a given task. By comparing and assessing algorithms based on their complexity, we can make informed decisions to optimize computational processes, minimize execution time, and conserve resources.

Understanding algorithm complexity is essential for designing and optimizing software and systems. It allows developers and engineers to balance computational power with practical constraints, ensuring that applications can efficiently manage large-scale data and meet real-time requirements. In short, algorithm complexity offers critical insights

into algorithm efficiency, helping us create faster, more responsive, and resource-efficient solutions suited to the demands of modern computing environments. [97]

In our proposal, we implement two algorithms: one for positioning and the other for leader selection. The first algorithm consists of elementary instructions, giving us a time complexity of $O(1)$ O(1). In the second algorithm, there are two 'while' loops, handling the sending of $(c1, c2)$ sk. As a result, the complexity of the second algorithm is sk $O(n)$.. Therefore, the overall complexity of the proposal can be expressed by the following equation:

$$C = O(n) \tag{2.9}$$

2.7 Conclusion

In our work, we introduce a novel approach to address the issue of GPS spoofing attacks on autonomous vehicles, which poses a significant challenge to the development, operation, and widespread adoption of autonomous driving technologies. Through our analysis, we observed that existing solutions are often either complex to implement, ineffective, or limited by environmental constraints. In contrast, our proposed method offers solutions that prevent the attacker from influencing the vehicles unless they gain control over the entire fleet. To jam the system, all vehicles would need to be within the attacker's jamming range, making such an attack difficult to execute. We evaluated the performance of our proposal by analyzing two key aspects: network traffic and algorithm complexity.

Looking ahead, we plan to carry out field tests to demonstrate the model's effectiveness against GPS spoofing attacks. Additionally, we aim to empirically assess the timing performance of our algorithm, focusing on delay measurements to further evaluate the feasibility and practicality of this approach.

Chapter 3

Efficient Mutual Authentication for Edge Networks

Due to its advantages of low latency and quick response times, edge computing has become a crucial complement to cloud computing, offering new possibilities for smart applications. By shifting some data processing tasks to the network's edge, edge computing enhances cloud computing capabilities. However, the adoption of edge computing brings various security challenges that must be addressed to maintain safety and reliability. A key issue is data privacy, as sensitive information processed at the edge must be safeguarded against unauthorized access and breaches. Strengthening security through encryption, secure authentication protocols, and frequent software updates is essential in edge computing environments. This paper introduces a robust and lightweight mutual authentication method, ideal for low-resource devices utilizing edge computing. Following the registration phase, the authentication process can be completed in two rounds, allowing the edge node and the user to authenticate each other effortlessly. Our method was assessed for communication and computation costs, achieving 982 bits and 5.955 ms, respectively. Both the analysis and experimental results confirm the high performance of the proposed technique.

3.1 Introduction

With the ongoing advancements in modern technologies like 5G and the Internet of Things (IoT), the number of users is increasing rapidly. As a result, the number of devices connected to the Internet is growing exponentially, leading to a surge in data generation. This creates greater demands on the cloud for data processing and storage [98]. On one hand, transmitting such large volumes of data puts significant strain on bandwidth. On the other hand, centralized computing struggles to provide real-time responses for handling vast amounts of data. Additionally, long-distance data transmission and centralized storage raise privacy concerns. In other words, traditional cloud computing has fallen short of meeting users' optimal requirements [99], including issues related to security, latency (particularly for users in remote locations or during peak usage), downtime, reliability, cost management, vendor lock-in, complexity, and system management.

Edge computing (EC) has emerged to overcome the limitations of cloud computing, particularly in the context of the Internet of Things (IoT) and Wireless Sensor Networks [100, 101]. EC utilizes a distributed architecture located near data sources, enabling computing and smart services to be executed on the edge network. There are numerous use cases for edge computing mechanisms. For example, in autonomous vehicles, EC allows for vehicle-to-vehicle communication with ultra-low latency, removing the need for drivers in all but the lead vehicle [102]. In the smart grid, EC enables enterprises to manage and analyze their energy consumption in real time [103]. Similarly, in smart homes, EC facilitates the processing of sensitive data closer to the source, supporting applications like traffic management, cloud gaming, predictive maintenance, and more [104–107]. In application development, a critical concern is how to authenticate components within the cooperative computing environment to ensure security and privacy [108, 109].

While it is true that diverse IoT-based smart applications can benefit from edge computing to enhance both utility and confidentiality, it is evident that not all existing security techniques can be directly applied to address the security challenges of smart applications in edge computing environments. The unique characteristics of edge computing can introduce new security threats to smart environments [110]. For instance, when numerous edge nodes, such as edge servers or controllers, are distributed across a network, the system becomes more vulnerable to attacks. However, with effective security measures in place, these challenges can be mitigated, allowing for the full advantage of edge computing's real-time capabilities. One of the most frequently discussed threats in the literature is authentication attacks [93, 111]. For example, malicious edge nodes may impersonate legitimate ones to steal user data. In such cases, an attacker could disguise a malicious edge node as a legitimate one, tricking users into accessing it and thereby gaining sensitive

information such as account credentials and passwords.

We have propose an efficient password-based authentication scheme applicable to all edge computing (EC) systems. For instance, in a smart home environment, authentication between the user and smart home devices is essential. The proposed architecture relies on transmitting a concealed password, enabling both parties to verify each other and establish secure mutual authentication 3.2).

A common example of authentication is logging into an email account using a username and password. When a user attempts to access their email, they visit the login page and input their unique username along with a secret password they previously set. The system then verifies these credentials by comparing the username and the hashed version of the provided password with the stored records. If both match, the user is granted access to the account. This approach ensures that only individuals who possess the correct username and password combination can access the account, offering a basic layer of security.

A. Contributions

In order to reach a more secure authentication model while achieving lower communication and computation costs in low-power devices, a new technique is presented with the following contributions:

1. A password-based mutual authentication approach that resists the famous attacks is introduced.

2. A security analysis of the proposal is presented.

3. The approach performance and experimental results discussion are discussed.

B. Organization

The rest of this paper is organized as follows: Section **??** presents some related work. In Section 3.2, we show relevant preliminaries. Section 3.3 describes the proposed approach. A scheme analysis is shown in Section 3.4. In Section 3.5, experimentation and performance are discussed. Section 2.7 is the conclusion.

3.2 Preliminaries

Some relevant preliminaries will be discussed in this Section. **A. Edge computing**

Edge computing (EC) is a form of computing performed either onsite or close to the data source, reducing the reliance on remote data centers. With the global expansion of 5G networks, EC represents the next phase in the evolution of cloud computing. Although both IoT and EC are still in their early stages, their full potential remains largely untapped. In this work, we aim to explore and unlock that potential.

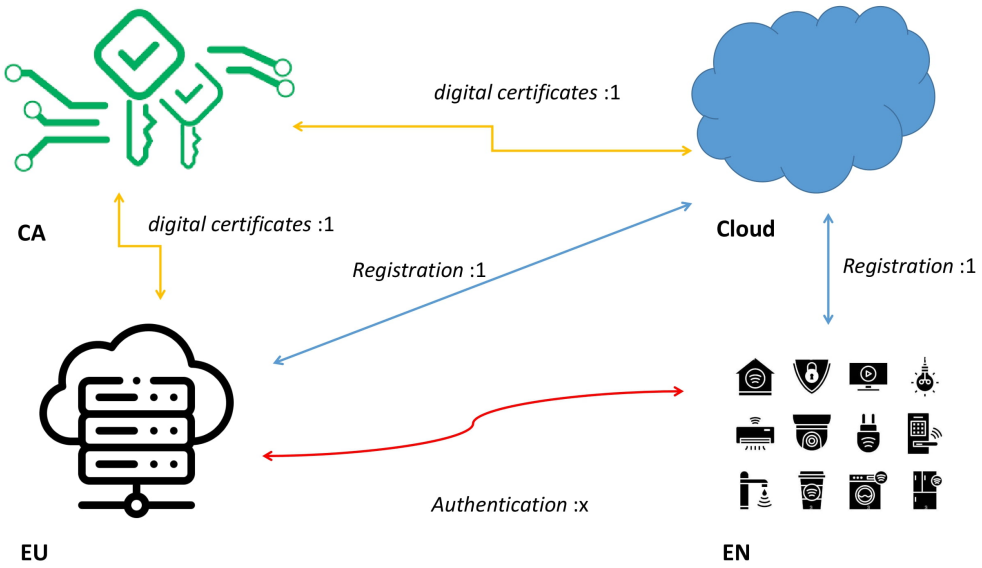

Figure 3.1: System model.

1. CA: the entity that provides, signs, and stores digital certificates which certify the license (ownership) of a public key. A CA works as a trusted third party, this allows cloud and edge nodes to rely upon signatures constructed about the private key that corresponds to the certified public key.

2. Cloud: It is the last destination for data, as it can accommodate a very large amount of data [112]. The connection between the device and the cloud is only once in order for the device to obtain an ID, after that most of the data exchange is between the cloud and the edge.

3. EN: an EN works as the aggregator of data or utility controller, EN has the necessary computation and storage resources. Also, it is responsible for providing timely data analysis, decisions, and service delivery. EDs can communicate with each other and with the cloud to perform further data analysis and make decisions.

4. EU: is usually a terminal device, for example, smart devices in smart homes send related information to an edge node. Each EU can access multiple edge nodes but generally selects the nearest one.

C. Assumptions and security requirements

In this section, we consider the following assumptions and objectives for security:

1. It is considered that the respective clocks of all the elements (devices and edge nodes) are well synchronized [113]. This hypothesis helps to generate fresh timestamps in

every process that supports nodes to resist replay attacks.

2. Under the threat model, an adversary (A) has full access to the channel, which permits A to intercept, alter, replay, and delete the exchanged data.

3. Adversary A can be either an insider or an outsider.

4. Data integrity and authentication: After receiving access data, the receiver can easily determine the legitimacy of the data sender, and decide whether the access data is forged, or modified during the communication process.

5. Forward Secrecy: Every authentication process has a short-term secret number (x) that is valid for this process only, which means that a revealed former secret number can not damage the security of the next secret key. Therefore, an attacker can not guess the later secret number successfully in a polynomial time using the expired one.

6. Resistance to Man-in-the-middle Attack: By using random numbers and hash functions between legal entities, the malicious entity can not extract information and modify the transmitted messages by the legal entities successfully in a polynomial time.

7. Collision resistance: The principal idea behind strong collision resistance is to give a hash function $H()$ and two arbitrary inputs X and Y, and see if there is a chance where $H(X)$ is equal to $H(Y)$. H collision resistance means that an attacker cannot modify a hashed value without detection. Collisions for MD5 function with $n = 128$ can be reached in 21 days in 1994, and 5 hours in 2004. Therefore, to ensure strong collision resistance for the next years, we have to use 180 bits or more.

8. Zero-knowledge proof (ZKP): is a cryptographic strategy that permits one party to prove to another that a statement is true without disclosing any information further the validity of the statement. ZKP helps introduce more robust privacy in any application that uses it.

9. Resistance against Spoofing attacks: it is impossible to use false information to impersonate another entity.

10. Resistance against dictionary attack: it is impossible to get the original password from the exchanged message.

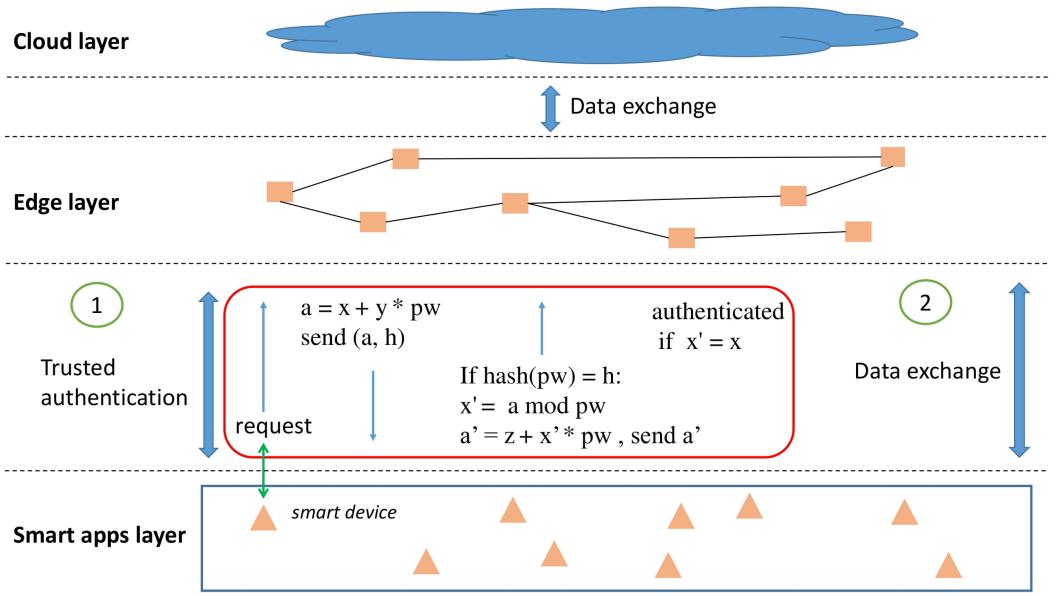

Figure 3.2: Proposed password-based authentication approach in edge computing paradigm.

3.3 Proposed approach

After the data collection process, the IoT device communicates with the edge node to send or exchange the collected data. This interaction requires an authentication process. This paper introduces an efficient and lightweight password authentication protocol to facilitate secure access.

3.3.1 Approach design

The proposed approach operates through the following phases: Setup, Registration, and Authentication.

A. Phase I: Setup Phase

This phase (Figure 3.3) comprises the core of the upcoming stages and initializes all necessary parameters to complete a valid authentication process and ensure safety. The corresponding details are outlined below.

Step 1: The Cloud begins this process by selecting a security parameter that we express here as the password length L.

Step 2: Next, the Cloud chooses a collision-resistant one-way hash functions $H()$.

Step 3: Lastly, $< L, H() >$ parameters are made public.

B. Phase II: Registration Phase

In this phase (Figure 3.4), the device registers itself with the Cloud and in return is given an identity ID_d. The detailed procedure is shown below.

Figure 3.3: Phase I: Setup Phase.

Step 1: Given an identity ID_d, the device selects a password PW_d and sends them to the Cloud over a secure channel.
Step 2: After receiving the device's registration request, the cloud makes sure that the identifier is unique and that the password has a correct length.
Step 3: This information $< ID_d, PW_d >$ is sent to all edge nodes through a secure channel, with a confirmation message sent to the device.
Step 4: both the device and edge node for the upcoming mutual authentication store the received data $< ID_d, PW_d >$.

Storing devices passwords as plain text to be used in authentication process may be not so safe, we can add here the Password-Salting strategy to make it more secure.

C. Phase III: Authentication Phase

In this phase (Figure 3.5), the device and the edge node authenticate each other via a challenge-response procedure intending to establish a shared session key for secure communication.
Step 1: This process is initiated by the device trying to access the services of the selected edge node. It records its time-stamp T_d. Then, the device sends the parameters $< ID_d, T_d >$ to the edge node.
Step 2: Upon receiving these parameters, the edge node validates the received time-stamp T_d. If the time stamp lies within the permissible time window, then the edge node passes the following step after extracting the device password. Otherwise, the connection is discontinued.

Step 3: In this step, the edge node generates two random numbers x and y where x is less than PW_d in order to be extracted by the device. Then, the edge node computes the value of α using Equation 3.1.

$$\alpha = x + y \times PW_d \tag{3.1}$$

After recording its time-stamp T_{en}, the edge node sends the parameters $< \alpha, H_{en}, T_{en} >$ to the device, where H_{en} is calculated using Equation 3.2.

$$H_{en} = H_d = Hash(ID_d \parallel PW_d) \tag{3.2}$$

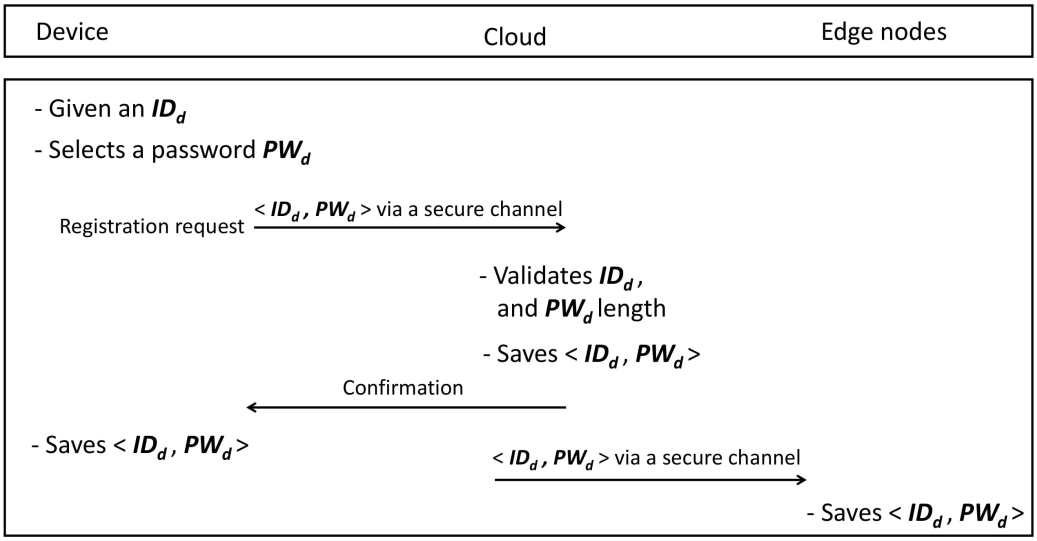

Figure 3.4: Phase II: Registration Phase.

Step 4:When the device receives these values, it checks if T_{en} is valid. If valid, the device verifies the received H_{en} by comparing it with its own calculated value, H_d. The device only needs to compute H_d once and can store it as the calculated H_d. At this point, if the received H_{en} matches the calculated H_d, then the edge node is successfully authenticated.

Step 5: After validating T_{en} and H_{en}, the device extracts the value of x expressed by x' shown in Equation 3.3.

$$x' = \alpha \mod PW_d \tag{3.3}$$

Then, the device generates a random number z with $z < PW_d$.

Step 6: After recording its time-stamp T_d, the device compute α' where $\alpha' = z + x' \times PW_d$, and send the parameters $< \alpha', T_d >$ to the edge node.

Step 7: Here, the edge node validates the received time-stamp T_d and extracts x' by using

Equations 3.4 and 3.5.

$$z = \alpha' \mod PW_d \tag{3.4}$$

$$x' = (\alpha' - z) \div PW_d \tag{3.5}$$

After extracting x' by the edge node, it authenticates the device if $x = x'$.

3.4 Scheme analysis

Mutual Authentication (MA) is a cryptographic method used to verify the identities of interacting parties before further exchanges, ensuring that sensitive data is not transmitted over insecure channels. This paper presents a password-based authentication approach that simulates the zero-knowledge proof (ZNP) technique. The primary reason for using ZNP is to minimize the risk of password leakage during the exchange. Our ZNP simulation enables a remote device to transmit the password to a chosen edge node in a concealed manner. The zero-knowledge proof allows the device to prove to the remote edge node that it possesses the correct password for each login attempt. Similarly, the remote edge node must demonstrate to the requesting device that it knows the valid credentials for the registered password.

In this approach, two random numbers are generated simultaneously, one of which is multiplied by the password, and the second is added to the result to create a new random number. If we define $r = \alpha \times x + \beta$, where x is the known password and α and β are random values, then r is treated as a new random number. This method is commonly used in many random number generation functions. Therefore, transmitting the password in this way serves as a challenge to the device, making it independent of the actual password. The device responds to this challenge in a similar manner. From this process, two messages are generated: the first is from the Prover (edge node) to the Verifier (device), and the second is from the Prover (device) to the Verifier (edge node). These messages can be seen as new random numbers, independent of the original password, which is why the technique is referred to as ZNP (Zero-Knowledge Proof) simulation.

The authentication protocol requires establishing the password between a terminal and an edge node only once. While the current modified Diffie-Hellman key exchange scheme may be sufficient for now, without factoring in future quantum computing capabilities, it faces vulnerabilities. A malicious device equipped with a quantum computer could compromise the initial key exchange by exploiting the weaknesses of the scheme. Quantum computers can easily solve the discrete logarithm problem, which is fundamental to both finite fields and elliptic curves. Consequently, traditional algorithms like Diffie-Hellman

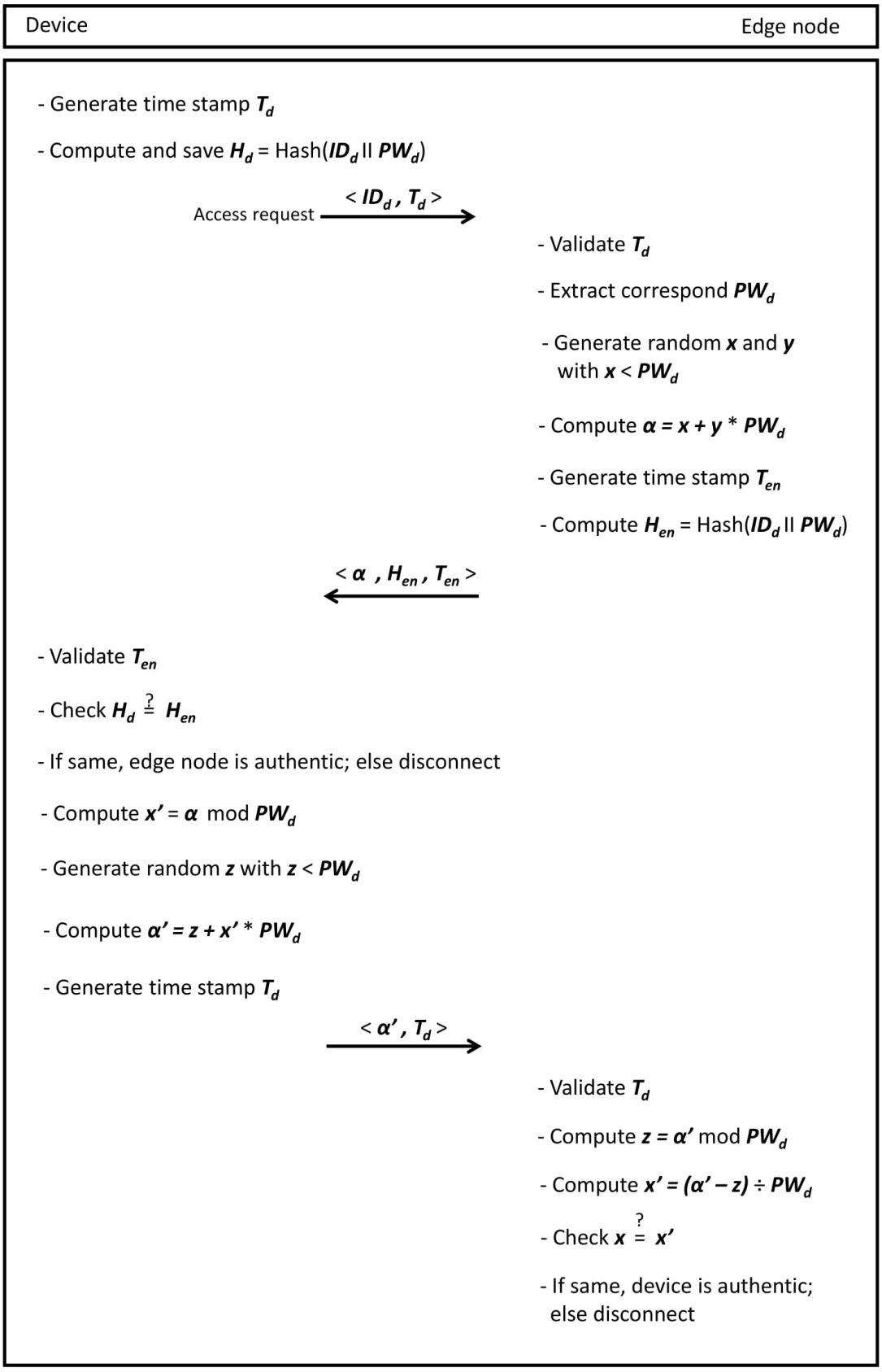

Figure 3.5: Phase III: Authentication Phase.

or RSA are no longer adequate in a post-quantum context. To address this issue, we must implement post-quantum key exchange methods such as Classic McEliece, CRYSTALS-KYBER, NTRU, and SABER.[1] These algorithms are designed to withstand attacks from quantum computers.

In traditional systems, a dictionary attack involves intercepting the password hash exchanged between communicating parties and attempting all possible words from a dictionary to crack it. However, in the method presented here, this type of attack becomes more challenging. The stolen hash is not a simple hash like $H = hash(ID \parallel pw)$. As a result, an adversary would need to perform two successful dictionary attacks: first on the ID and then on the hash H. This added complexity makes the attack significantly more difficult.

Man-in-the-middle (MITM) attacks occur when an adversary intercepts or eavesdrops on a transmitted message and potentially alters it. If terminals and edge nodes accept messages without verifying their source, they may not realize that an attacker has inserted themselves between the communication parties, compromising security. However, mutual authentication can effectively prevent this type of attack, as both devices and edge nodes verify each other's legitimacy by using the same password.

To execute an MITM attack, the attacker would need to intercept a message, such as α or α', which contains the password PW. For the attack to succeed, the attacker must either compute (or guess) the random number x, which is hidden in α, or compute (or guess) the random number z, which is hidden in α' (as shown in Figure 3.5). Due to the complexity introduced by the addition operation in calculating α or α', the attacker would face numerous possibilities in verifying $x_{attacker} + (y \times pw)_{attacker} = x_{edge} + (y \times pw)_{edge}$. This makes it highly complicated to perform a successful MITM attack. Ultimately, if either the device or the edge node cannot verify the sender's identity, the session will be terminated.

A replay attack, similar to a MITM attack, involves retransmitting previously captured messages to deceive the edge node. However, this attack is ineffective against methods employing mutual authentication. To counter such attacks, we have introduced timestamps as an additional verification factor. If the time difference between the current and received messages exceeds the maximum allowed delay, the session will be automatically terminated. Additionally, messages can include unique random numbers for each transmission, further preventing replay attacks and ensuring secure communication.

Given that an adversary cannot access passwords or the generated random values x, y, and z, our method is secure against both Chosen Ciphertext Attack (CCA) and

[1] $https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions$

Chosen Plaintext Attack (CPA). This is implicitly demonstrated by the fact that while an adversary may query encryption and decryption oracles, they have no access to the generated random values. As a result, the proposal is semantically secure, meaning that an adversary who possesses a ciphertext α gains no more information than one who does not have the ciphertext at all. Despite this, the potential set of passwords is not necessarily small, and an adversary might even have chosen the password. Therefore, it is crucial to emphasize the importance of using a secure random number generator, such as a hardware-based generator, to further strengthen security.

	[114]	[115]	[116]	[117]	Ours
Mutual authentication	x	✓	✓	✓	✓
Stolen verifier attack	✓	x	✓	✓	✓
Man-in-the-middle attack	x	✓	✓	✓	✓
Impersonation attack	x	✓	✓	✓	✓
Anonymity	x	✓	✓	✓	x
Forward secrecy	-	-	✓	✓	✓
Chosen Ciphertext Attack	x	x	x	x	✓
Chosen Plaintext Attack	x	x	x	x	✓

Tableau 3.1: Comparison of security features

In spoofing attacks, a malicious device uses false information to impersonate a legitimate device in an attempt to gain access to a remote edge node. However, mutual authentication, as proposed in this approach, is effective against such attacks. The edge node not only authenticates the device but also verifies if it is a legitimate requester before proceeding with any exchange or granting access. This verification relies on a zero-knowledge challenge, where the device must prove possession of a valid password. Even if a malicious device manages to obtain the legitimate device's ID, it cannot spoof the identity without also possessing the correct password.

In impersonation attacks, both the device and the edge node authenticate each other by exchanging random values that only the legitimate counterpart can manipulate, thus verifying their identities as authorized sources. Since malicious devices do not have access to the valid password required for this exchange, they are unable to perform impersonation attacks and cannot act as legitimate devices.

This technique guarantees data privacy by establishing a random session key between the device and the edge node, ensuring that all exchanged messages are protected. As a result, a malicious device cannot extract any private information from the radio channel. Furthermore, it provides perfect forward secrecy, meaning that even if a session key is compromised, the randomness of the values x, y, and z generated for each session ensures that the malicious device cannot derive new session keys from the stolen one.

Table 3.1 provides a summary of the key security features in comparison to other methods.

Balancing security and performance is essential when designing and implementing systems. High security typically introduces additional computational overhead, which can slow down performance and negatively impact the user experience. On the other hand, prioritizing performance might require reducing certain security features, leaving the system vulnerable to potential attacks. For instance, using a shorter random value for x or selecting a weaker hash function like SHA-1 (160-bit) or MD5 to reduce processing time or data size compromises the scheme's overall security. Achieving the right balance between robust security and optimal performance is critical to ensuring both protection and usability.

Tableau 3.2: The used notations, runtimes, and sizes

Notation	Description	Value (ms)
T_m	multiplication	1.97
T_a	Addition	0.012
T_h	hash function	0.009
T_{ed}	encryption-decryption	5.6
T_{be}	big-exponent modular exponentiation	9.352
T_{se}	small-exponent modular exponentiation	0.156
T_s	encryption	0.0005
Notation	Description	Value (bit)
ID	identifier	256
K	token	160
G	Bit length of public parameter	1024
T	Bit length of a timestamp	32
H_1	Bit length of hash	160
H_2	Bit length of hash	256
M_{u1}	message user	320
M_{u2}	message user	2688
M_{s1}	message IoT node	384
M_{s2}	message IoT node	1344
M_{g1}	message gateway	512
M_{g2}	message gateway	1812

3.5 Experimentation and performance discussion

This section presents the efficiency evaluation of our proposed approach in comparison with those of Jing et al. [63], Kuljeet et al. [75], Dipanwita et al. [117], and Chenyu et al. [116]. To standardize the measurements and ensure a more accurate and logical

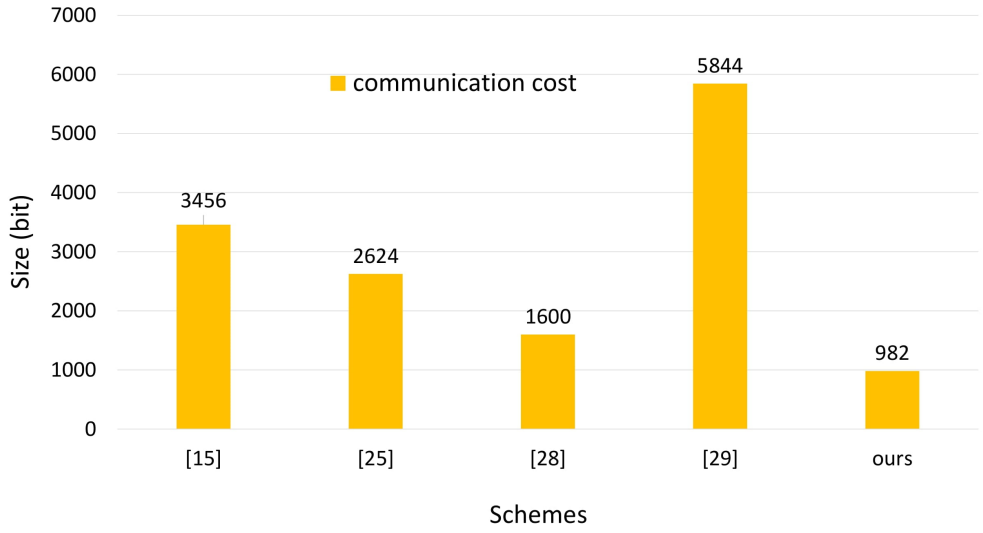

Figure 3.6: Communication costs.

Tableau 3.3: A comparative summary of computation and communication costs

Ref.	Year	Computation cost	Total (ms)	Communication cost	Total (bit)
[63]	2019	$4T_m + T_a + 5T_h$	7.901	$3G + 2H_1 + 2T$	3456
[75]	2019	$4T_m + 4T_h$	7.916	$2G + 2H_2 + 2T$	2624
[117]	2021	$2T_m + 8T_{ed} + 5T_h$	48.785	$3M_{u1} + 2M_{s1} + 1M_{g1}$	1600
[116]	2022	$1T_{be} + 12T_h + 3T_s + 1T_{se}$	9.6175	$1M_{u2} + 2M_{s2} + 1M_{g2}$	5844
Ours	-	$3T_m + 3T_a + 1T_h$	5.955	$1ID + 2k + 2H_2 + 3T$	982

comparison, we utilized the work of Jia et al. [69]. Jia et al. evaluated the performance of several cryptographic processes used in these related schemes on the Alibaba Cloud under the Ubuntu system, measuring the computational power of both the edge node and the terminal. We directly apply these parameters for our comparison. Table 3.2 displays the results from their experiments. For cryptographic operations not covered in their results, we referred to the original values provided in the corresponding reference.

Since they have minimal impact on the overall system performance, the runtime for lightweight processes such as modular arithmetic, concatenation, and random number generation is excluded from the evaluation. Additionally, the performance evaluation of the registration phase is also omitted. A comparative summary of the computation and communication costs for the four techniques [63], [75], [117], [116], along with our proposed method, is presented in **Table 3.3, Figure 3.6, and Figure **??**. These comparisons provide an overview of the relative efficiency of each technique in terms of both computational and communication overhead.

Our implementation settings are as follows: we used an ID size of 256 bits, and the password size is 80 bits. The passwords consist of ten characters, with each character represented by 8 bits. For the hash function, we selected SHA-2, specifically SHA-256, which uses 32-bit words.

- - **Step 1**: The device sends a request message containing an ID and a timestamp of 32 bits, resulting in $M_1 = 256 + 32 = 288$ bits.
- – **Step 2**: The edge node computes a token α of size 80 bits and sends a challenge message that includes the token, a hash value, and a timestamp. This gives $M_2 = 80 + 256 + 32 = 448$ bits.
- – **Step 3**: The device sends a response message containing the token α' and a timestamp, resulting in $M_3 = 160 + 32 = 192$ bits.

In total, the combined size of all transmitted messages is 928 bits.
For the computation cost:

- - **Edge node computation**: The edge node computes α, which involves one addition and one multiplication. It also performs a hash operation to calculate H. The execution time for this step is:

  $0.012\,\text{ms}$ (addition)$+1.97\,\text{ms}$ (multiplication)$+0.009\,\text{ms}$ (hash operation) $= 1.991\,\text{ms}$.

- - **Device computation**: The device computes α', which involves one addition

and one multiplication. The execution time for this step is:

$$0.012\,\text{ms (addition)} + 1.97\,\text{ms (multiplication)} = 1.982\,\text{ms}.$$

- - **Final step**: In the last step, the device performs two operations: subtraction and division. These are equivalent to addition and multiplication, respectively, so the execution time is:

$$0.012\,\text{ms (subtraction/addition)} + 1.97\,\text{ms (division/multiplication)} = 1.982\,\text{ms}.$$

Thus, the total execution time for the entire process is:

$$1.991\,\text{ms} + 1.982\,\text{ms} + 1.982\,\text{ms} = 5.955\,\text{ms}.$$

In Table 3.3, we observe that both [63] and [75] are quite similar in performance when it comes to computation costs, as both employ four multiplication operations. Additionally, the time taken by the hash operation (0.009 ms) and addition process (0.012 ms) is minimal and does not significantly affect overall performance.

The total execution time of [116] is approximately 2 ms longer than both [63] and [75]. This increase is primarily due to the big-exponent modular exponentiation process used in [116], which alone takes 9.352 ms.

On the other hand, [117] has the highest execution time, with a substantial difference compared to the others. This is attributed to the use of encryption-decryption operations, which take 5.6 ms each and are executed eight times, significantly contributing to the overall delay.

In contrast, our scheme achieves the smallest execution time, outperforming [63] and [75] by approximately 2 ms. The total number of operations in each protocol is as follows: 11 for [63], 8 for [75], 15 for [117], 17 for [116], and 7 in our proposed method. This reduction in the number of operations contributes to the improved efficiency of our approach.

Table 3.3 also highlights the total data size transmitted during the authentication phase for each protocol. The largest data size is found in [116], which includes a message of 2688 bits. In contrast, the smallest data size is seen in [117], with a total of 1600 bits, due to its relatively small message sizes (320 and 384 bits).

In [63], although a smaller 160-bit hash function is used compared to other protocols like [75] and ours (which use 256-bit hashes), the total data size is still 3456 bits. This larger size is primarily due to the use of a 1024-bit public parameter in its token.

Our protocol demonstrates the best performance in terms of communication cost, with a total of just 982 bits. This significant difference is achieved by using a small 160-bit token and employing a single hash operation, greatly reducing the overall data size transmitted during authentication.

The proposed authentication system is straightforward, with a clear layout and simple operations, making it suitable for use even by non-specialists. It ensures compatibility with existing edge computing platforms such as AWS IoT Greengrass, Microsoft Azure IoT Edge, and EdgeX Foundry by following industry-standard protocols like OAuth 2.0, OpenID Connect, and TLS. Ease of integration is a key consideration, and the system is designed to include plug-and-play functionality, detailed documentation, and compatibility with legacy systems to minimize configuration efforts.

However, potential deployment challenges may arise, such as ensuring scalability to manage a large volume of authentication requests, optimizing performance for resource-constrained edge devices, addressing security vulnerabilities, handling network reliability issues, and providing sufficient training and support for IT staff and end-users to ensure smooth implementation and operation.

Scalability refers to the system's capacity to expand efficiently. The proposed technique, requiring only two rounds to complete the authentication process, results in a low communication cost. Additionally, the mathematical operations involved are simple, leading to a low computation cost. These factors make the technique well-suited to handle scalability and the resource constraints typically found in edge computing environments, ensuring that it can maintain performance as the system grows or more devices are added.

Although this work has successfully met the research goals outlined in Section 1, particularly in terms of shorter execution time and lower communication costs, it lacks a significant feature that some other techniques offer: anonymity. In our approach, the device transmits its ID in plaintext during the access request step, which could expose the identity of the device. This absence of anonymity presents a potential privacy vulnerability that other methods, which obscure or anonymize the device's identity, effectively address.

3.6 Conclusion

Achieving low latency and providing real-time responses are two critical challenges in traditional cloud-based smart systems, which has driven the need to adopt edge computing architectures. In this work, we present a novel password-based authentication architecture for edge computing, which aims to enable efficient data sharing across different IoT systems. We detail the design of our scheme, and in the security analysis section, we

demonstrate its ability to withstand multiple types of attacks over insecure channels. The proposed model features a lightweight implementation, allowing each party to authenticate the other with just one message, meaning that the entire protocol requires only two messages to complete mutual authentication. Finally, we conclude that the proposed protocol achieves low communication and computation costs (982 vs. 1600 bits and 5.955 vs. 7.901 ms, respectively) based on performance evaluations, making it highly suitable for low-power devices. In the future, we aim to further enhance the proposal by incorporating anonymity features.

Chapter 4

Edge Computing: A Catalyst for Advancing Healthcare

Heart disease remains a leading cause of death worldwide, making early detection crucial for saving lives. Leveraging advancements in healthcare technology, we have developed a system designed to detect heart conditions, including heart attacks, particularly in remote areas. Our solution utilizes edge computing to address network challenges and ensure real-time operation. This is accomplished by integrating Internet of Things (IoT) devices with a local server, enabling local data storage and archiving while maintaining a connection to the cloud for processing larger data sets. The study examines three machine learning algorithms, with comparative results showing the random forest algorithm as the most effective. Our research focuses on evaluating the proposed models' performance and their processing times.

4.1 Introduction

Our healthcare system is vital for treating illnesses, saving lives, and improving community well-being. By utilizing secure healthcare information technology, we can harness advancements that safeguard the confidentiality of health data, ensure its availability when needed, and promote safer, higher-quality, and more efficient healthcare at a lower cost. The rise of the Internet has driven significant progress in healthcare, with ongoing developments continuing to make a global impact. Experts are consistently working to create new techniques, mechanisms, and innovations across various fields, including healthcare, to enhance outcomes and services.

Significant advancements in healthcare have been achieved in the early detection of diseases, the development of precise systems, and the timely monitoring of patient conditions in both clinical and non-clinical environments. Many people now rely on electronic healthcare applications, accessible via mobile devices or within healthcare facilities, to help improve their health and track physical and physiological changes. The use of IoT technologies facilitates the collection of a wide range of health data, which can be shared with specialists and stored on various platforms, including edge devices and cloud systems. This real-time accessibility ensures that doctors can retrieve patient information from any location, enhancing the efficiency of care.

Heart disease, especially cardiac arrest, remains one of the deadliest health conditions worldwide. This paper focuses on utilizing edge computing technology to create a predictive model for cardiac arrest. By effectively harnessing technology and maximizing available resources, our approach aims to improve both the efficiency and accuracy of early detection, potentially saving more lives.

"Health information technology" (Health IT) refers to electronic systems utilized by healthcare providers and, increasingly, by patients to store, share, and analyze health data. It encompasses a broad array of tools and technologies that facilitate the management and use of health information, enhancing the overall efficiency and quality of healthcare services. [118]:

- • Electronic health records (EHRs).
- Personal health records (PHRs)
- Electronic prescribing (E-prescribing)
- Privacy and security,

A smart hospital framework consists of three key layers: data, insight, and access. While hospitals currently collect data, it is not always integrated across all systems, lim-

iting the ability to generate meaningful and "smart" insights. For a smart hospital to be effective, these insights need to be easily accessible to stakeholders such as doctors, nurses, and facilities staff via interfaces like desktop computers, smartphones, or handheld devices. This seamless accessibility enables faster, more informed decision-making, ultimately enhancing operational efficiency. [119] To effectively utilize edge devices in smart healthcare, it is essential to process collected data, share timely insights, and take appropriate actions when needed. Edge computing facilitates this by enabling data processing directly on the edge devices, thus eliminating the need to transfer data to external server environments. This approach reduces latency, improves response times, and ensures more efficient and immediate decision-making in healthcare settings. [8]

4.2 Material and Methodes

The Materials and Methods section provides a detailed overview of the tools, techniques, procedures, and methodologies used in the study.

A complete model of the proposed health care system is shown in Figure 1. It consists of components:

1. Sensing Network:A sensor network comprises sensors strategically placed in various locations to monitor and collect data, which is then transmitted to a central hub for storage, viewing, and analysis. In healthcare, this network is crucial for monitoring patient health by gathering vital signs and transmitting them to the edge computing system for processing .

2. Edge Computing:

   Edge computing leverages advancements in the Internet, computing devices, and secure communication technologies to enable efficient data processing at the edge of the network. This approach allows for real-time processing, offering results comparable to traditional cloud-based methods. Both edge devices and the cloud contribute to different aspects of data processing and management .

3. Data Cleaning:

   The data cleaning process involves removing irrelevant or inaccurate data from datasets and includes data transformation, converting data into usable formats. These steps are vital for preparing data for accurate analysis, warehousing, and other tasks .

4. Data Storage:

Efficient data storage is essential for continuously monitoring patient conditions and tracking their progress or signs of relapse. It ensures that data can be preserved for future reference or reprocessing, as needed.

5. Data Analysis:

   The edge server serves as a data analysis platform, utilizing machine learning and data mining techniques to extract meaningful insights from the collected data. This analysis helps in identifying trends and detecting abnormalities in patient health.

6. Emergency Warning:

   The system can issue prompt alerts or warnings based on the processed data from the edge model, enabling real-time first aid and intervention to patients in emergency situations.

7. Graphical User Interface (GUI) - Screen Monitor:

   Doctors and healthcare specialists interact with the system via a graphical user interface (GUI) displayed on a screen monitor. This user-friendly interface provides easy and convenient access to the results of data analysis, as processed by the edge computing system, facilitating swift decision-making and patient care. These components collectively enable an efficient and reliable healthcare monitoring system, providing real-time insights and responses to improve patient outcomes.

4.3 Architecture Discription

To continuously monitor a patient's condition and provide accurate results, a substantial amount of data is gathered through a network of sensors placed on the patient. A machine learning model operating at the edge processes this data, with the outcomes stored in a local database on a nearby server. This setup enables the creation of patient histories, which can be further analyzed or reprocessed using cloud resources. The data flow process is illustrated in Fig 2. The core concept of this healthcare system is to integrate modern technology with a sensor network that gathers a variety of data, which is then used by a machine learning model to predict the risk of cardiac arrest. The system employs edge computing to process data locally, minimizing dependence on internet connectivity. Data storage is managed in two ways: locally on a dedicated server and in the cloud, where further processing and analysis of large datasets take place to support the machine learning model. The data flow starts with sensors, which send information to the edge system where the machine learning model operates. If a potential health risk is detected,

the system can immediately alert hospital staff via a user interface or alarm device. The cloud operates on a similar principle, processing data and triggering alarms if any health risks are identified.

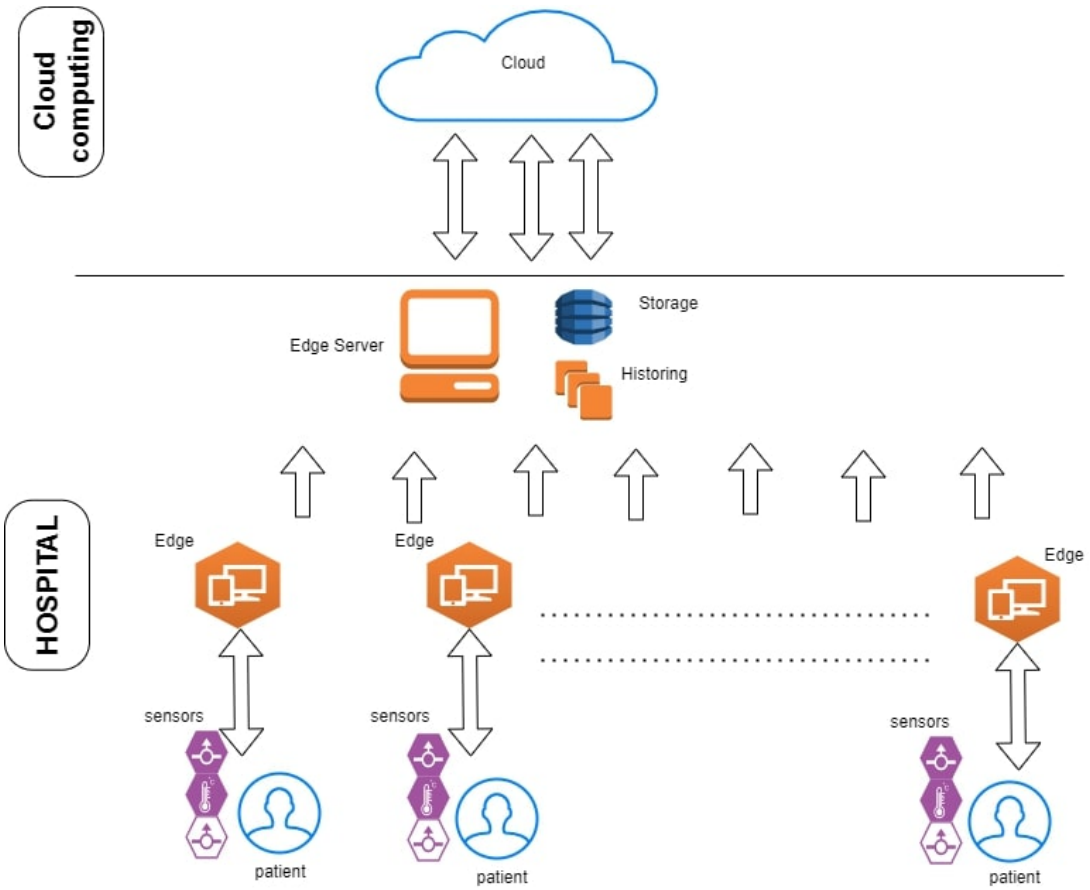

Figure 4.1: System Architecture and Experimental Workflow: From Data Collection to Classification.

4.4 system discription

The architecture of the healthcare edge computing system is designed to provide efficient, real-time monitoring and analysis of patient health data. It consists of several interconnected layers, each performing specific functions to ensure seamless data flow and processing. Below is a detailed description of the key components and their interactions:

4.4.1 Sensing Layer:

- Components: IoT sensors and wearable devices
- Function: This layer includes sensors deployed on patients or in their environment

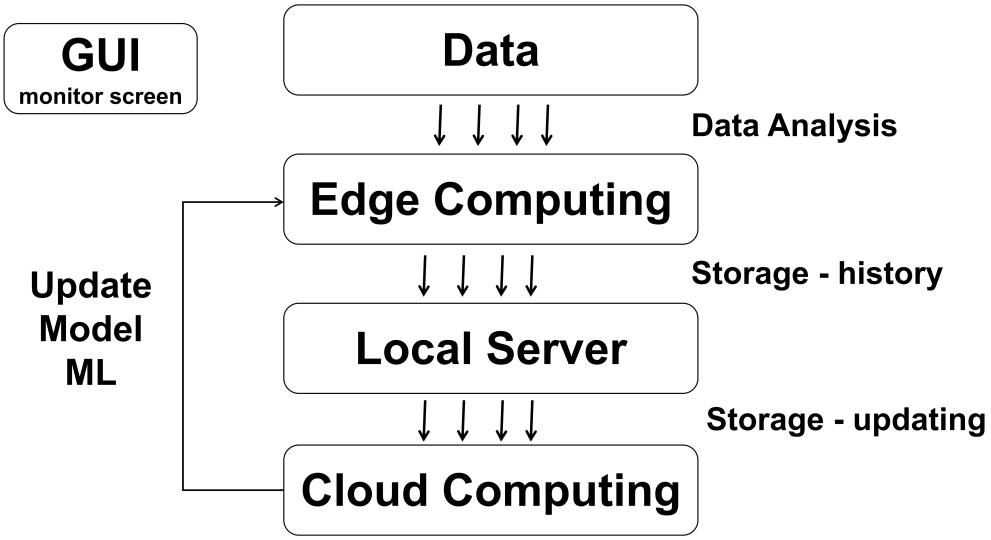

Figure 4.2: Data Flow System.

to continuously monitor vital signs such as heart rate, blood pressure, temperature, and ECG readings. The sensors gather real-time data, which is then transmitted to the edge devices for processing.

- Data Flow: Raw health data is collected and sent to the Edge Layer.

4.4.2 Edge Computing Layer:

- Components: Edge devices, local servers
- Function: This layer is responsible for processing data close to the source, enabling faster analysis and reducing the need for transmitting large volumes of data to a central cloud. It performs tasks such as data cleaning, initial analysis, and real-time alert generation. The edge devices can preprocess and analyze data using machine learning algorithms, allowing for immediate decision-making and response, such as sending emergency alerts.
- Data Flow: Processed data and alerts are generated and can be forwarded to the cloud or sent directly to healthcare providers.

4.4.3 Data Cleaning and Processing Layer:

- Components: Integrated within the Edge and Cloud Layers

- Function: This layer handles data cleaning, removing any irrelevant or erroneous data, and transforming it into structured formats suitable for analysis. The cleaning process ensures data accuracy and reliability, which is essential for making precise health predictions.
- Data Flow: Cleaned and transformed data is passed to the analysis platforms.

4.4.4 Cloud Layer:

- Components: Cloud servers, data storage solutions
- Function: While the edge layer handles real-time processing, the cloud layer serves as a centralized repository for storing larger datasets and performing in-depth analysis. It is used for archiving patient data, training machine learning models on extensive datasets, and supporting long-term data storage. The cloud also provides backup services and ensures data availability across different locations.
- Data Flow: Data from the edge devices is synchronized with the cloud for deeper analysis, long-term storage, and further processing.

4.4.5 Data Analysis and Machine Learning Layer:

- Components: Machine learning algorithms, data mining tools
- Function: This layer focuses on the analysis of patient data using machine learning techniques. The system can predict potential health risks by analyzing patterns in the data, providing insights into patient conditions, and generating recommendations for preventive care. It supports both real-time analysis (on the edge) and more extensive batch processing (on the cloud).
- Data Flow: Analyzed data is used to generate insights, which can be shared with healthcare professionals and the alert system.

4.4.6 Communication and Alert Layer:

- Components: Notification systems, alert mechanisms, communication protocols
- Function: This layer is responsible for issuing alerts based on processed data. If any abnormalities or emergencies are detected (such as a sudden drop in heart rate), the system triggers immediate alerts to notify healthcare professionals or caregivers.

Alerts can be sent through various communication channels, including SMS, email, or mobile app notifications.

- Data Flow: Alerts and insights are directed to healthcare providers, enabling quick responses.

4.4.7 User Interface Layer:

- Components: Graphical User Interfaces (GUIs) for desktops, mobile apps, handheld devices
- Function: This layer provides an interactive platform for doctors, nurses, and other healthcare personnel to access processed data, insights, and patient health records. The GUI is designed to be user-friendly, enabling quick navigation and easy access to critical information, supporting effective decision-making and patient management.
- Data Flow: Users receive data, alerts, and insights directly from the edge or cloud systems via the interface.

4.4.8 Data Flow Overview:

- Collection: Sensors gather data from patients and transmit it to the edge devices.
- Processing: Edge devices process data in real-time, performing cleaning, preliminary analysis, and generating alerts.
- Storage and Advanced Analysis: Processed data is sent to the cloud for storage and in-depth analysis, including training and deploying machine learning models.
- Communication: Alerts and critical insights are communicated to healthcare providers.
- User Access: Healthcare personnel access data and insights via user interfaces on various devices.

This architecture ensures a robust, efficient, and scalable system that enhances patient care through real-time monitoring, predictive analysis, and seamless communication between edge and cloud components.

4.5 Implementation :

The proposed engineering approach in this paper involves deploying an edge device near each patient, along with a local server at the hospital level, and another edge device and server in the cloud to provide continuous monitoring of patient conditions. Various healthcare sensors, such as those for measuring blood glucose, blood pressure, coagulation rate, body temperature, oxygen saturation, air humidity, and room temperature, are attached to the patient. These sensors are directly linked to the edge device, which hosts a machine-learning model responsible for detecting and predicting potential health risks.

Data collected from these sensors, which operate on a 6LoWPAN network, is processed by the edge device in real-time. Additionally, the data is stored temporarily on a local server that maintains comprehensive patient records, including personal details, health history, existing conditions, and prior treatments. This setup ensures that a patient's complete health information is accessible even before they arrive at the hospital. When internet connectivity is available, the local server communicates with the cloud, transferring data using the 6LoWPAN protocol. The machine learning model is implemented both at the edge and in the cloud, with real-time processing and immediate predictions handled at the edge, while more extensive analysis and predictions are performed in the cloud.

The prediction process, whether conducted at the edge or cloud level, focuses on detecting and forecasting heart disease and related risks stemming from changes in various physiological parameters. If a potential health threat is detected, the attending physician is notified through a graphical interface or display screen, accompanied by an alarm to ensure quick response to critical situations. Communication between devices, displays, and interfaces is facilitated by the MQTT protocol, which supports lightweight and efficient messaging in Internet of Things (IoT) environments with limited resources.

Data management and storage on the local server are handled using MySQL, a robust database management system that efficiently organizes, stores, and retrieves data, allowing multiple users to access and manipulate the databases simultaneously.

Overall, this engineering approach enables real-time prediction and early detection of heart disease, even in remote locations. It combines edge and cloud processing, local data storage, and seamless communication between devices and interfaces to enhance patient monitoring and improve healthcare outcomes.

4.6 Machine-Learning Models:

Our research aimed to develop a machine learning algorithm for predicting heart disease by creating and comparing three different models to identify the most accurate one. These algorithms were designed to analyze key data points associated with heart disease, including blood pressure, heart rate, and biliary artery levels, as well as additional patient information such as age, smoking habits, and alcohol consumption. To train and test these models, we used a dataset of around 300 patients from the UCI dataset repository. The algorithms explored in this study included K-Nearest Neighbor, Random Forest, and Gradient Boosting.

1. **K-Nearest Neighbor (KNN) Algorithm: The K-Nearest Neighbor (KNN) algorithm is a supervised learning classifier that relies on proximity to classify or predict the grouping of a data point. Although applicable to both classification and regression tasks, it is primarily used for classification, based on the principle that similar data points tend to be near each other. [120]

2. Random Forest Algorithm: Random Forest is a supervised machine learning method often used for classification and regression tasks. It generates multiple decision trees from various samples, using the average outcome for regression and majority voting for classification. One of its key strengths is its ability to handle datasets containing both continuous (for regression) and categorical (for classification) variables, making it particularly effective for classification problems. [121]

3. Support Vector Machine (SVM): SVM is a supervised learning technique used for both classification and regression tasks. The main goal of the SVM algorithm is to find the optimal line or decision boundary that can divide an n-dimensional space into distinct classes, making it easier to classify new data points. This decision boundary, known as a hyperplane, is determined by selecting extreme vectors and points, called support vectors, which help define and create the hyperplane, forming the core of the SVM method. [122]

4.7 Results and Discusion

Evaluation metrics assist in assessing the performance of different classification models and algorithms.

4.7.1 Confusion Matrix:

The performance of a classification model is assessed using a confusion matrix, which can be utilized to compute various performance metrics such as accuracy, precision, recall, and F1-score. These metrics help evaluate the effectiveness of the classification model.

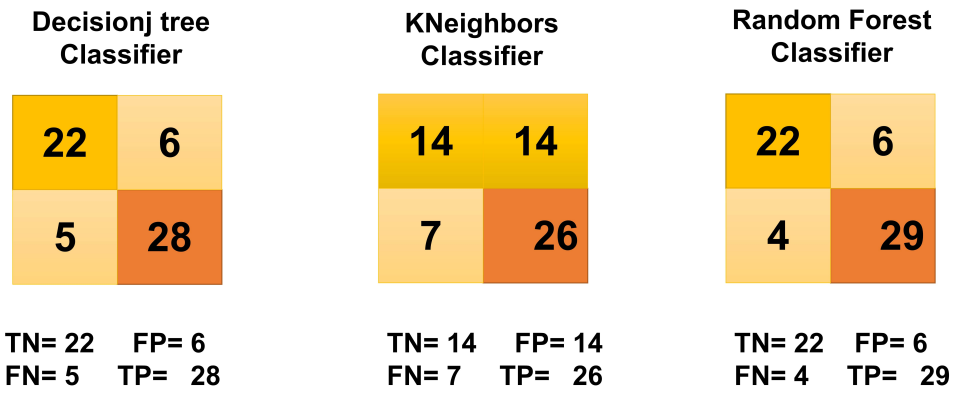

Figure 4.3: confusion matrix.

4.7.2 Accuracy score:

The accuracy score function calculates a classifier's accuracy by determining the ratio of correctly predicted samples to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

4.7.3 Precision-Recall:

When classes are highly imbalanced, precision and recall become important indicators of prediction success. In information retrieval, recall evaluates the number of relevant results that are correctly returned, while precision assesses the relevance of those results.

- The Recall Score refers to the percentage of actual positive events that were accurately predicted, representing the true positive rate.

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

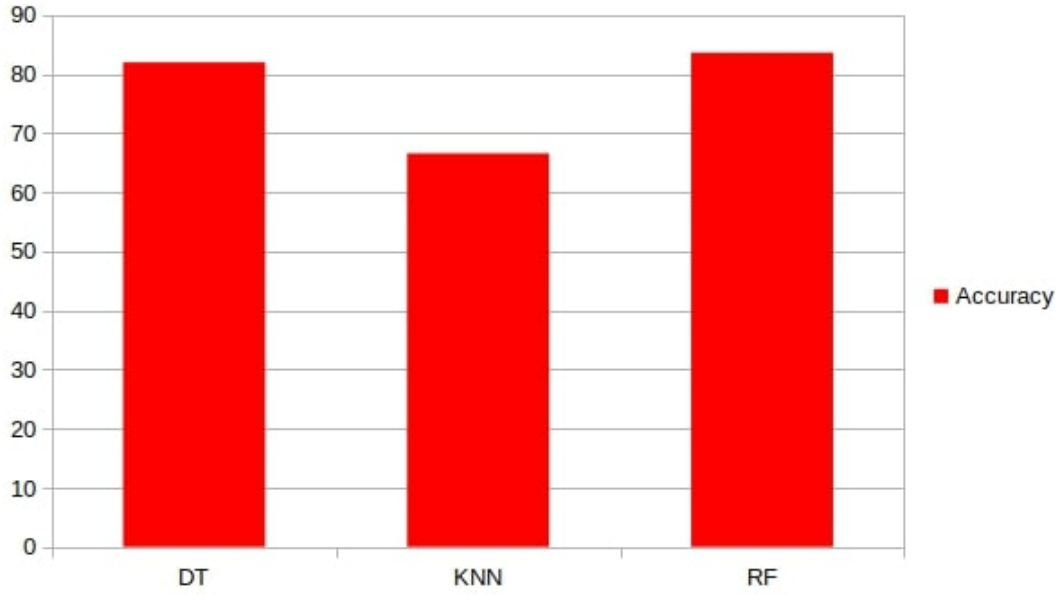

Figure 4.4: Accuracy score.

- The Precision Score indicates the proportion of predicted positive events that are genuinely positive, serving as a measure of accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

4.7.4 F1-Score:

The accuracy of a model on a dataset is gauged by the F-score, also known as the F1-score. Systems for binary classification are evaluated using it:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.7.5 Receiver Operating Characteristics (ROC):

The receiver operating characteristic (ROC) curve is a graphical representation that illustrates the performance of a rating model across various rating thresholds.

4.7.6 Area Under the Curve (AUC) :

The area under the ROC curve is referred to as the Area Under the Curve (AUC) Score. A larger AUC indicates greater accuracy of the test.

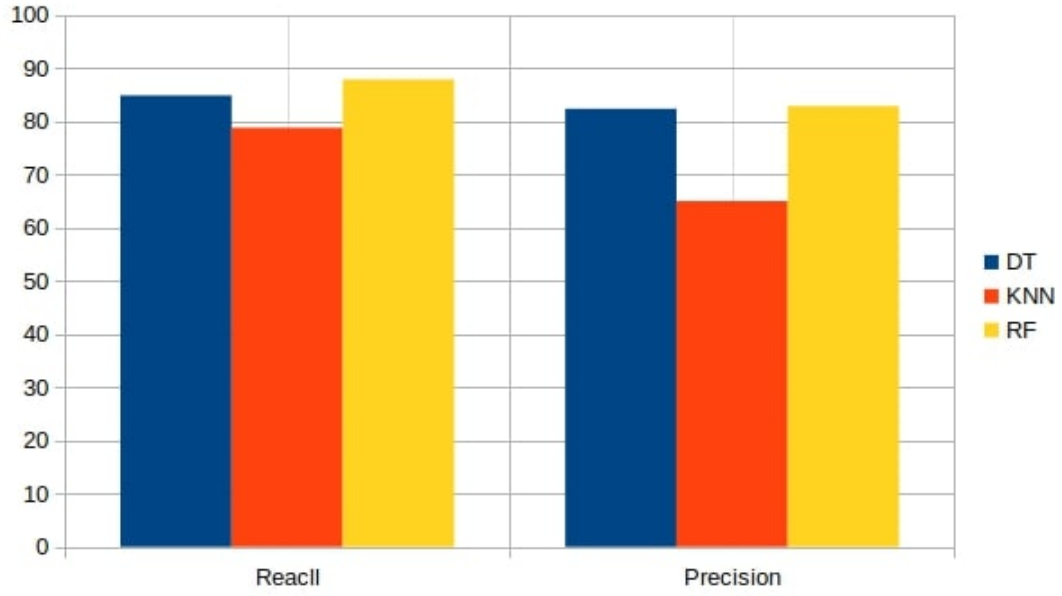

Figure 4.5: Precision-Recall.

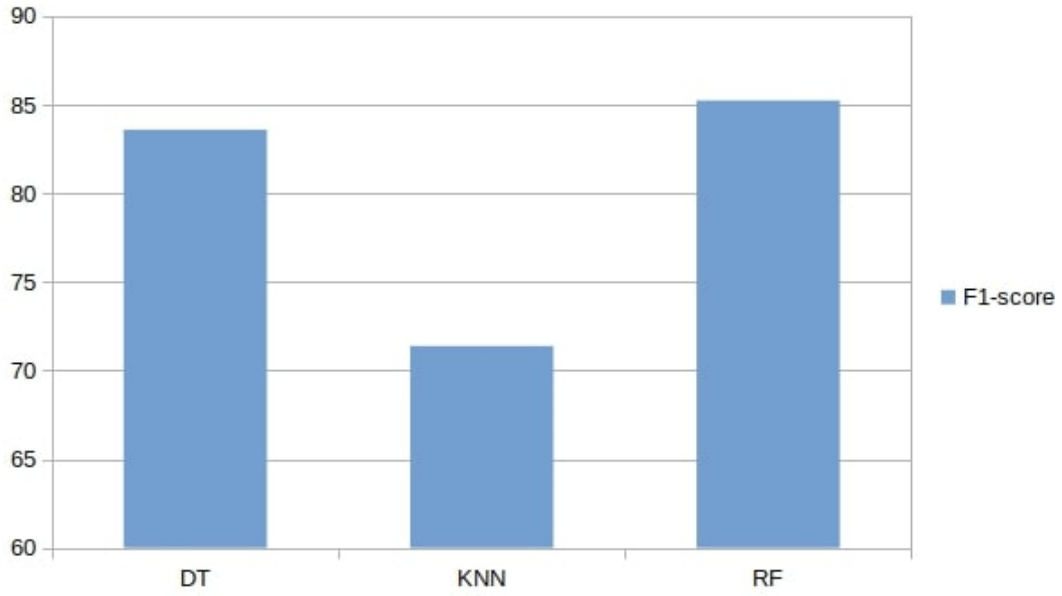

Figure 4.6: F1-Score.

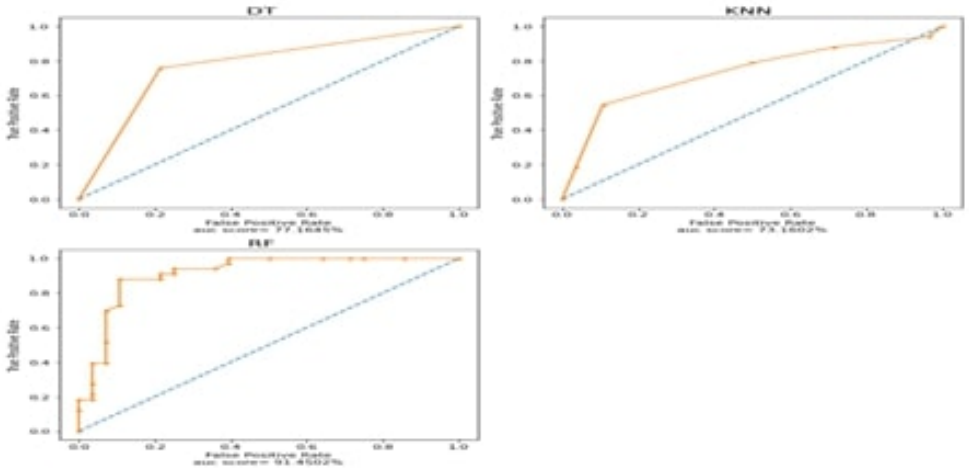

Figure 4.7: ROC result.

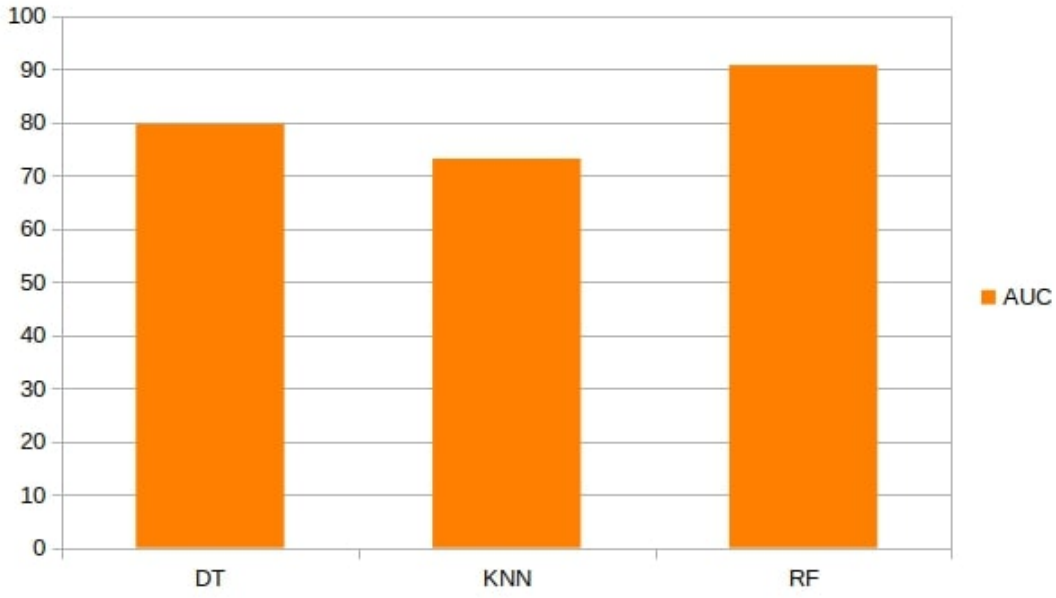

Figure 4.8: AUC result.

4.7.7 Cross Validation:

Cross-validation is a statistical technique employed to assess the performance of machine learning models. Various methods can be used for cross-validation, but they all follow the same basic algorithm:

- Split the dataset into two parts: a training set and a testing set.
- Train the model using the training set.
- Evaluate the model's validity on the testing set.
- Repeat steps 1-3 multiple times. The results can vary depending on the specific approach employed.

The findings revealed that the most accurate and effective Random Forest (RF) algorithm achieved a correct result rate of 91.57%. In comparison, the Decision Tree (DT) algorithm had an acceptable accuracy of 82.63%, while the K-Nearest Neighbors (KNN) algorithm produced a less favorable result of 76.92%, which is notably lower than its predecessors.

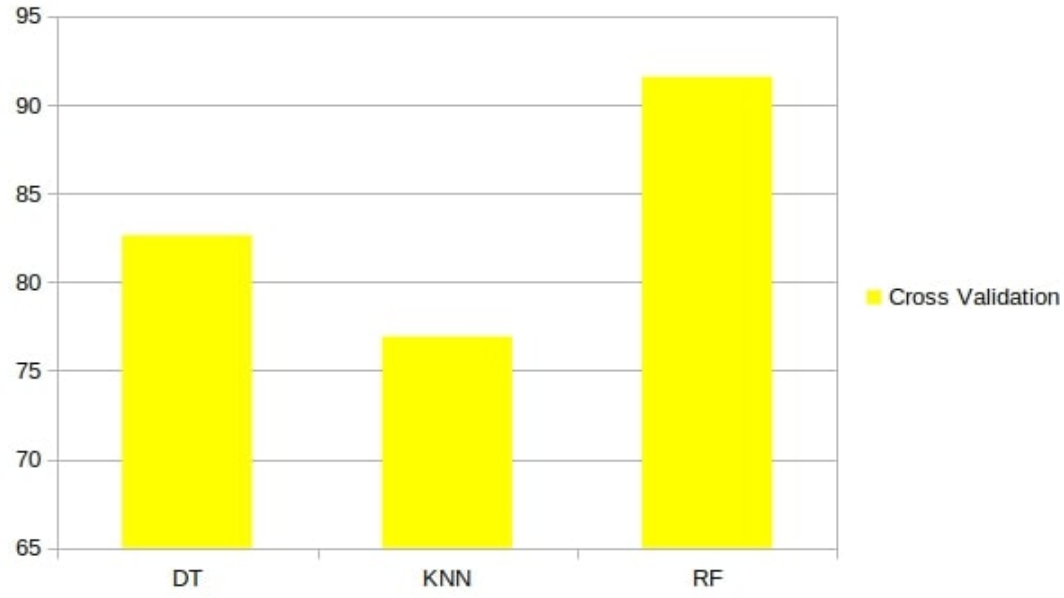

Figure 4.9: Execution Time Result.

4.7.8 Execution Time:

In addition to accuracy, evaluating performance is essential for defining machine learning algorithms. Key performance metrics include training and execution times. The "training time" refers to the duration it takes for a model to train on a dataset, while "execution

time" encompasses all computations involved, such as data splitting, preprocessing, and model evaluation. Figure 11 illustrates the training and execution times for each machine learning algorithm.

The training and execution times for the Random Forest (RF) algorithm are both 97 seconds, making it the slowest algorithm in terms of both metrics, as it builds and computes multiple decision trees (DTs). In contrast, the K-Nearest Neighbors (KNN) algorithm has a training time of 69 seconds. The Decision Tree (DT) algorithm, with a training time of only 42 seconds, is faster to learn and implement than both the KNN and RF algorithms.

Upon analyzing the results from our application of the three classification algorithms, we observed significant differences. The RF algorithm demonstrated clear superiority in performance, achieving classification accuracy exceeding 90%, with sensitivity at 85% and specificity at 88%. To further validate the quality of this model, the ROC curve is also presented. Conversely, the KNN and DT algorithms produced acceptable results in terms of accuracy, precision, and AUC, with results being comparable in some instances. Overall, the preference was given to the RF algorithm based on the prediction results. All findings have been explained and illustrated through the attached graphs.

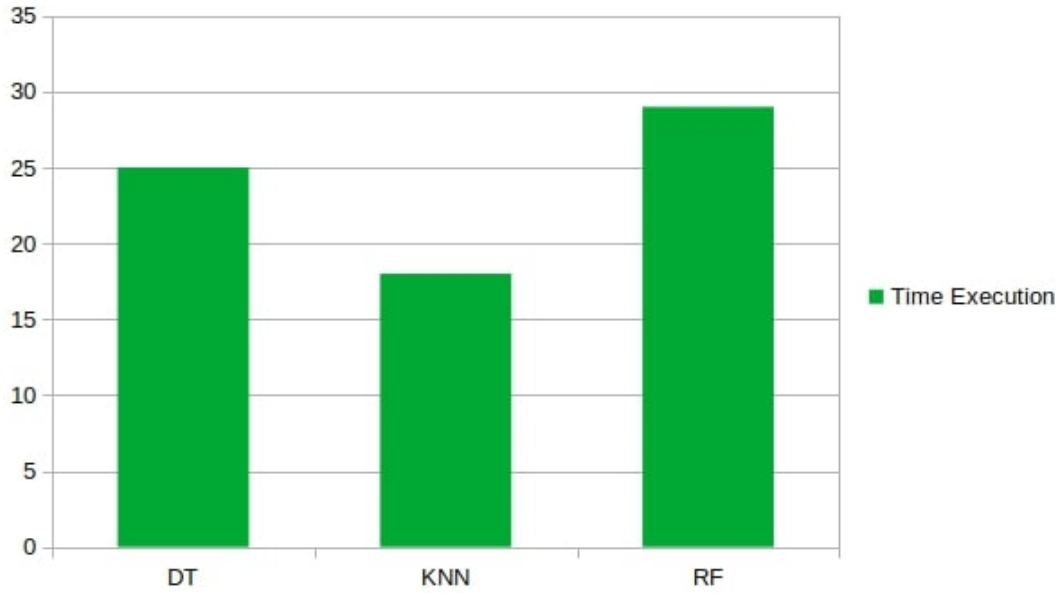

Figure 4.10: Execution Time Result.

4.8 Conclusion :

Heart disease remains a leading cause of mortality, claiming lives on a continuous basis. The collection of accurate patient data and early detection are vital for saving lives. In this study, we aimed to enhance this process by incorporating edge computing techniques for AI predictions, allowing real-time machine learning algorithms to function locally without dependence on cloud communication. Our findings highlighted the superior performance of the Random Forest algorithm, which demonstrated reliable and accurate predictions. To evaluate our architecture, we compared three machine learning algorithms based on two criteria: model performance and execution time. Our primary objective was to develop a machine learning model capable of providing effective real-time results within an edge computing environment. Through the analysis of output charts, we validated the superiority of the Random Forest model.

General conclusion

In this thesis, we have explored the integration of edge computing within the framework of smart cities, proposing three significant contributions aimed at addressing various challenges related to security, efficiency, and reliability. Edge computing represents a paradigm shift from traditional, centralized cloud-based systems to more distributed, localized processing, allowing for faster, more secure, and scalable solutions, particularly suited for the dynamic environments of modern urban infrastructures. Through the proposed contributions, we have demonstrated how edge computing can be effectively utilized to enhance smart city operations, paving the way for more intelligent and responsive urban ecosystems. For this, we proposed three contribution:

Contribution 01 presented a novel engineering model based on hybrid cooperation for secure positioning measurements in autonomous vehicles. In the context of smart cities, where autonomous transportation is expected to play a major role, the security of positioning and communication is paramount. The proposed model leverages inter-vehicle cooperation to provide a robust and reliable solution capable of resisting common threats, such as GPS spoofing, signal jamming, and fraudulent attacks. By securing inter-vehicle communication, our model ensures that autonomous vehicles can operate safely and efficiently, even in challenging urban environments. This approach not only enhances the accuracy of positioning data but also strengthens the overall security framework, making it a crucial contribution to the broader vision of autonomous, connected transportation systems in smart cities.

Contribution 02 focused on enhancing security within edge computing environments by developing robust yet lightweight authentication mechanisms. The challenge of securing edge devices, especially those with limited processing power and resources, requires innovative solutions that do not compromise on performance. Our proposed method addresses this by implementing a strong mutual authentication protocol, specifically designed for low-cost edge devices. The method enables secure and seamless communication between edge nodes and users, providing a reliable verification process that ensures data integrity and privacy. The approach we designed is scalable and efficient, enabling it to be

deployed across a wide array of devices, enhancing the overall security infrastructure of smart city applications. This contribution is particularly relevant as the number of connected devices continues to grow, and the demand for secure, real-time data processing increases.

Contribution 03 addressed one of the key advantages of edge computing over traditional cloud-based systems—real-time data processing. In many scenarios, especially in remote or network-congested areas, edge computing can provide better performance by reducing latency and enhancing the efficiency of data processing. Our work demonstrated that by deploying lightweight machine learning models at the edge, coupled with the optimal selection of algorithms, we can achieve results that are not only faster but also more accurate than those processed via centralized cloud systems. This is particularly valuable in scenarios where real-time decisions are critical, such as traffic management, emergency response, and environmental monitoring. By processing data closer to the source, edge computing reduces the dependency on cloud connectivity, offering a more resilient and reliable solution for smart city applications.

Collectively, the three contributions presented in this thesis illustrate the diverse potential of edge computing to overcome traditional challenges faced by centralized systems. As smart cities continue to develop, the need for efficient, secure, and scalable solutions becomes increasingly important. Edge computing, with its ability to process data locally, ensures lower latency, improved data privacy, and enhanced security. Our proposed approaches demonstrate that edge computing can handle the demands of modern smart city applications, from autonomous vehicles to secure data transmission and real-time analytics.

Furthermore, the integration of advanced technologies such as machine learning and robust encryption protocols within the edge computing framework opens new avenues for innovation. These technologies not only enhance the performance and security of edge computing systems but also provide the necessary tools to develop adaptive and intelligent urban infrastructures. As cities grow and become more complex, the ability to efficiently manage resources, ensure public safety, and provide high-quality services will depend on the seamless integration of such technologies.

Looking forward, the implementation of edge computing in smart cities has the potential to revolutionize urban living. However, it also presents new challenges that must be addressed. Security remains a primary concern, as more devices become interconnected and share sensitive information. Therefore, ongoing research and development are essential to ensure that the technologies deployed are resilient against emerging threats. Additionally, as the scope of edge computing expands, it will be necessary to address issues related to data interoperability, standardization, and infrastructure scalability.

The insights gained from this research indicate that edge computing is not merely a supporting technology but a foundational component of the future of smart cities. To fully realize its potential, stakeholders, including city planners, technology developers, and policymakers, must collaborate to build the necessary infrastructure, promote standardization, and invest in ongoing innovation. The future of smart cities lies in creating a seamless, integrated network of devices and systems that can process information intelligently, adapt to changing circumstances, and provide reliable services to urban residents.

In conclusion, this thesis contributes to the growing field of edge computing by offering practical, secure, and efficient solutions to some of the most pressing challenges in smart city development. Through our proposed models, we have demonstrated that edge computing can effectively address the limitations of traditional cloud-based systems, providing enhanced security, real-time processing capabilities, and greater reliability. As technology continues to evolve, the insights and solutions presented in this work can serve as a foundation for further research and development, helping to shape the future of smart, connected urban environments.

Future works

Based on the work completed, the following directions for future studies are being pursued:

First, regarding the GPS spoofing solution for self-driving cars, future work involves conducting field tests to verify the effectiveness of the proposed model against GPS spoofing attacks in real-world conditions. While the current approach shows promise in theory and controlled environments, practical demonstrations are necessary to confirm its reliability under diverse circumstances. Additionally, we plan to experimentally evaluate the timing performance of the algorithms, with a particular focus on latency measurements. This evaluation will help assess the feasibility and effectiveness of the approach in real-world scenarios, ensuring that it can respond quickly to threats without introducing significant latency.

For the password-based authentication protocol in edge computing, we aim to improve the current model by incorporating anonymity features. While the existing solution effectively ensures secure and efficient data sharing across IoT systems, adding anonymity would further protect user identities and enhance overall privacy. This enhancement would make the system more robust against potential threats and broaden its application in sensitive environments where data privacy is a priority.

Finally, in the context of heart disease prediction using edge computing, future work will focus on enhancing the edge-based architecture for real-time AI predictions. Although the random forest algorithm has demonstrated reliable performance, there is room for further improvement. Future studies may explore other machine learning models or techniques to improve accuracy and efficiency within the edge computing environment. This ongoing development will aim to make real-time health monitoring systems more efficient, enabling faster and more accurate predictions, with an emphasis on incorporating deep learning algorithms to achieve more robust and precise results.

Bibliographie

[1] Imed Ben Dhaou, Mousameh Ebrahimi, Meriam Ben Ammar, Ghada Bouattour, and Olfa Kanoun. Edge devices for internet of medical things : technologies, techniques, and implementation. *Electronics*, 10(17) :2104, 2021.

[2] Ricardo S Alonso, Inés Sittón-Candanedo, Óscar García, Javier Prieto, and Sara Rodríguez-González. An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario. *Ad Hoc Networks*, 98 :102047, 2020.

[3] Malong Ke, Zhen Gao, Yang Huang, Guoru Ding, Derrick Wing Kwan Ng, Qihui Wu, and Jun Zhang. An edge computing paradigm for massive iot connectivity over high-altitude platform networks. *IEEE Wireless Communications*, 28(5) :102–109, 2021.

[4] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing : A survey. *Future Generation Computer Systems*, 97 :219–235, 2019.

[5] Marc Timme. Revealing network connectivity from response dynamics. *Physical review letters*, 98(22) :224101, 2007.

[6] Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong. Big data analytics for intelligent manufacturing systems : A review. *Journal of Manufacturing Systems*, 62 :738–752, 2022.

[7] Kornelia Batko and Andrzej Ślęzak. The use of big data analytics in healthcare. *Journal of big Data*, 9(1) :3, 2022.

[8] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2) :293–314, 2014.

[9] John T Hancock and Taghi M Khoshgoftaar. Catboost for big data : an interdisciplinary review. *Journal of big data*, 7(1) :94, 2020.

[10] Dilpreet Singh and Chandan K Reddy. A survey on platforms for big data analytics. *Journal of big data*, 2 :1–20, 2015.

[11] Jasmine Zakir, Tom Seymour, and Kristi Berg. Big data analytics. *Issues in Information Systems*, 16(2), 2015.

[12] Shivaji P Mirashe and Namdeo V Kalyankar. Cloud computing. *arXiv preprint arXiv :1003.4074*, 2010.

[13] Ali Sunyaev and Ali Sunyaev. Cloud computing. *Internet computing : Principles of distributed systems and emerging internet-based technologies*, pages 195–236, 2020.

[14] Nick Antonopoulos and Lee Gillam. *Cloud computing*, volume 51. Springer, 2010.

[15] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, and Zhenghu Gong. The characteristics of cloud computing. In *2010 39th International Conference on Parallel Processing Workshops*, pages 275–279. IEEE, 2010.

[16] Asif Ali Laghari, Kaishan Wu, Rashid Ali Laghari, Mureed Ali, and Abdullah Ayub Khan. A review and state of art of internet of things (iot). *Archives of Computational Methods in Engineering*, pages 1–19, 2021.

[17] Radouan Ait Radouan Ait Mouha et al. Internet of things (iot). *Journal of Data Analysis and Information Processing*, 9(02) :77, 2021.

[18] Vijay K Garg. *Elements of distributed computing*. John Wiley & Sons, 2002.

[19] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation : numerical methods*. Athena Scientific, 2015.

[20] Mohammed Maray and Junaid Shuja. [retracted] computation offloading in mobile cloud computing and mobile edge computing : Survey, taxonomy, and open issues. *Mobile Information Systems*, 2022(1) :1121822, 2022.

[21] Xiaomin Jin, Wenqiang Hua, Zhongmin Wang, and Yanping Chen. A survey of research on computation offloading in mobile cloud computing. *Wireless Networks*, 28(4) :1563–1585, 2022.

[22] Shahab Shamshirband, Mahdis Fathi, Anthony T Chronopoulos, Antonio Montieri, Fabio Palumbo, and Antonio Pescapè. Computational intelligence intrusion detection techniques in mobile cloud computing environments : Review, taxonomy, and open research issues. *Journal of Information Security and Applications*, 55 :102582, 2020.

[23] Benjamin K Sovacool and Dylan D Furszyfer Del Rio. Smart home technologies in europe : A critical review of concepts, benefits, risks and policies. *Renewable and sustainable energy reviews*, 120 :109663, 2020.

[24] Yolande Strengers and Jenny Kennedy. *The smart wife : Why Siri, Alexa, and other smart home devices need a feminist reboot*. Mit Press, 2021.

[25] Davide Ferraris, Daniel Bastos, Carmen Fernandez-Gago, and Fadi El-Moussa. A trust model for popular smart home devices. *International Journal of Information Security*, 20 :571–587, 2021.

[26] Eunil Park, Yongwoo Cho, Jinyoung Han, and Sang Jib Kwon. Comprehensive approaches to user acceptance of internet of things in a smart home environment. *IEEE Internet of Things Journal*, 4(6) :2342–2350, 2017.

[27] Olutosin Taiwo, Absalom E Ezugwu, Olaide N Oyelade, and Mubarak S Almutairi. Enhanced intelligent smart home control and security system based on deep learning model. *Wireless communications and mobile computing*, 2022(1) :9307961, 2022.

[28] Sungjin Yu, Namsu Jho, and Youngho Park. Lightweight three-factor-based privacy-preserving authentication scheme for iot-enabled smart homes. *IEEE Access*, 9 :126186–126197, 2021.

[29] Amjad Qashlan, Priyadarsi Nanda, Xiangjian He, and Manoranjan Mohanty. Privacy-preserving mechanism in smart home using blockchain. *IEEE Access*, 9 :103651–103669, 2021.

[30] Chanthaphone Sisavath and Lasheng Yu. Design and implementation of security system for smart home based on iot technology. *Procedia Computer Science*, 183 :4–13, 2021.

[31] Kari Dahlgren, Sarah Pink, Yolande Strengers, Larissa Nicholls, and Jathan Sadowski. Personalization and the smart home : Questioning techno-hedonist imaginaries. *Convergence*, 27(5) :1155–1169, 2021.

[32] Abdulkadir Gozuoglu, Okan Ozgonenel, and Cenk Gezegin. Cnn-lstm based deep learning application on jetson nano : Estimating electrical energy consumption for future smart homes. *Internet of Things*, page 101148, 2024.

[33] Lawal Babangida, Thinagaran Perumal, Norwati Mustapha, and Razali Yaakob. Internet of things (iot) based activity recognition strategies in smart homes : A review. *IEEE Sensors Journal*, 22(9) :8327–8336, 2022.

[34] Abiodun E Amoran, Ayodele S Oluwole, Enitan O Fagorola, and RS Diarah. Home automated system using bluetooth and an android application. *Scientific African*, 11 :e00711, 2021.

[35] Cristina Stolojescu-Crisan, Calin Crisan, and Bogdan-Petru Butunoi. An iot-based smart home automation system. *Sensors*, 21(11) :3784, 2021.

[36] Hyung Woo Choi, Dong-Wook Shin, Jiajie Yang, Sanghyo Lee, Cátia Figueiredo, Stefano Sinopoli, Kay Ullrich, Petar Jovančić, Alessio Marrani, Roberto Momentè, et al. Smart textile lighting/display system with multifunctional fibre devices for large scale smart home and iot applications. *Nature communications*, 13(1) :814, 2022.

[37] Mohamed Abdel-Basset, Gunasekaran Manogaran, Abduallah Gamal, and Victor Chang. A novel intelligent medical decision support model based on soft computing and iot. *IEEE Internet of Things Journal*, 7(5) :4160–4170, 2019.

[38] Chao Li, Xiangpei Hu, and Lili Zhang. The iot-based heart disease monitoring system for pervasive healthcare service. *Procedia computer science*, 112 :2328–2334, 2017.

[39] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. An overview of clinical decision support systems : benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1) :17, 2020.

[40] Ali Hassan Sodhro, Zongwei Luo, Arun Kumar Sangaiah, and Sung Wook Baik. Mobile edge computing based qos optimization in medical healthcare applications. *International Journal of Information Management*, 45 :308–318, 2019.

[41] Shangguang Wang, Yali Zhao, Jinlinag Xu, Jie Yuan, and Ching-Hsien Hsu. Edge server placement in mobile edge computing. *Journal of Parallel and Distributed Computing*, 127 :160–168, 2019.

[42] Frédérick Garcia and Emmanuel Rachelson. Markov decision processes. *Markov Decision Processes in Artificial Intelligence*, pages 1–38, 2013.

[43] Soraia Oueida, Yehia Kotb, Moayad Aloqaily, Yaser Jararweh, and Thar Baker. An edge computing based smart healthcare framework for resource management. *Sensors*, 18(12) :4307, 2018.

[44] Ali Hassan Sodhro, Abdul Sattar Malokani, Gul Hassan Sodhro, Muhammad Muzammal, and Luo Zongwei. An adaptive qos computation for medical data processing in intelligent healthcare applications. *Neural computing and applications*, 32 :723–734, 2020.

[45] Ali Hassan Sodhro and Giancarlo Fortino. Energy management during video transmission in wireless body sensor networks. In *2017 IEEE 14th international conference on networking, sensing and control (ICNSC)*, pages 655–660. IEEE, 2017.

[46] Yasuhiro Nagai, Takao Okamawari, and Teruya Fujii. A novel streaming method using qos control function of lte to prevent video freezing. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE, 2016.

[47] Nafisah Kheshaifaty and Adnan Gutub. Engineering graphical captcha and aes crypto hash functions for secure online authentication. *Journal of Engineering Research*, 11(3), 2023.

[48] Nafisah Kheshaifaty and Adnan Gutub. Preventing multiple accessing attacks via efficient integration of captcha crypto hash functions. *Int. J. Comput. Sci. Netw. Secure.(IJCSNS)*, 20(9) :16–28, 2020.

[49] Adnan Gutub and Nafisah Kheshaifaty. Practicality analysis of utilizing text-based captcha vs. graphic-based captcha authentication. *Multimedia Tools and Applications*, 82(30) :46577–46609, 2023.

[50] Shaoyong Guo, Xing Hu, Song Guo, Xuesong Qiu, and Feng Qi. Blockchain meets edge computing : A distributed and trusted authentication system. *IEEE Transactions on Industrial Informatics*, 16(3) :1972–1983, 2019.

[51] Ashish Singh, Adnan Gutub, Anand Nayyar, and Muhammad Khurram Khan. Redefining food safety traceability system through blockchain : findings, challenges, and open issues. *Multimedia Tools and Applications*, 82(14) :21243–21277, 2023.

[52] Khaled Chait, Abdelkader Laouid, Mostefa Kara, Mohammad Hammoudeh, Omar Aldabbas, and Abdullah T Al-Essa. An enhanced rsa-based aggregate signature scheme to reduce blockchain size. *IEEE Access*, 2023.

[53] Avleen Malhi and Shalini Batra. Privacy-preserving authentication framework using bloom filter for secure vehicular communications. *International Journal of Information Security*, 15 :433–453, 2016.

[54] Jing Zhang, Hong Zhong, Jie Cui, Miaomiao Tian, Yan Xu, and Lu Liu. Edge computing-based privacy-preserving authentication framework and protocol for 5g-enabled vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(7) :7940–7954, 2020.

[55] Shunrong Jiang, Xiaoyan Zhu, and Liangmin Wang. An efficient anonymous batch authentication scheme based on hmac for vanets. *IEEE Transactions on Intelligent Transportation Systems*, 17(8) :2193–2204, 2016.

[56] Maria Azees, Pandi Vijayakumar, and Lazarus Jegatha Deboarh. Eaap : Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(9) :2467–2476, 2017.

[57] Jing Wang, Libing Wu, Kim-Kwang Raymond Choo, and Debiao He. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Transactions on Industrial Informatics*, 16(3) :1984–1992, 2019.

[58] Khaled Chait, Mostefa Kara, Abdelkader Laouid, Mohammad Hammoudeh, and Ahcène Bounceur. One digit checksum for data integrity verification of cloud-

executed homomorphic encryption operations. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 71–75, 2023.

[59] Debiao He, Huaqun Wang, Muhammad Khurram Khan, and Lina Wang. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. *Iet Communications*, 10(14) :1795–1802, 2016.

[60] Pardeep Kumar, Andrei Gurtov, Mangal Sain, Andrew Martin, and Phuong H Ha. Lightweight authentication and key agreement for smart metering in smart energy networks. *IEEE Transactions on Smart Grid*, 10(4) :4349–4359, 2018.

[61] Mostefa Kara, Konstantinos Karampidis, Giorgos Papadourakis, Abdelkader Laouid, and Muath AlShaikh. A probabilistic public-key encryption with ensuring data integrity in cloud computing. In *2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 59–66. IEEE, 2023.

[62] Muath AlShaikh, Malek Alzaqebah, Nabil Gmati, Nashat Alrefai, Mutasem K Alsmadi, Ibrahim Almarashdeh, Rami Mustafa A Mohammad, Sultan Alamri, and Mostefa Kara. Image encryption algorithm based on factorial decomposition. *Multimedia Tools and Applications*, pages 1–21, 2024.

[63] Xiaoying Jia, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Systems Journal*, 14(1) :560–571, 2019.

[64] Run-Fa Liao, Hong Wen, Jinsong Wu, Fei Pan, Aidong Xu, Huanhuan Song, Feiyi Xie, Yixin Jiang, and Minggui Cao. Security enhancement for mobile edge computing through physical layer authentication. *IEEE Access*, 7 :116390–116401, 2019.

[65] Jen-Ho Yang and Chin-Chen Chang. An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Computers & security*, 28(3-4) :138–143, 2009.

[66] Eun-Jun Yoon and Kee-Young Yoo. Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc. In *2009 International conference on computational science and engineering*, volume 2, pages 633–640. IEEE, 2009.

[67] Jia-Lun Tsai and Nai-Wei Lo. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE systems journal*, 9(3) :805–815, 2015.

[68] Qi Jiang, Jianfeng Ma, and Fushan Wei. On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE systems journal*, 12(2) :2039–2042, 2016.

[69] Kuljeet Kaur, Sahil Garg, Georges Kaddoum, Mohsen Guizani, and Dushantha Nalin K Jayakody. A lightweight and privacy-preserving authentication protocol for mobile edge computing. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[70] Guanjie Cheng, Yan Chen, Shuiguang Deng, Honghao Gao, and Jianwei Yin. A blockchain-based mutual authentication scheme for collaborative edge computing. *IEEE Transactions on Computational Social Systems*, 9(1) :146–158, 2021.

[71] Neyire Deniz Sarier. Multimodal biometric authentication for mobile edge computing. *Information Sciences*, 573 :82–99, 2021.

[72] Jin Cui, Lin Shen Liew, Giedre Sabaliauskaite, and Fengjun Zhou. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks*, 90 :101823, 2019.

[73] Ashish Nanda, Deepak Puthal, Joel JPC Rodrigues, and Sergei A Kozlov. Internet of autonomous vehicles communications security : overview, issues, and directions. *IEEE Wireless Communications*, 26(4) :60–65, 2019.

[74] Wenyuan Xu, Chen Yan, Weibin Jia, Xiaoyu Ji, and Jianhao Liu. Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles. *IEEE Internet of Things Journal*, 5(6) :5015–5029, 2018.

[75] Raj Gautam Dutta, Feng Yu, Teng Zhang, Yaodan Hu, and Yier Jin. Security for safety : A path toward building trusted autonomous vehicles. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–6. IEEE, 2018.

[76] Sagar Dasgupta, Abdullah Ahmed, Mizanur Rahman, and Thejesh N Bandi. Unveiling the stealthy threat : Analyzing slow drift gps spoofing attacks for autonomous vehicles in urban environments and enabling the resilience. *arXiv preprint arXiv :2401.01394*, 2024.

[77] Sagar Dasgupta, Tonmoy Ghosh, and Mizanur Rahman. A reinforcement learning approach for global navigation satellite system spoofing attack detection in autonomous vehicles. *Transportation research record*, 2676(12) :318–330, 2022.

[78] Sagar Dasgupta, Mizanur Rahman, Mhafuzul Islam, and Mashrur Chowdhury. Prediction-based gnss spoofing attack detection for autonomous vehicles. *arXiv preprint arXiv :2010.11722*, 2020.

[79] Ziyuan Gu, Zelin Wang, Zhiyuan Liu, and Meead Saberi. Network traffic instability with automated driving and cooperative merging. *Transportation Research Part C : Emerging Technologies*, 138 :103626, 2022.

[80] Aissaoua Habib, Abdelkader Laouid, and Mostefa Kara. Secure consensus clock synchronization in wireless sensor networks. In *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, pages 1–6. IEEE, 2021.

[81] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6) :60–68, 2015.

[82] Xianbin Cao, Peng Yang, Mohamed Alzenad, Xing Xi, Dapeng Wu, and Halim Yanikomeroglu. Airborne communication networks : A survey. *IEEE Journal on Selected Areas in Communications*, 36(9) :1907–1926, 2018.

[83] Yair Wiseman. Autonomous vehicles. In *Research anthology on cross-disciplinary designs and applications of automation*, pages 878–889. Igi Global, 2022.

[84] Joanna Moody, Nathaniel Bailey, and Jinhua Zhao. Public perceptions of autonomous vehicle safety : An international comparison. *Safety science*, 121 :634–650, 2020.

[85] Khattab M Ali Alheeti, Abdulkareem Alzahrani, and Duaa Al Dosary. Lidar spoofing attack detection in autonomous vehicles. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2. IEEE, 2022.

[86] Rony Komissarov and Avishai Wool. Spoofing attacks against vehicular fmcw radar. In *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, pages 91–97, 2021.

[87] Mostefa Kara, Konstantinos Karampidis, Zaoui Sayah, Abdelkader Laouid, Giorgos Papadourakis, and Mohammad Nadir Abid. A password-based mutual authentication protocol via zero-knowledge proof solution. In *International Conference on Applied CyberSecurity*, pages 31–40. Springer, 2023.

[88] Mostefa Kara, Abdelkader Laouid, and Mohammad Hammoudeh. An efficient multi-signature scheme for blockchain. *Cryptology ePrint Archive*, 2023.

[89] Mostefa Kara, Abdelkader Laouid, Mohammad Hammoudeh, Ahcène Bounceur, et al. One digit checksum for data integrity verification of cloud-executed homomorphic encryption operations. *Cryptology ePrint Archive*, 2023.

[90] Sangwan Lee. Opinions of active transportation users on policies to ensure their perceived safety in the era of autonomous vehicles. *Case studies on transport policy*, 12 :101002, 2023.

[91] Eva Papadogiannaki and Sotiris Ioannidis. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Computing Surveys (CSUR)*, 54(6) :1–35, 2021.

[92] Patrick McEnroe, Shen Wang, and Madhusanka Liyanage. A survey on the convergence of edge computing and ai for uavs : Opportunities and challenges. *IEEE Internet of Things Journal*, 2022.

[93] Waqas Ahmad, Aamir Rasool, Abdul Rehman Javed, Thar Baker, and Zunera Jalil. Cyber security in iot-based cloud computing : A comprehensive survey. *Electronics*, 11(1) :16, 2022.

[94] Bin Liu, Zhongqiang Luo, Hongbo Chen, and Chengjie Li. A survey of state-of-the-art on edge computing : Theoretical models, technologies, directions, and development paths. *IEEE Access*, 10 :54038–54063, 2022.

[95] Ashish Singh, Suresh Chandra Satapathy, Arnab Roy, and Adnan Gutub. Ai-based mobile edge computing for iot : Applications, challenges, and future scope. *Arabian Journal for Science and Engineering*, 47(8) :9801–9831, 2022.

[96] Sabur Baidya, Yu-Jen Ku, Hengyu Zhao, Jishen Zhao, and Sujit Dey. Vehicular and edge computing for emerging connected and autonomous vehicle applications. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

[97] Cheng Feng, Yi Wang, Qixin Chen, Yi Ding, Goran Strbac, and Chongqing Kang. Smart grid encounters edge computing : Opportunities and applications. *Advances in Applied Energy*, 1 :100006, 2021.

[98] Muneera Alotaibi, Daniah Al-hendi, Budoor Alroithy, Manal AlGhamdi, and Adnan Gutub. Secure mobile computing authentication utilizing hash, cryptography, and steganography combination. *Journal of Information Security and Cybercrimes Research*, 2(1) :73–82, 2019.

[99] Adnan Gutub. Adjusting counting-based secret-sharing via personalized passwords and email-authentic reliability. *Journal of Engineering Research*, 12(1) :107–121, 2024.

[100] Adnan Gutub and Esraa Almehmadi. Arabic text watermarking tuned for medical e-record semi-authentication. *Journal of Engineering Research*, 11(3), 2023.

[101] Adnan Gutub. Watermarking images via counting-based secret sharing for lightweight semi-complete authentication. *International Journal of Information Security and Privacy (IJISP)*, 16(1) :1–18, 2022.

[102] Dongqi Liu, Haolan Liang, Xiangjun Zeng, Qiong Zhang, Zidong Zhang, and Minhong Li. Edge computing application, architecture, and challenges in ubiquitous power internet of things. *Frontiers in Energy Research*, 10 :850252, 2022.

[103] Dhasarathan Chandramohan, Dananjayan Sathian, Dayalan Rajaguru, Thirumal Vengattaraman, and Ponnurangam Dhavachelvan. A multi-agent approach : To preserve user information privacy for a pervasive and ubiquitous environment. *Egyptian Informatics Journal*, 16(1) :151–166, 2015.

[104] Norah Farooqi, Adnan Gutub, and Mohamed Osama Khozium. Smart community challenges : enabling iot/m2m technology case study. *Life Science Journal*, 16(7) :11–17, 2019.

[105] Maurantonio Caprolu, Roberto Di Pietro, Flavio Lombardi, and Simone Raponi. Edge computing perspectives : architectures, technologies, and open security issues. In *2019 IEEE International Conference on Edge Computing (EDGE)*, pages 116–123. IEEE, 2019.

[106] Kebache Romaissa, Laouid Abdelkader, Bounceur Ahcene, Kara Mostefa, Karampidis Konstantinos, Papadourakis Giorgos, and Hammoudeh Mohammad. Reducing the encrypted data size : Healthcare with iot-cloud computing applications. *Computer Systems Science & Engineering*, 48(4) :1055–1072, 2024.

[107] Saci Medileh, Mostefa Kara, Abdelkader Laouid, Ahcene Bounceur, and Ismail Kertiou. A secure clock synchronization scheme in wsns adapted for iot-based applications. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 674–681, 2023.

[108] Khalid Mahmood, Xiong Li, Shehzad Ashraf Chaudhry, Husnain Naqvi, Saru Kumari, Arun Kumar Sangaiah, and Joel JPC Rodrigues. Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure. *Future Generation Computer Systems*, 88 :491–500, 2018.

[109] Arij Ben Amor, Mohamed Abid, and Aref Meddeb. A privacy-preserving authentication scheme in an edge-fog environment. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 1225–1231. IEEE, 2017.

[110] Chenyu Wang, Ding Wang, Guoai Xu, and Debiao He. Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0. *Science China Information Sciences*, 65(1) :112301, 2022.

[111] Dipanwita Sadhukhan, Sangram Ray, GP Biswas, Muhammad Khurram Khan, and Mou Dasgupta. A lightweight remote user authentication scheme for iot communi-

cation using elliptic curve cryptography. *The Journal of Supercomputing*, 77 :1114–1151, 2021.

[112] Sandeep Reddy, Sonia Allan, Simon Coghlan, and Paul Cooper. A governance model for the application of ai in health care. *Journal of the American Medical Informatics Association*, 27(3) :491–497, 2020.

[113] Roberto Moro Visconti and Donato Morea. Healthcare digitalization and pay-for-performance incentives in smart hospital project financing. *International Journal of environmental research and Public Health*, 17(7) :2318, 2020.

[114] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE access*, 8 :85714–85728, 2020.

[115] Wei Zhang, Xiaohui Chen, Yueqi Liu, and Qian Xi. A distributed storage and computation k-nearest neighbor algorithm based cloud-edge computing for cyber-physical-social systems. *Ieee Access*, 8 :50118–50130, 2020.

[116] Peng Liu, Yifan Zhang, Huifeng Wu, and Tingting Fu. Optimization of edge-plc-based fault diagnosis with random forest in the industrial internet of things. *IEEE Internet of Things Journal*, 7(10) :9664–9674, 2020.

[117] Tiantian Xu, Guangjie Han, Xingyue Qi, Jiaxin Du, Chuan Lin, and Lei Shu. A hybrid machine learning model for demand prediction of edge-computing-based bike-sharing system using internet of things. *IEEE Internet of Things Journal*, 7(8) :7345–7356, 2020.