People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Kasdi Merbah Ouargla
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technologies

A Thesis for the Degree of Master LMD
Domain: Computer Science
Specialty: Industrial Computer Science

Title

Optimizing MEC Deployment Using Vehicle Density: A Deep Learning Approach

Presented by:
BoussaidHadjer.
DjoudiDalale.
Reguig Nada.

Supervised by:
Dr. Mohamed Ben Bezziane
Dr. BasmaHamrouni

Jury Members

Pr.	Mohamed Lamine Kherfi	Examiner	Ouargla University
Dr.	Adel Zega	President	Ouargla University
Dr.	Yacine khaldi	Examiner	Ouargla University

Academic year: 2023/2024

DEDICATION

إلى رفيقتيَّ في الخطوة الأولى والخطوة ما قبل الأخيرة، إلى من كنتما في السنوات العجاف سحابًا ممطرًا، أنا ممتنة لكل لحظة قضيناها معًا.

إلى الصديقتين اللتين لا تزالان معي حتى الآن، أشعر بالفخر حين أقول "صديقتاي منذ سنوات"، أحبكما بقدر الأيام والضحكات التي جمعتنا.

ولأن الصداقة حظ ورزق، أشكر الله أنه رزقني إياكما. شكرًا لكما من القلب.

من حسن الحظ انكما صديقتاي.

Acknowledgments

رَبِّ أَوْزِعْنِىٓ أَنْ أَشْكُرَ نِعْمَتَكَ ٱلَّتِىٓ أَنْعَمْتَ عَلَىَّ وَعَلَىٰ وَٰلِدَىَّ وَأَنْ أَعْمَلَ
صَٰلِحًا تَرْضَىٰهُ وَأَدْخِلْنِى بِرَحْمَتِكَ فِى عِبَادِكَ ٱلصَّٰلِحِينَ - النمل - **19**

First, many thanks go to **Allah** the Almighty for giving us the strength, courage, and health necessary to accomplish this work.

We would like to express our sincere gratitude to Dr. **Mohamed Ben Bezziane**, our supervisor, for their invaluable guidance, support, and encouragement throughout the entire process of conducting this research and writing this thesis.

We also thank our dear parents for their support from the beginning, sacrifices, encouragement, and unconditional love without which we would never have been able to complete our work.

A very big thank you to our dear brothers and sisters. To all friends, To all our professors and colleagues.

To all who contributed directly or indirectly to the completion of this work.

Abstract

This thesis explores the application of Long Short-Term Memory (LSTM) models to predict vehicle trajectories, aiming to optimize urban transportation systems and enhance the deployment of Mobile Edge Computing (MEC) units in smart city environments. The research utilizes a comprehensive dataset comprising GPS coordinates of taxis in San Francisco, processed into time series data for predictive modeling. Through meticulous experimentation, the thesis demonstrates the superior performance of LSTM models over traditional Simple Recurrent Neural Networks (RNNs) in predicting vehicle positions with high accuracy. These predictions are crucial for determining optimal locations for MEC units, which are strategically hosted on or near 5G antennas to improve data throughput and reduce communication latency. By clustering the predicted positions, the study provides a framework for efficient MEC placement, contributing to the reduction of network congestion and enhancement of urban infrastructure management. This work not only highlights the capabilities of LSTM models in real-time data analysis but also underscores their potential in supporting the infrastructure for next-generation wireless networks in urban settings.

Keywords: Long Short-Term Memory (LSTM), Mobile Edge Computing (MEC), dataset, GPS, Simple Recurrent Neural Networks (RNNs), vehicle positions, 5G.

ملخص

تستكشف هذه الأطروحة تطبيق نماذج الذاكرة قصيرة الأمد طويلة الأمد (LSTM) للتنبؤ بمسارات المركبات، بهدف تحسين أنظمة النقل الحضري وتعزيز نشر وحدات الحوسبة المتنقلة على الحافة (MEC) في بيئات المدينة الذكية. تستخدم البحث مجموعة بيانات شاملة تتضمن إحداثيات GPS لسيارات الأجرة في سان فرانسيسكو، تم معالجتها إلى بيانات متسلسلة زمنيًا للنمذجة التنبؤية. من خلال التجارب المدققة، تُظهر الأطروحة أداءً متفوقًا لنماذج LSTM على الشبكات العصبية التكرارية البسيطة (RNNs) التقليدية في التنبؤ بمواقع المركبات بدقة عالية. هذه التنبؤات ضرورية لتحديد المواقع المثلى لوحدات MEC، التي يتم استضافتها على أو بالقرب من هوائيات الجيل الخامس لتحسين معدلات نقل البيانات وتقليل تأخيرات الاتصال. من خلال تجميع المواقع المتوقعة، تقدم الدراسة إطارًا لوضع MEC بكفاءة، مما يسهم في تقليل ازدحام الشبكة وتعزيز إدارة البنية التحتية الحضرية. لا تسلط هذه العمل الضوء فقط على قدرات نماذج LSTM في تحليل البيانات الزمنية الحقيقية ولكنها تؤكد أيضًا على إمكانياتها في دعم البنية التحتية لشبكات الجيل اللاحق اللاسلكية في البيئات الحضرية.

الكلمات المفتاحية :الذاكرة الطويلة قصيرة المدى (LSTM)، حوسبة الحافة المتنقلة (MEC)، مجموعة البيانات، نظام تحديد المواقع العالمي (GPS)، الشبكات العصبية البسيطة المتكررة (RNNs)، مواقع المركبات، G.5

Table of Contents

List of Figure

List of Tables

List of Abbreviations

MANETs: Mobile Ad Hoc Networks

VANET: Vehicular Ad-hoc Networks

RSU: Roadside Unit

OBU: On-Board Unit

V2V: Vehicle-to-Vehicle Communication

DSRC: Dedicated Sho

rt-Range Communication

V2I: Vehicle-to-Infrastructure communication

I2I: Inter-Infrastructure communication

QoS: Quality of service

LBRP: Load Balancing Routing Protocol

5G: Fifth generation

OTP: Open Transport Protocol

IoT: Internet Of Things

AR: Augmented Reality (AR)

VR: Virtual Reality

MEC: Multi-access edge computing

AI: Artificial Intelligence

ML: Machine learning

DL: Deep learning

ANNs: Artificial Neural Network

RNNs: Recurrent Neural Networks

ReLU: Rectified Linear Unit

tanh: Hyperbolic tangent function

LSTM: Long Short-Term Memory

HTML: Hypertext Types and Markup Languages

CSS: Cascading Style Sheets

MSE: Mean Squared Error

MAE: Mean Absolute Error

FNN: Feedforward Neural Networks

CNN: Convolutional Neural Networks

WCSS: Within the Cluster Sum of Square

General Introduction

General Introduction

This thesis explores the transformative impact of integrating Vehicular Ad hoc Networks (VANETs) with emerging technologies such as 5G, Multi-access Edge Computing (MEC), and Artificial Intelligence (AI). These integrations are pivotal in advancing the capabilities of smart transportation systems. This dissertation is structured into several chapters, each dedicated to understanding and enhancing different aspects of vehicular communication technologies.

Chapter 1, provides an overview of VANETs, explaining their importance in improving road safety and transportation efficiency. It discusses the architecture, functionalities, and challenges of VANETs, setting the foundational knowledge required to appreciate the enhancements that 5G, MEC, and AI can offer. Chapter 2 shifts to 5G technology, detailing its development, capabilities, and expected impact on vehicular networks. The chapter examines how 5G's high data transmission speeds, reduced latency, and increased reliability are crucial for supporting the advanced needs of modern vehicular networks. In Chapter 3 The MEC is explored in depth, illustrating how its implementation at the network edge significantly reduces latency and allows real-time data processing near data sources. This chapter elucidates how MEC supports the heavy data demands of advanced vehicular applications, integrating seamlessly with 5G to enhance network performance. Chapter 4 delves into the principles and applications of AI, with a focus on machine learning and deep learning. It outlines how these technologies enable machines to perform tasks traditionally requiring human intelligence, such as speech recognition and decision-making. The text discusses various forms of machine learning, including supervised, unsupervised, and reinforcement learning, as well as the structure and training of neural networks. It also addresses the challenges associated with neural networks, such as the need for large datasets, the risk of overfitting, and issues with interpretability. The chapter highlights the significant potential of AI while acknowledging the complexities and limitations inherent in current technologies. Chapter 5 This chapter outlines the results of an experiment using LSTM models to predict vehicle trajectories from a San Francisco taxi dataset. It details the tools and programming languages used, such as Python, TensorFlow, and Keras, and describes the preprocessing, model training, and evaluation. The LSTM model demonstrated high accuracy in predicting vehicle positions, which were then effectively clustered using the K-means algorithm. The chapter highlights the model's superior performance over simple RNN models and its application in optimizing urban infrastructure and Mobile Edge Computing deployment.

chapter I:

I Generalities about Vehicular Ad hoc Networks (VANET)

1. Introduction

Vehicular ad hoc networks (VANETs) have emerged as an exciting and rapidly evolving field of research and application. These networks facilitate the real-time exchange of digital data among vehicles, promising significant advancements in road safety and transportation efficiency. The adoption of VANET technology is increasingly necessary due to the alarming frequency of accidents globally, which claim thousands of lives annually. Additionally, VANETs address the chronic congestion of transport infrastructures, which imposes substantial socio-economic costs through heightened air pollution, increased fuel consumption, and extensive time losses for commuters. This chapter explores VANETs in detail, offering insights into their components, functionalities, applications, and the challenges they address within the realm of modern transportation systems.

2. Wireless networks

Wireless networks provide connection flexibility between users in different places. Moreover, the network can be extended to any place or building without the need for a wired connection. Wireless networks are classified into two categories; Infrastructure networks and Ad Hoc networks [1]. As shown in Figure 1.

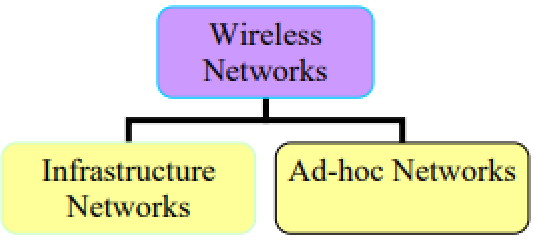


Figure 1. Wireless networks.

2.1. Infrastructure networks

The infrastructure here is the use of access points, each node is connected to this access point also called the base station which must be fixed with a wireless link to send and receive messages, so if two Nodes want to communicate they must go through the base station so that the client can access the Internet and exchange digital data [2].

2.2. Ad Hoc networks

An ad hoc or infrastructure-free network is a network whose nodes communicate without the use of centralized base stations, each machine that wants to communicate with another plays a dual role which is both a communicator and an access point [2].

2.2.1. Mobile Ad Hoc Networks (MANETs)

A mobile ad hoc network or MANET is a set of nodes that can move in any territory and communicate with each other by radio waves, as depicted in Figure 2. A link is created between two nodes only if they are located at a distance less than or equal to a certain well-defined transmission radius, this network is used in important applications such as natural disaster management systems surveillance systems, and others [3].

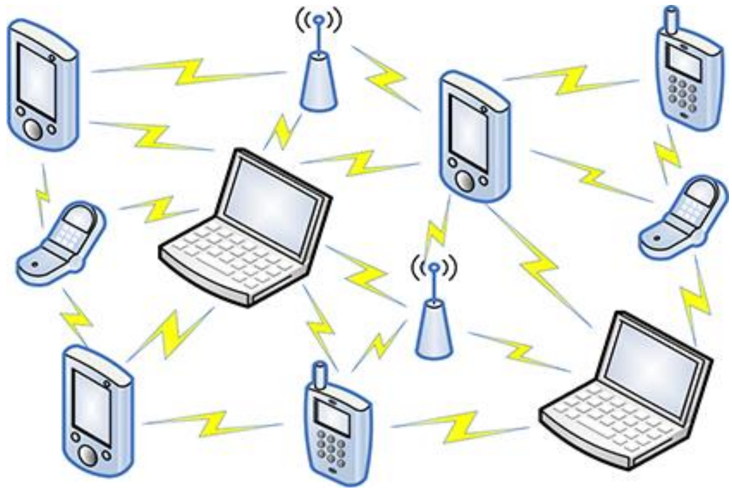


Figure 2. Mobile Ad Hoc Networks (MANETs)

2.2.2. Vehicular Ad Hoc Networks (VANETs)

Belonging to the family of mobile ad hoc networks, a vehicular ad hoc network is a communication network between intelligent vehicles equipped with different types of sensors that exchange information between them as shown in Figure 3, or with road infrastructures which are platforms placed at the side of the road, called roadside units (RSUs). The purpose of a VANET is to improve road safety, avoid traffic jams, reduce the time spent on the road, and provide digital services to road users. VANET is intended for safety applications, traffic management applications, and driver and passenger comfort applications [3].

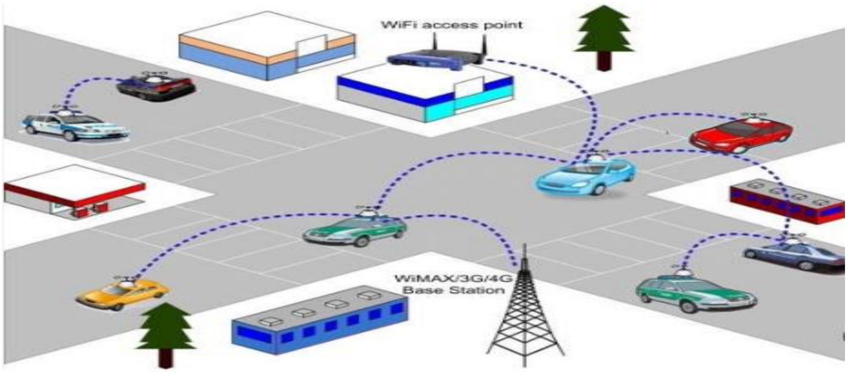


Figure 3. Vehicular Ad Hoc Networks (VANETs)

2.2.2.1. Components of VANETs

The essential components that can communicate in a VANET network are:

- **The Vehicle:** An intelligent vehicle is equipped with sensors and a unit called an On-Board Unit (OBU) [4]. This unit can store, process, locate, and send messages over a network interface.
- **The infrastructure:** By using the Roadside Unit (RSU) [4], which is a device usually installed at the side of the road or in dedicated locations such as at intersections or near parking spaces, this device can inform vehicles about the specific traffic conditions of the road by collecting and analyzing traffic data provided by intelligent vehicles.

2.2.2.2. VANET characteristics

VANET has particular properties, which influence vehicular applications. The most important among them could be cited:

- **High mobility:** Easily the speed node is located between 0 k/h when the vehicle is stuck in traffic jams and 200 k/h on highways so it changes according to the environment. Consequently, there is a high dynamic change in vehicular networking [5].
- **Vehicle resources:** vehicles are equipped with large storage capacity, computing power, and unlimited energy, unlike MANET, which is limited in terms of energy [5].
- **Mobility model:** Contrary to other networks which contain notes moving around arbitrarily, VANETs are characterized by their predicted mobility because of the infrastructure (roads, highways, signs, and limited speeds) and diver behaviors and their reactions against critical situations such as accidents

and traffic jams, etc [5].

- **Density:** The deployment of VANET nodes can cause a high density in the transmission range of one node that can exceed 250 nodes due to various reasons as well as traffic jams and accidents. However, a low density can occur if the area contains a low number of vehicles, particularly in rural areas or at night [5].
- **Radio channel characteristics:** The communication in VANET is done in the external environment; this last contains many obstacles such as urban areas, which cause decreasing in the quality of services [5].

2.2.2.3. VANET architectures

Vehicle architectures are divided into three modes as follows as shown in Figure 4:

1. **Vehicle-to-Vehicle Communication (V2V):**It is a communication and exchange of data concerning the state of the vehicle and internal information and the environment and the world surrounding the vehicle, this communication takes place between vehicles traveling in the same area using sensors and without going through intermediate infrastructures, if a vehicle detects an event on the road or it breaks down, it passes the information to other vehicles[6].
2. **Vehicle-to-Infrastructure Communication (V2I):**It is a communication between devices located along the road (RSUs) and the sensors of vehicles traveling within the range of these infrastructures. These means make it possible to transmit information and share data relating to traffic services, namely tolls, parking, public transport, and also data collection and vehicle/vehicle connections. It also helps to accurately determine the lateral position of the vehicle on the road to warn the driver of a risk[6].
3. **Hybrid Communication:**It is a combination of V2V communication and V2I communication, it is created due to the limitations of each of the two architectures, on one side the V2I mode and its short distance which is solved by the use of vehicles as relays, and on the other side the V2V mode and the routing problem during the routing of data across long distances which is solved by the access to an infrastructure that can improve network performance[6].

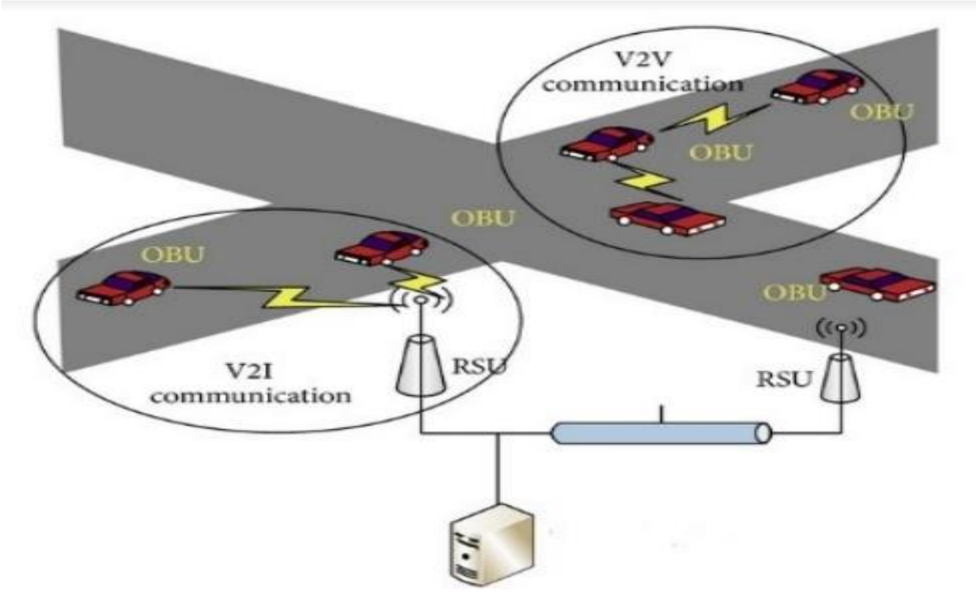


Figure 4. VANET architectures.

2.2.2.4. Communication standards in VANET

Vehicular communication is based on standards that define certain technical rules to allow telecommunications operators and the automotive industry to produce certain portable devices with common transmission characteristics, the main standards are as follows:

- **Dedicated Short-Range Communications (DSRC):**Dedicated Short Range Communications uses a non-voice telecommunications technique to transmit data over short distances between roadside systems (devices located along the road) and mobile units (vehicles), belongs to the IEEE 802.11p protocol family, and the 5.9 GHz band is dedicated to this use.

  DSRC systems are used for various applications such as traffic light control, traffic monitoring, providing warnings to motorists, automatic toll collection, traffic jam detection, and electronic inspection of moving trucks by transmitting data to roadside inspection stations[7].

- **Wireless Access in Vehicular Environment (WAVE):**A new generation of DSRC systems includes wireless Internet applications for drivers, the best known is WAVE (Wireless Access in Vehicular Environment) which allows vehicle-to-vehicle and vehicle-to-infrastructure communication for data exchange, for this WAVE has defined two types of devices: mobile device (OBU) and a fixed device (RSU), this standard has been defined as a set of rules to ensure health or environmental protection or to ensure against interference with or from any other device or networks, improve safety in transport, intelligent management, and side the V2V mode and the routing problem during the routing of data across long distances which is solved by the access to an infrastructure that can improve network performance[7].

2.2.2.5. Challenges of VANET

Ad hoc vehicular networks address several challenges[8]:

- **Quality of service (QoS):** the demand for quality service depends on the supported applications. The main constraint in the security applications is latency. The validity of the information is limited in time; messages need to reach their destination in a short time to be considered relevant.
- **Reliable radio channel:** the role of radio channel management mechanisms is to offer reliable and robust transmission and a fair sharing of the communication medium. To achieve this objective in the case of vehicular networks, it is necessary to define methods to deal with two major problems of transmissions: inter-symbol interference due to the wave propagation and multipath effect Doppler caused by the movement of vehicles.
- **Routing:** Routing protocols are used in ad hoc communications. They can determine the result of nodes that packets must traverse to an exchange of information between remote entities. The problems to be met by routing protocols are the intermittent connectivity which makes them obsolete the already established routes and network partitioning that prevents the spread of packets.
- **Geographical address and geocast:** routing geocast is a mechanism similar to multicasting in which the recipients are identified by geographical constraints. It is used by applications displaying data that are useful only for vehicles which are in a specific geographic area.

2.2.2.6. VANET applications

Different applications are found in VANETs, and are cited in the following:

- **Road safety applications:** it has a significant impact on saving people's lives by sharing and gathering information by safety messages.
- **Vehicles authority services:** this kind of application helps the transport authorities such as police and emergency to contribute to road safety by

sending warning and emergency messages from authority vehicles to ordinary vehicles which also can send the information to the authority centers using surveillance applications like stolen vehicles tracking and electronic license plate verification.

- **Enhanced driving:** the information is disseminated in the local area to propose some beneficial actions in the case of awful weather and supply informative services as well as the nearest fuel station, etc.
- **Business and entertainment services:** they provide drivers and passengers with a set of services through the Internet or other private networks such as parking payment and road usage payment. Moreover, entertainment and interactive multimedia by downloading movies, and music and playing online games, etc.
- **Traffic Management Applications:**The messages exchanged in traffic management applications aim to improve road traffic by optimizing it through the selection of the appropriate paths and roads, taking into account potential traffic jams or obstacles to pass by. This spreads the traffic, reduces the travel driver's time, and saves on fuel consumption.

2.2.2.7. VANET Routing Protocols

Due to the high mobility feature of VANET, the use of the correct routing protocol is of great concern. The packets in the network are sent from vehicle to vehicle that are moving with speed and also the density of vehicles is increasing and decreasing which increases the challenges related to routing protocols. Due to the highly challenging nature of VANET, the researcher came out with different types of protocols which will be explained in the following sections. Moreover, Figure 5 shows the classification of routing protocols.

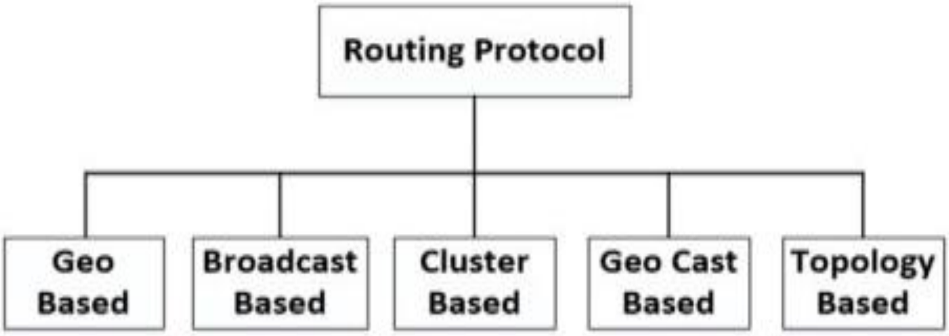


Figure 5. Classification of Routing Protocols

1. **Geo Based Protocols**

   In these protocols, a source will communicate with the destination by using geographical positions as well as with its network address. Load Balancing Routing Protocol (LBRP) calculates as well as configures the route as per information based on the location of nodes. Therefore, there is no such need to build the routing tables. The protocol consists of three components such as beaconing, location, and forwarding services. The drawback of this protocol is that it requires the Global Positioning System (GPS) assistance to get the location of the vehicle. Moreover, the satellite signals get weak when the vehicle goes into an area like a tunnel. But as far as the highway environment is concerned it gives the best performance. Additionally, its advantages include efficiency in high-mobility environments. Examples of these protocols are Greedy Perimeter Stateless Routing and Distance Routing Effect Algorithm for Mobility (DREAM) [9].

2. **Broadcast Based Routing Protocols**

   This protocol floods the data packet from the entire VANET to all the available nodes in the broadcast domain. The said protocol is used whenever the destination node is out of the range of the source node. Mostly these protocols are used with applications that are concerned with safety such as road and weather condition warnings, emergency warning messages, etc. Example of broadcasting routing protocols includes Distributed Vehicular Broadcast Protocol (DV-CAST), Position Aware Reliable Broadcasting Protocol (POCA), and Density Aware Reliable Broadcasting Protocol (DECA). The positive point of the said protocols is their reliability. But these types of protocols consume more bandwidth and many duplicate packets reach the node which is not good for the protocol performance [10].

3. **Cluster Based Routing Protocols**

In this protocol, the vehicles with the same characteristics such as speed, direction, etc. are combined in one cluster. Moreover, if a vehicle node needs to communicate with the node within the cluster then the data will follow a direct path as it is considered to be a local communication. Moreover, if the vehicle node needs to communicate with the node that is outside the cluster than it requires the help of its cluster head for reaching the destination. The scalability factors make it a good choice for network designers. But traffic delays are its drawbacks. Clustering for Open Inter Vehicular Communication Network (COIN) is the prime example of this protocol [11].

4. **Geo-cast Routing Protocols**

Geo cast protocol consists of two main zones i.e. Zone Of Relevance (ZOR) and Zone Of Forwarding (ZOF). ZOR is that area dedicated to the nodes of that region. The main goal of this protocol is to make communication possible between the vehicles present in ZOR. Moreover, if the source vehicle wants to communicate with the vehicle that is not in ZOR of that vehicle then the vehicle will become part of ZOF and any vehicle that comes in ZOF has the responsibility of forwarding the information to other ZORs. Due to the frequent changes in the zones, the connection disconnection can take place regularly and the sand point comes in the bracket of drawbacks [12].

5. **Topology Based Routing Protocols**

The focus of this paper is on topology-based protocols. The said type consists of two kinds of protocol such as proactive and reactive. Moreover, the classification of routing protocols used in an ad-hoc environment [13].

2.2.2.8. Types of messages in VANET

Messages in VANET are divided into two types[14]:

1. **Service Messages:** These means the messages used when requesting or responding to a service we find two types:
   - **Safety Messages:** It works to help road users by giving alerts, for

example, in the event of congestion or an accident (beacons).

- **Non-safety Messages:** These types of messages are less priority than the first type, but they have an important role in making driving more comfortable as they provide helpful information.

2. **Control Messages:** This type of message is used to control the network for example Network setup, Authentication, and Network policy update.

3. Conclusion

In this chapter, we have explored ad hoc networks, with a particular focus on Vehicular Ad hoc Networks (VANETs), which are pivotal in improving road safety through the utilization of communication devices such as roadside units and on-board units. We have examined their applications, including the various communication modes that facilitate information exchange among vehicles and between vehicles and infrastructure. Additionally, we have highlighted key communication standards like DSRC and WAVE for their diverse applications within VANETs.

Looking ahead, while VANETs encounter challenges related to latency, reliability, and scalability, and DSRC faces limitations in range and bandwidth, the imminent advent of 5G technology offers significant promise. The advancements in high-speed data transmission, low latency, enhanced reliability, and extensive coverage brought by 5G are poised to markedly improve the safety and efficiency of vehicular communication networks.

In the next chapter, we will delve deeper into understanding 5G technology and its seamless integration with smart transportation net

chapter II:

II Fifth generation (5G) networks

1. Introduction

In the ever-evolving landscape of wireless communication, the need for faster, more reliable, and efficient connectivity has become paramount. The advent of 5G technology has revolutionized the telecommunications industry, promising significant advancements in data transmission speeds, capacity, and network performance. This breakthrough in connectivity is set to be a game-changer, enabling a host of new applications and services that were previously impossible, such as remote surgery, self-driving cars, and smart cities. Furthermore, 5G will enhance the quality of internet connectivity, making it more reliable and accessible to a broader range of people. As we delve into this chapter, we wsill explore the history, architecture, evolution, and applications of 5G networks, as well as their integration with Vehicular Ad Hoc Networks (VANETs), highlighting the transformative impact of this cutting-edge technology.

2. Brief History of Mobile Telephone Networks

The mobile telephone network has historically and mainly been divided into five generations. Each generation has specific characteristics that set it apart from others. Each generation is different from the other in terms of frequency, data rate, maximum number of users...etc[15].

Table 1 summarizes the specific characteristics of the different generations of mobile communication systems.

Table 1. Generations of mobile communication systems.

Generation	1G	2G	3G	4G	5G
1st year of deployment	1981	1992	2001	2010	2020
Datarate	2 Kb/s	64 Kb/s	2 Mb/s	100 Mb/s	10 Gb/s

Frequencies	900 MHz	900 MHz and 1,8 GHz	800/900MHz 1.7 to 1.9GHz 2100 MHz	800MHz 900MHz 1800MHz 2100MHz 2600MHz	28GHz 37 GHz 39 GHz 64 –71 GHz
General Functional description	Analog cell phones	Digital cellular phones (GSM/CDMA)	First mobile bandwidth using IP protocols (WCDMA 2000)	Mobile broadband on aunified standard (LTE)	Internet Touch – Improve M2M communication network

3. The 5th Generation

5G is the fifth generation of mobile networks, succeeding 1G, 2G, 3G, and 4G. This new global wireless standard is designed to connect virtually everyone and everything, including machines, objects, and devices as it's presented in Figure 6. With significantly higher upload and download speeds, more consistent connections, and lower latency, 5G is much faster and more reliable than 4G[15]. This advancement has the potential to transform how we use the internet, enhancing our access to applications, social networks, and information. 5G's capabilities enable a wide range of innovative technologies and services, paving the way for advancements in various fields.

Figure 6. 1G to 5G Journey

4. 5G Mobile Network Architecture

5G concepts correspond to OSI (Open Systems Interconnection)layers, though 5G utilizes four basic layers [16]. Table 2 presents a comparison between the seven layers of the OSI model and the corresponding four layers in the 5G architecture.

Table 2. Comparison between OSI and 5G layers

Application layer	Application (services)
Presentation layer	
Session layer	Open Transport Protocol (OTP)
Transport layer	
Network layer	Upper network layer
	Lower network layer
Data link layer (MAC)	Open Wireless Architecture (OWA)
Physical layer	

4.1. Physical layers / MAC

The physical access control and media access layers, corresponding to OSI layers 1 and 2, define the wireless technology used in 5G, as shown in Table 2. For these layers,5G mobile networks are likely to be based on Open Wireless Architecture [16].

4.2. Network layer

Layer 2 of 5G is subdivided into upper and lower layers, as shown in Table 2. The network layer of 5G technology corresponds to OSI layer 3. This layer will use the Internet Protocol (IP), with IPv4 being widely used globally. However, IPv4 has limitations, such as limited address space and lack of support for Quality of Service (QoS). These issues are addressed in IPv6, which will be used in 5G networks [16].

4.3. Open Transport Protocol (OTP) layer

The open transport protocol layer is the third layer of 5G technology, corresponding to the transport and session layers of the OSI model. Mobile and wireless networks differ from wired networks at the transport layer. In all versions of the Transmission Control Protocol (TCP), packet loss is assumed to be due to network congestion. However, in wireless technology, packet loss may also occur due to a higher bit error rate in the air interface. Therefore, TCP modifications are planned for mobile and wireless networks to retransmit damaged TCP segments only over the wireless link.

For 5G mobile terminals, a downloadable and installable transport layer will be required. These mobile devices will have the capability to download a version targeted to the specific wireless technology installed in the base stations (BS). This is known as the open transport protocol (OTP) [16].

4.4. Application layer

The application layer is the final layer in both the 5G and OSI models. In terms of applications, the ultimate goal for 5G mobile terminals is to provide intelligent Quality of Service (QoS) management across various networks.

Currently, mobile phone users manually select the wireless interface for specific Internet services without the ability to use QoS history to choose the best wireless connection for a given service. The 5G phone will offer the capability to test QoS and store the measurement information in the mobile terminal's information databases [16].

5. Evolution of cellular networks from 4G to 5G

Compared with 4G, the 5G wireless network has many functions and advantages, providing users with a faster and more reliable communication experience:

- **Higher data transfer speeds:** 5G networks greatly improve data transfer speeds, enabling faster download and upload speeds. This enables users to access high-quality media content, download files, and use a variety of applications more quickly [17].

- **Low-latency communication:** 5G significantly reduces communication latency, potentially to 1 millisecond or less, enabling near real-time communications. Applications that require real-time communication, such as video conferencing, online gaming, and self-driving cars, all rely on this[17].
- **Increased network capacity:** 5G can support many more devices simultaneously, potentially up to 1 million per square kilometer. This offers enhanced assistance in situations like smart cities, IoT devices, and large-scale machine communications[17].
- **Stronger Security:** 5G operates network slicing technology, which allows operators to provide customized virtual networks on the same physical infrastructure to meet specific business needs. This approach ensures better security and privacy, improves infrastructure efficiency and flexibility, speeds service deployment, and reduces dependence on hardware, resulting in cost savings[17].

6. 5G Application

The following section outlines several principal applications of 5G technology:

- **Internet Of Things (IoT):** 5G facilitates the connectivity of a vast number of IoT devices, enabling applications like smart homes, industrial automation, and smart cities.
- **Agriculture:** Developed through the use of sensors and motors, for example, to measure and link soil quality, rain, temperature, and wind to monitor crop growth.
- **Health:** includes several minor improvements to the complex, such as exercise monitoring, consumer health sensors, wireless connectivity in hospitals, patient monitoring, telehealth, remote surgery, etc.
- **Vehicles:** Many applications concern intelligent wireless communication, for example, to regulate roads, ensure vehicle-to-vehicle communication and avoid accidents.
- **Augmented Reality (AR) and Virtual Reality (VR):** 5G enables immersive

AR and VR experiences with its high-speed, low-latency connections, unlocking new possibilities in gaming, education, healthcare, and entertainment.

- **Connectivity for Edge Computing**: With the move to cloud-native 5G networks, enterprises can take advantage of strategically distributed computational power, allowing more data to be processed and stored in the right place based on the needs of the application. Intelligent edge computing operates at the convergence of 5G's ultra-low latency, IoT, and AI technologies. Devices and applications can tap into edge cloud computing resources without needing to access a centralized data center potentially thousands of miles away.

5G is likely to have many other incredible applications and improvements across the board as presented in Figure 7.

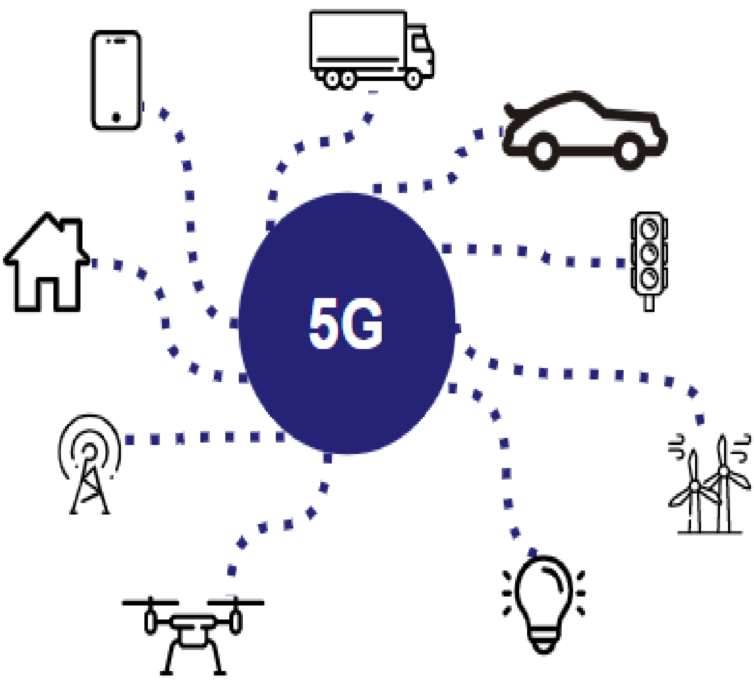


Figure 7. 5G Application

7. Integration of VANET and 5G

The integration of Vehicular Ad-hoc Networks (VANETs) with 5G technology holds great potential to enhance road safety, traffic efficiency, and overall transportation systems [18]. Here's how the integration can benefit:

- **Improved Communication Infrastructure:** 5G networks provide high-speed, low-latency connectivity, allowing vehicles to communicate with each other and with the infrastructure more effectively. This enables real-time sharing of critical information such as traffic conditions, road hazards, and emergencies [18].
- **Enhanced Safety:** VANETs combined with 5G can enable advanced safety applications such as cooperative collision avoidance and platooning. Vehicles can exchange data about their speed, direction, and proximity, allowing them to anticipate and avoid accidents more effectively [18].
- **Traffic Management:** By integrating VANETs with 5G, traffic management systems can gather real-time data from vehicles and roadside infrastructure to optimize traffic flow, reduce congestion, and improve overall transportation efficiency. This includes dynamic route planning, traffic signal optimization, and adaptive traffic control [18].
- **Vehicle-to-Infrastructure (V2I) Communication:** 5G enables seamless communication between vehicles and roadside infrastructure such as traffic lights, road signs, and sensors. This allows for more efficient traffic management, optimized traffic signal timings, and enhanced situational awareness for drivers [18].
- **Fleet Management and Logistics:** The integration of VANETs with 5G can benefit fleet management and logistics operations by providing real-time tracking of vehicles, optimizing routes based on traffic conditions, and improving delivery efficiency [18].
- **Smart City Integration:** VANETs integrated with 5G can be part of broader smart city initiatives, contributing to safer and more efficient urban mobility. This includes applications such as smart parking, public transportation management, and environmental monitoring [18].

8. Conclusion

Deploying 5G antennas optimally is crucial for enhancing network performance by minimizing latency and maximizing data throughput, ensuring robust coverage and capacity even in densely populated areas. This strategic placement not only reduces energy consumption and operational costs but also supports advanced applications such as IoT, AR, and VR, significantly improving user experiences with faster, more reliable connections. Integration of 5G with Vehicular Ad Hoc Networks (VANETs) marks a transformative step in transportation technology. This integration enhances vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, bolstering road safety, traffic management, and overall transportation system efficiency.

chapter III:

III Multi-access edge computing (MEC)

1. Introduction

Multi-access Edge Computing (MEC) is a key solution that enables operators to open their networks to new services and IT ecosystems to leverage edge-cloud benefits in their networks and systems. Located close to the end users and connected devices, MEC provides extremely low latency and high bandwidth while always enabling applications to leverage cloud capabilities as necessary. This chapter explores the evolution, characteristics, applications, and integration of MEC with 5G technologies, highlighting its pivotal role in shaping the future of edge computing and telecommunications infrastructure.

2. Cloud Computing

Cloud computing refers to the use of hosted services, such as data storage, servers, databases, networking, and software over the Internet. The data is stored on physical servers, which are maintained by a cloud service provider. Computer system resources, especially data storage and computing power, are available on-demand, without direct management by the user in cloud computing [19].

3. Multi-access edge computing (MEC)

Multi-Access Edge Computing (MEC) moves the computing of traffic and services from a centralized cloud to the edge of the network and closer to the customer. Instead of sending all data to a cloud for processing, the network edge analyzes, processes, and stores the data as illustrated in Figure 8. Collecting and processing data closer to the customer reduces latency and brings real-time performance to high-bandwidth applications [20].

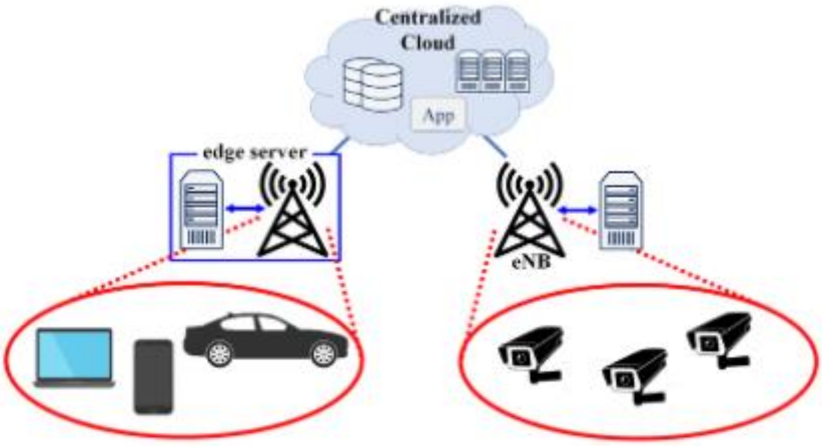


Figure 8. MEC architecture.

4. Multi-access edge computing vs. mobile edge computing

We can use the term Multi-access Edge Computing (MEC) when Mobile Edge Computing (MEC) is combined with access technologies and media other than cellular networks, such as Wi-Fi or wired networks. Therefore, when there is an expansion in the range of supported technologies for access, we can use the term Multi-access Edge Computing to describe this comprehensive platform. While multi-access edge computing refers to the edge of any network, mobile edge computing refers to the edge of only the mobile network [21].

5. The Rise of Edge Computing

With the rise of the Internet of Things we now have various devices actively communicating and also sharing, delivering information. Edge computing brings computation and data storage closer to the devices where it’s being gathered, rather than relying on centralized cloud computing that can be thousands of miles away. This is done so that data, especially real-time data, does not suffer latency issues that can affect an application’s performance [20].

6. Edge vs. cloud

1. **Edge:** Edge computing is the deployment of computing and storage resources at the location where data is produced. This ideally puts compute and storage at the same point as the data source at the network edge. For example, a railway station might place a modest amount of computing and storage within

the station to collect and process myriad track and rail traffic sensor data. The results of any such processing can then be sent back to another data center for human review, and archiving and to be merged with other data results for broader analytics [22].

2. **Cloud:** Cloud computing is a huge, highly scalable deployment of compute and storage resources at one of several distributed global locations. Cloud providers also incorporate an assortment of pre-packaged services for IoT operations, making the cloud a preferred centralized platform for IoT deployments. But even though cloud computing offers far more than enough resources and services to tackle complex analytics, the closest regional cloud facility can still be hundreds of miles from the point where data is collected. In practice, cloud computing is an alternative to traditional data centers. The cloud can get centralized computing much closer to a data source, but not at the network edge [22].

7. How use of MEC

MEC shortens the distance between where data is produced, collected, and analyzed in the cloud. Processing that’s typically offloaded to the data center is now done virtually. Mobile edge clouds collect store, and process information close to wireless devices within a cloud network. Proximity to devices, and by extension users, helps drive significant performance enhancements, including higher bandwidth, lower latency, and faster response times and decision-making [20].

8. MEC Characteristics

Multi-access edge computing has five key characteristics, these are:

1. **Proximity**: MEC deployments are often close to the source of information or data to be processed, reducing the need for back-and-forth transfer of data to core.
2. **Real-time operations**: use cases that require real-time data processing and decision-making benefit greatly from multi-access edge computing, thanks to accelerated connectivity [20].
3. **Ultra-low latency**: with a latency of under 20 milliseconds, MEC guarantees

faster response and enhanced user experience [20].

4. **Continuous operations**: applications using MEC architecture are localized; hence they run independently even when disconnected from the core network [20].
5. **Interoperability**: multi-access edge computing allows apps and systems to communicate easily without the need to migrate or adapt them to a new environment [20].

9. The importance of MEC

A. **Security and surveillance:** Digital cameras in near constant use generate data. Having MEC as part of a camera network allows facilities to store and process data from many sources, faster. High bandwidth supports high image quality, and reliable computing power can analyze data quickly on-site (instead of at a centralized data center). For example, traffic cameras can adapt to traffic conditions remotely rather than redirect to a control center [23].

B. **Bandwidth:** Bandwidth is the amount of data that a network can carry over time, usually expressed in bits per second. All networks have a limited bandwidth, and the limits are more severe for wireless communication. This means that there is a finite limit to the amount of data - or the number of devices - that can communicate data across the network. Although it's possible to increase network bandwidth to accommodate more devices and data, the cost can be significant, there are still (higher) finite limits and it doesn't solve other problems [23].

C. **Latency:** Latency is the time needed to send data between two points on a network. Although communication ideally takes place at the speed of light, large physical distances coupled with network congestion or outages can delay data movement across the network. This delays any analytics and decision-making processes and reduces the ability of a system to respond in real time. It even costs live in the autonomous vehicle example [23].

10. Applications of Edge Computing

In principle, edge computing techniques are used to collect, filter, process, and analyze data in place at or near the network edge as presented in Figure 9.

1. **IoT (Internet of Things):** Edge computing is essential for processing the vast amounts of data generated by IoT devices in real time, enabling quick decision-making and actionable insights [24].
2. **Autonomous vehicles:** Autonomous platooning of truck convoys will likely be one of the first use cases for autonomous vehicles. Here, a group of trucks travel close behind one another in a convoy, saving fuel costs and decreasing congestion. With edge computing, it will be possible to remove the need for drivers in all trucks except the front one, because the trucks will be able to communicate with each other with ultra-low latency [24].
3. **Network optimization:** Edge computing can help optimize network performance by measuring performance for users across the internet and then employing analytics to determine the most reliable, low-latency network path for each user's traffic. In effect, edge computing is used to "steer" traffic across the network for optimal time-sensitive traffic performance [24].
4. **Improved Healthcare:** The healthcare industry has dramatically expanded the amount of patient data collected from devices, sensors, and other medical equipment. That enormous data volume requires edge computing to apply automation and machine learning to access the data, ignore "normal" data and identify problem data so that clinicians can take immediate action to help patients avoid health incidents in real time [24].
5. **Workplace safety:** Edge computing can combine and analyze data from on-site cameras, employee safety devices, and various other sensors to help businesses oversee workplace conditions or ensure that employees follow established safety protocols [24].
6. **The ongoing deployment of 5G networks:** As the next generation of a global wireless standard, 5G and the innovations that stem from it will rely on MEC to connect machines, objects, devices, and people everywhere. The applications for 5G are as varied as MEC itself, capable of not only consumer-facing technology like video games and entertainment, but also mission-

critical operations in education, agriculture, transportation logistics, and healthcare areas where split-second action is the difference between success and failure, or even life and death [24].

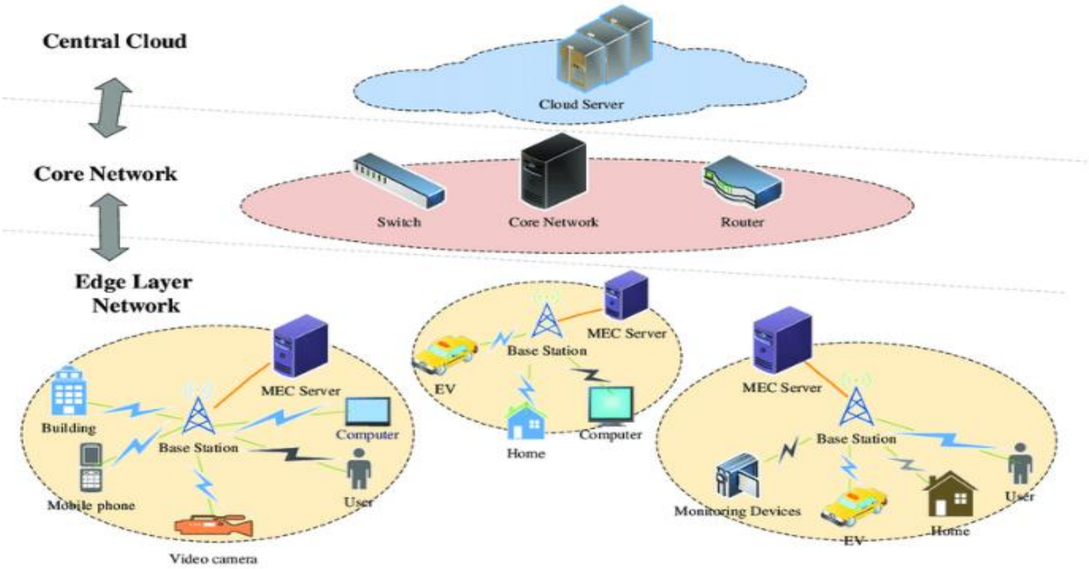


Figure 9. Applications of MEC

11. Integration of MEC with 5G technologies

5G and edge computing are symbiotic technologies: they are both poised to significantly improve the performance of applications and enable huge amounts of data to be processed in real-time as shown in Figure 10. 5G increases speeds by up to ten times that of 4G, edge computing reduces latency by bringing compute capabilities into the network, closer to the end user [25].

11.1. 5G needs MEC for two reasons

1. It is inherent to 5G standards as it is the only way to meet the latency targets that have been set (1ms network latency). While telecom operators have reported that 5G in the lab can deliver network speeds that are more than twenty times faster than LTE1, this will not reflect the experience of the average user. We feel it is likely that 5G will rely upon edge computing to reach the targets that are being set [25].
2. The gradual approach operators are taking to deploy 5G (the 5G go slow cycle) will mean coverage of “full 5G” will be insufficient to cultivate an ecosystem of new applications. However, the edge could seed a 5G market

even before widespread coverage [25].

11.2. The Benefits of 5G in Edge Computing

5G augments the existing capabilities of edge computing. Additional network speed accelerates inter-device communication, creating a collaborative environment that can effectively leverage automation while incorporating real-time workflow optimization. Collectively, this dramatically enhances performance. Plus, it makes incorporating high-demand technologies including AI and machine learning more viable, supporting better application response times and accelerating data collection and processing [25].

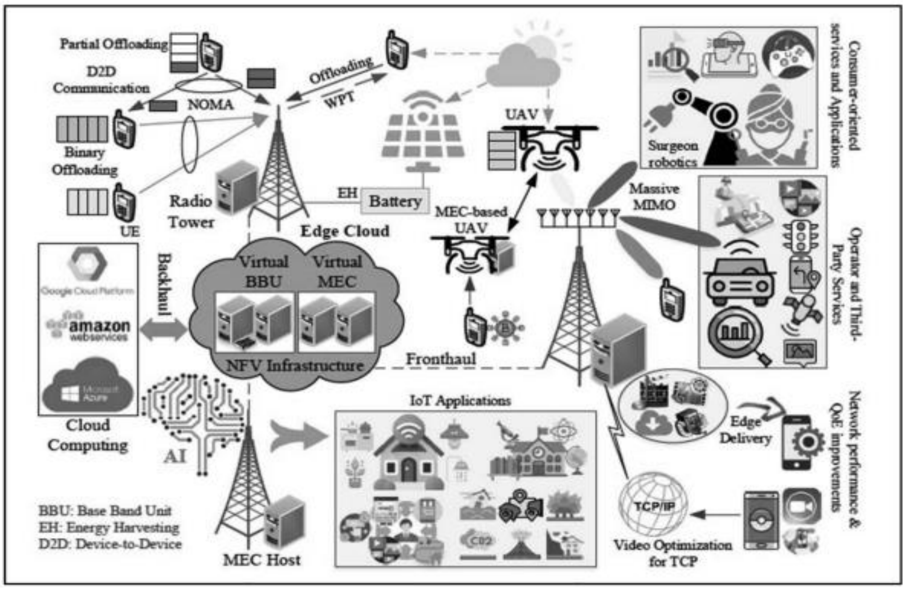


Figure 10. Integration of MEC with 5G technologies

12. How 5G and edge computing work together

5G and edge computing are key complementary technologies for delivering data-intensive consumer and enterprise applications like real-time inferencing for AI, cloud gaming, autonomous drones, or remote telesurgery. This is because these applications require a shorter, faster pipe to transfer data from the end-user to where data is processed to reduce latency and maintain a good user experience. 5G increases the speed the data travels, and edge computing reduces the distance it travels before it is processed. In short, the edge enhances the performance of 5G [25].

13. 5G and edge computing combine to support AI and GenAI

Much of the value of AI and GenAI applications is reliant on them being able to 'think' in real-time. For many applications, 5G and edge computing are the best combination of technologies to enable the lowest latency to deliver real-time inferencing. Rather than hosting an AI model in the hyperscale cloud, models can be trained in the hyperscale cloud and then run at the edge, with 5G delivering fast data rates between the edge node and the end user [26].

14. Conclusion

In summary, the combination of mobile edge computing (MEC) and 5G networks has the potential to completely transform the computer and telecommunications industries. MEC lowers latency and improves data processing efficiency by putting computational resources closer to end users, while 5G provides previously unheard-of speed and bandwidth. When combined, they allow for cutting-edge applications in a variety of fields and propel technological advancement toward a future where technology is more responsive, integrated, and effective.

chapter IV:

IV Artificial Intelligence (AI)

1. Introduction

Artificial Intelligence (AI) is the branch of computer science, which makes computers mimic human behavior to assist humans in better performance in the field of science and technology. Replicating human intelligence, solving knowledge-intensive tasks, building machines that can perform tasks that require human intelligence, and creating a system that can learn by itself are the few specific goals of AI. Machine learning and deep learning are two subsets of AI that are used to solve problems using high-performance algorithms and multilayer neural networks, respectively.

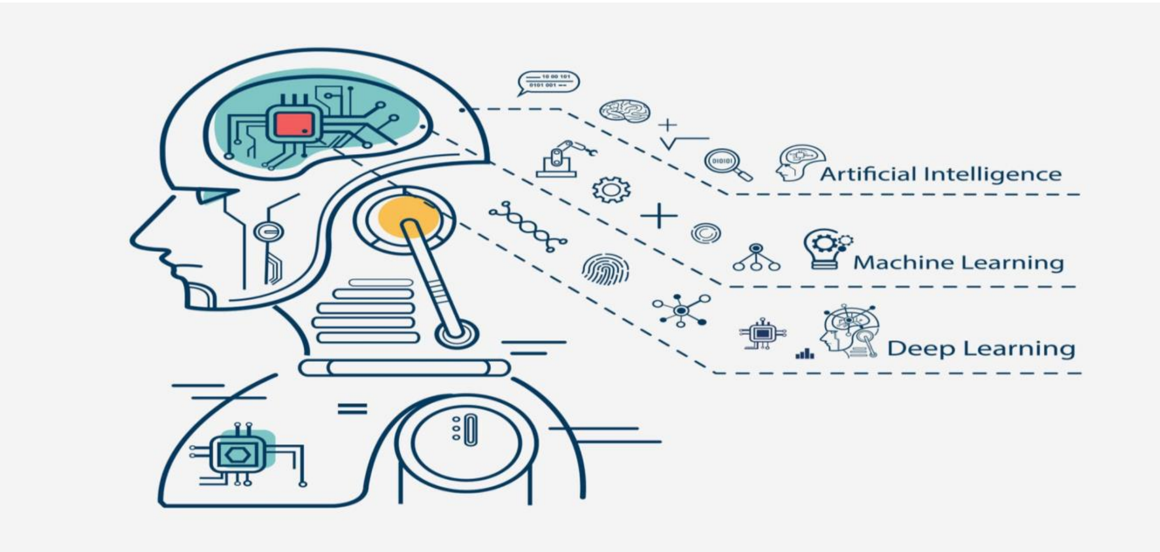


Figure 11. IA, ML, DL.

2. Artificial intelligence (AI)

Is the theory and development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns [27]. AI is an umbrella term that encompasses a wide variety of technologies, including machine learning, and deep learning, As shown in Figure 12.

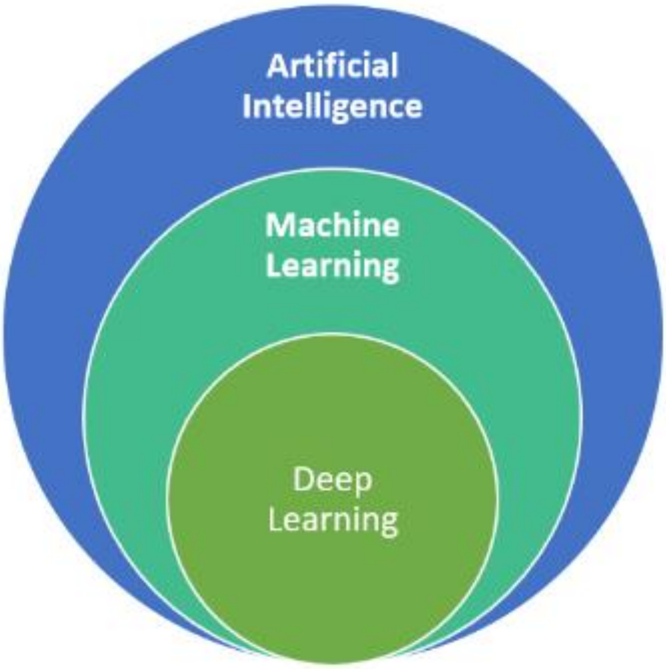


Figure 12. Artificial intelligence

3. Machine Learning

Today, intelligent systems that offer artificial intelligence capabilities often rely on machine learning to solve complex problems. These systems can learn from experience and make decisions that are based on what they have learned. They can recognize patterns and make predictions based on those patterns. In other words, machine learning is a type of artificial intelligence that allows computers to learn from data and make predictions based on patterns. Machine learning is used in everything from self-driving cars to online shopping recommendations, Google, Facebook, and Amazon are all using machine learning to make their services better. For example, when you search for something on Google, the results you see are tailored to what you've searched for before. That's because Google is using machine learning to customize your results based on the information in your search history. We can define Machine Learning with this sentence: To predict the future, we rely on the past [28].

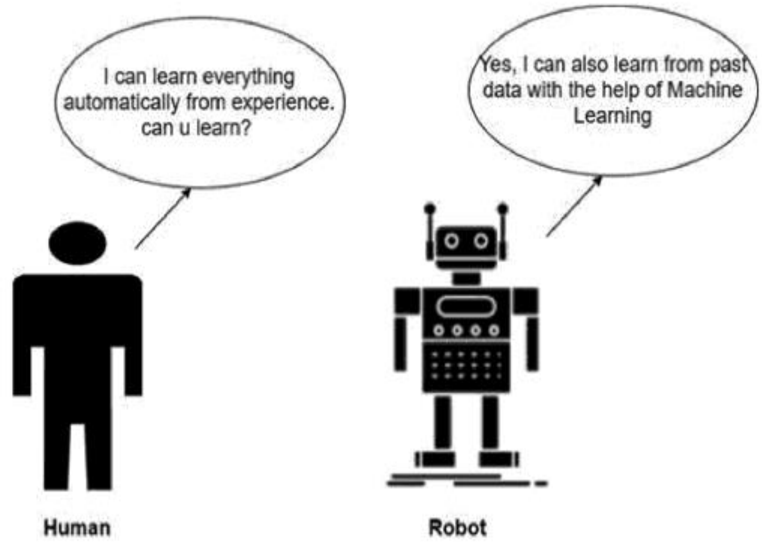


Figure 13. Human and robot

Figure 13, shows that the human learns everything automatically from experience & the robot can also learn from previous experience data with the help of machine learning.

3.1. Type of Machine Learning

In this part, we will discuss the types of machine learning in detail, from **Figure 14** these types :

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

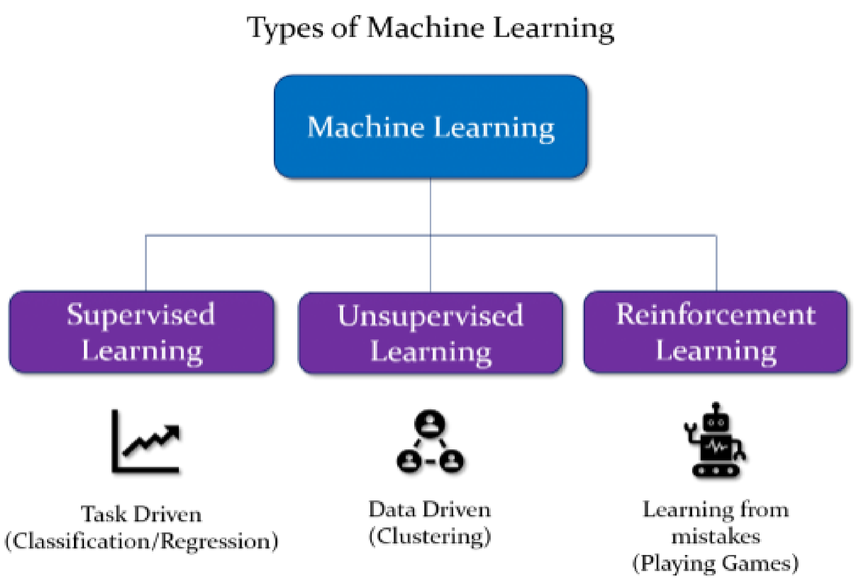


Figure 14. Type of Machine learning

3.1.1. Supervised Learning

Supervised machine learning algorithms are designed to learn by example. It is called supervised learning because the process of algorithm learning from the training dataset can be thought of as a teacher supervising the learning process.

In supervised learning, the data consists of an input variable and an output variable, or the dataset is labeled. Labeled data means the input data and its associated output are available in the dataset. During the training process, the algorithm searches for the patterns in the input data and correlates with the desired output data. After the training process, a supervised learning algorithm will take the unseen data as inputs and will determine which label the new inputs will be classified as based on prior training data. The objective of a supervised learning model is to predict the correct label for the newly presented input data [29].

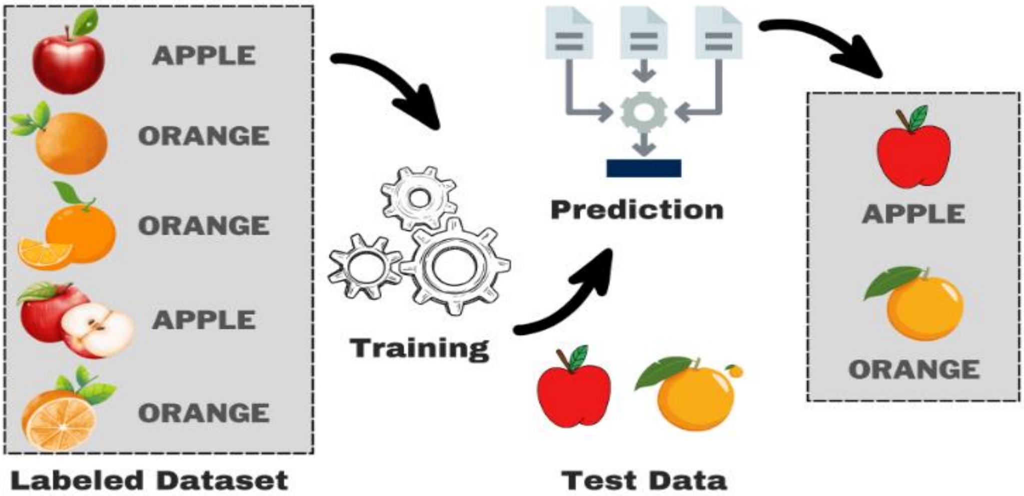


Figure 15.Supervised Learning

In “Figure 15”, we have images that have been labeled as either an orange or an apple.

This labeled data is fed into the machine, which then analyzes and learns the associations between these images based on their features such as shape, size, sharpness, etc. When a new image is presented to the machine without any label, the machine can accurately predict that it is an orange or an apple based on the information it has learned from the labeled data.

3.1.1.1. Types of Supervised machine learning

Supervised learning can be further divided into two types:

1. Classification
2. Regression

In Figure 16 we understand the difference between them

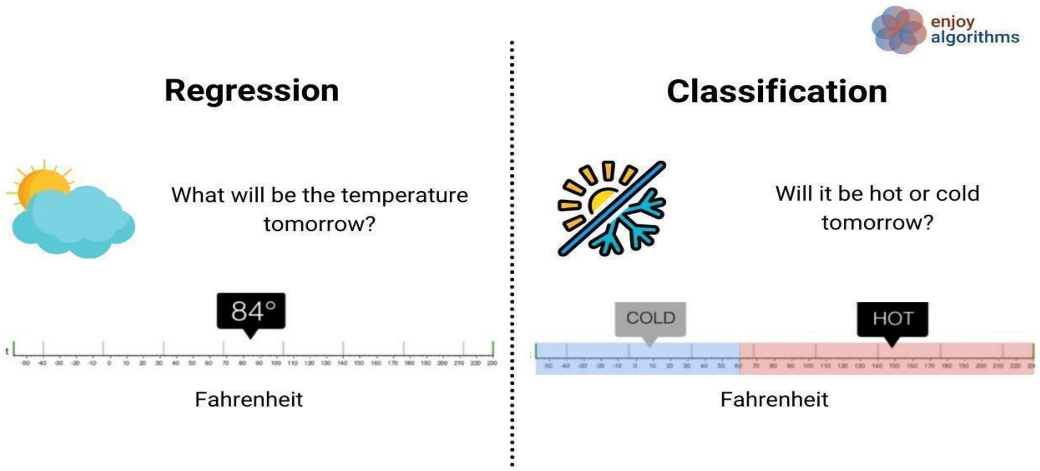


Figure 16. Types of Supervised Learning

3.1.1.1.1. Classification

Classification is the process of grouping the output into different classes based on one or more input variables. Classification is used when the value of the output variable is discrete or categorical, such as email is "spam" or "not spam" or "disease" and "no disease" or "rain" and "not rain" or "Yes" or "No" and 0 or 1, and so on [30].

If the algorithm tries to classify input variables into two different classes As illustrated in Figure 17, it is called binary classification, such as if the email is "spam" or "not spam". When the algorithm tries to classify input variables into more than two classes, it is called multiclass classification, such as handwritten character recognition where classes go from 0 to 9.

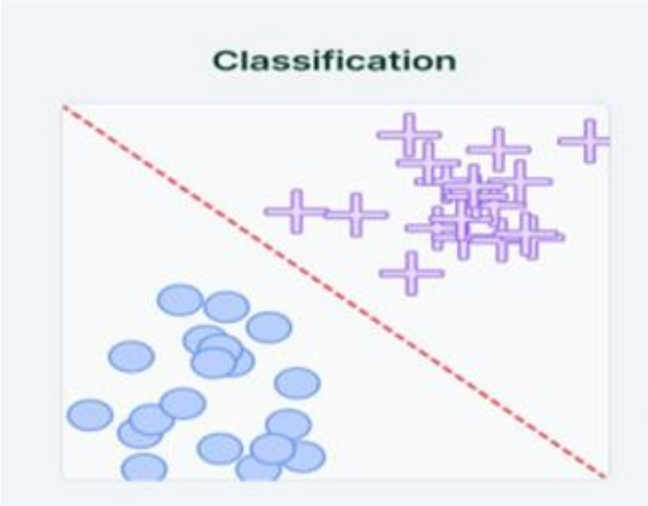


Figure 17. Graph of Classification

3.1.1.1.2. Regression

Regression algorithms handle regression problems where input and output variables have a linear relationship Figure 18 illustrates that. These are known to predict continuous output variables. Examples include weather prediction, market trend analysis, etc [31].

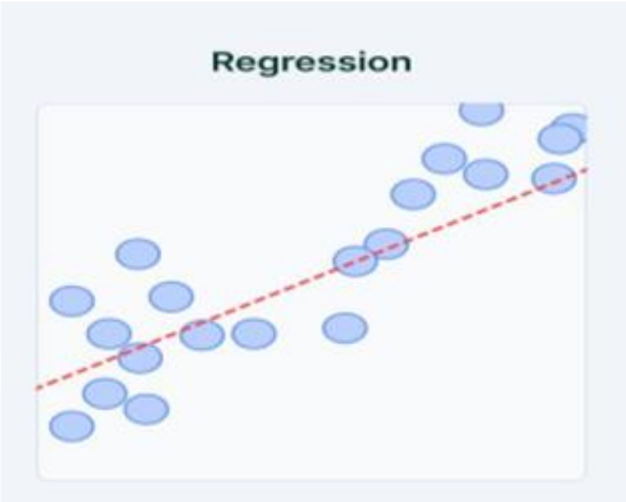


Figure 18. Graph of Regression

3.1.2. Unsupervised Learning

Unsupervised learning deals with unlabeled data means here we have input data and no corresponding output variable. This is the opposite of supervised machine learning. In unsupervised learning, the users do not need to teach/supervise the model. There is no correct output and no supervisor to teach. The algorithm itself learns from the input data and discovers the patterns and information from the data to learn and group the data according to similarities [32].

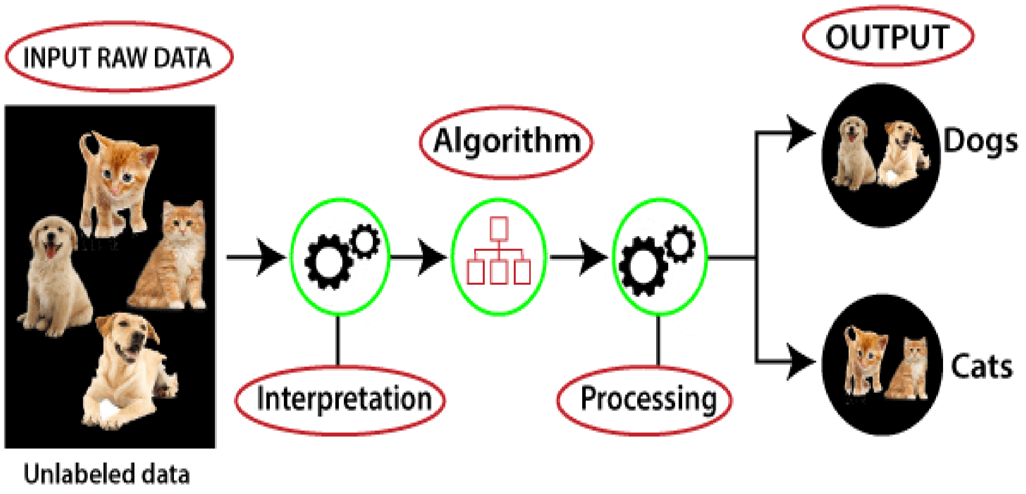


Figure 19. Unsupervised Learning

As shown in Figure 19, suppose we have an image dataset of different types of cats and dogs. The unsupervised learning algorithm is given that dataset. The algorithm is never seen/ trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithms will perform this task by clustering the image dataset into groups according to the similarities between the images.

3.1.2.1. Types of Unsupervised machine learning

Unsupervised machine learning is further classified into two types:

1. Clustering
2. Association

3.1.2.1.1. Clustering Unsupervised Learning

Clustering is an important concept of unsupervised learning. It is used to find out the hidden structures or patterns in uncategorized data. The clustering algorithm processes the uncategorized data and divides them into different clusters (groups) such that objects with many similarities remain in the same group and have fewer or no similarities with the objects of another group [33], as depicted in the example in Figure 20.

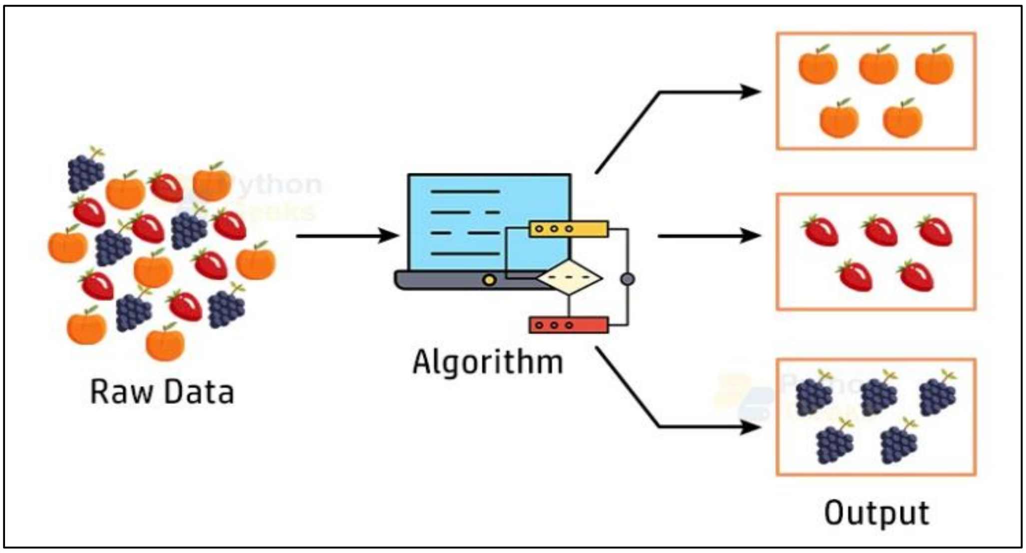


Figure 20. Clustering

Some known clustering algorithms include the K-Means Clustering Algorithm, which we use in our system.

K-Means Algorithm

K-means clustering is a popular unsupervised learning algorithm used for cluster analysis, particularly in scenarios where the data is unlabelled and the goal is to identify natural groupings or clusters within the dataset. The algorithm aims to partition a set of objects into a predetermined number of clusters, denoted as K, in such a way that the sum of the distances between each object and its assigned cluster center, (cluster mean or centroid) is minimized [34].

Figure 21 provides an overview of the data after K-Means.

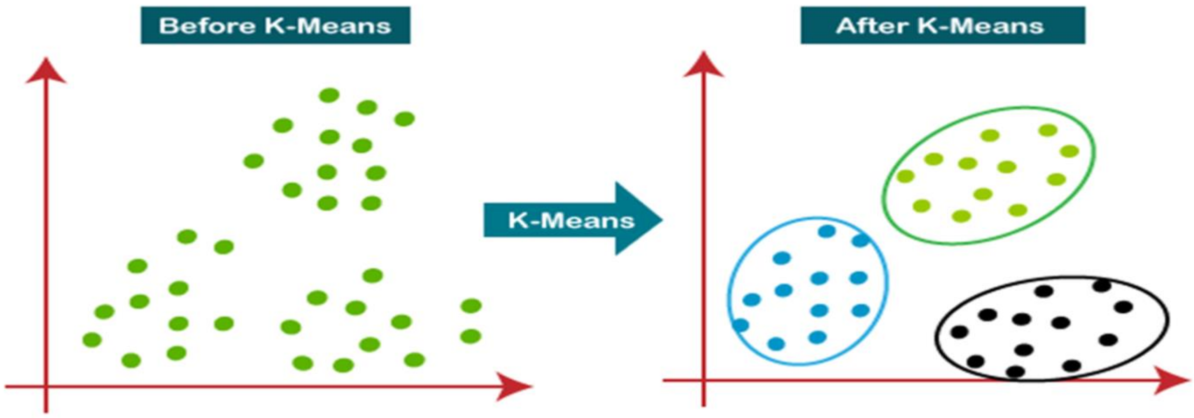


Figure 21. K-Means Algorithm

How K-Means Clustering Works

Here’s how it works [35]:

1. **Initialization:** Start by randomly selecting K points from the dataset. These points will act as the initial cluster centroids.
2. **Assignment:** For each data point in the dataset, calculate the distance between that point and each of the K centroids. Assign the data point to the cluster whose centroid is closest to it. This step effectively forms K clusters.
3. **Update centroids**: Once all data points have been assigned to clusters, recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster.
4. **Repeat:** Repeat steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when a specified number of iterations is reached.
5. **Final Result:** Once convergence is achieved, the algorithm outputs the final cluster centroids.

Here's an example of how it works

Step 1: Suppose we have the following data in ***Figure 22***. The x-y axis scatter plot of these data is given below Select the number K to decide the number of clusters. K=2 It means here we will try to group these datasets into two different clusters.

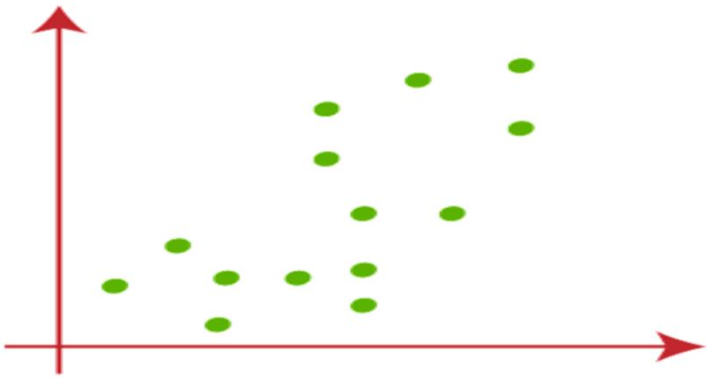


Figure 22. data before K-means

We need to choose some random k points or centroid to form the cluster like in ***Figure 23***. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not part of our dataset.

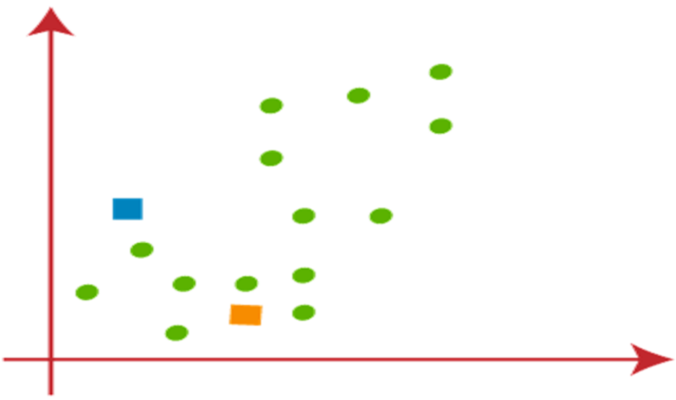


Figure 23. Step 1

Step 2: Now, let's assign data points to clusters based on the nearest centroid using the Euclidean distance.

In 2d the Euclidean distance is the same as Pythagorean theorem: $\sqrt{x^2 + y^2}(1)$

In 3d the Euclidean distance is: $\sqrt{x^2 + y^2 + z^2}(2)$

In 4d the Euclidean distance is: $\sqrt{x^2 + y^2 + z^2 + w^2}(3)$

As illustrated in Figure 27, we will draw a median between both centroids.

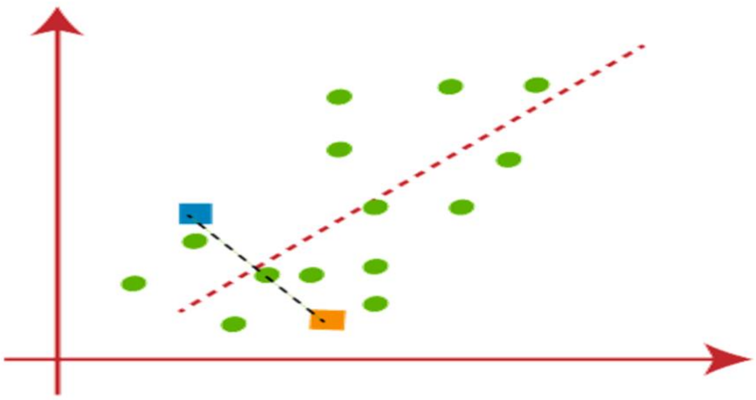


Figure 24. Step 2

According to Figure 24, the points on the left side of the line are near the blue centroid, and the points to the right of the line are close to the orange centroid. Let's color them blue and yellow for clear visualization.

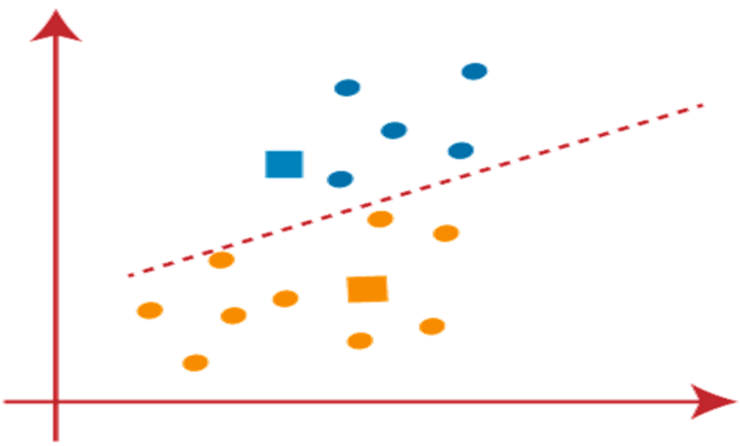


Figure 25. devise the data

Step 3: So we will repeat the process by choosing a new centroid by computing the center of gravity of these centroids, and will find new centroids as below in ***Figure 26***:

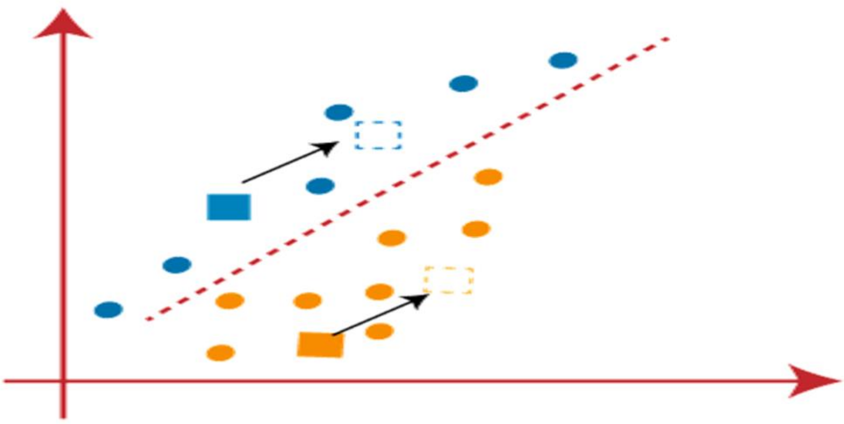


Figure 26. Step 3

The update formula for the centroid ci of cluster i is given by:$c_i = \frac{1}{n}\sum x$ (4)

Where:

• c_i: is the centroid of cluster i.

• n: is the number of data points assigned to cluster i.

• x: iterates over all data points assigned to cluster i.

In simpler terms, to update the centroid of each cluster

Explaining:

Sum the coordinates of all data points assigned to that cluster along each dimension.

Divide the sum by the number of data points assigned to the cluster to compute the mean along each dimension.

The resulting mean values become the new coordinates of the centroid.

Next, we will reassign each data point to the new centroid.

Step 4: We will repeat the process by finding the center of gravity of centroids,As shown in Figure 27.

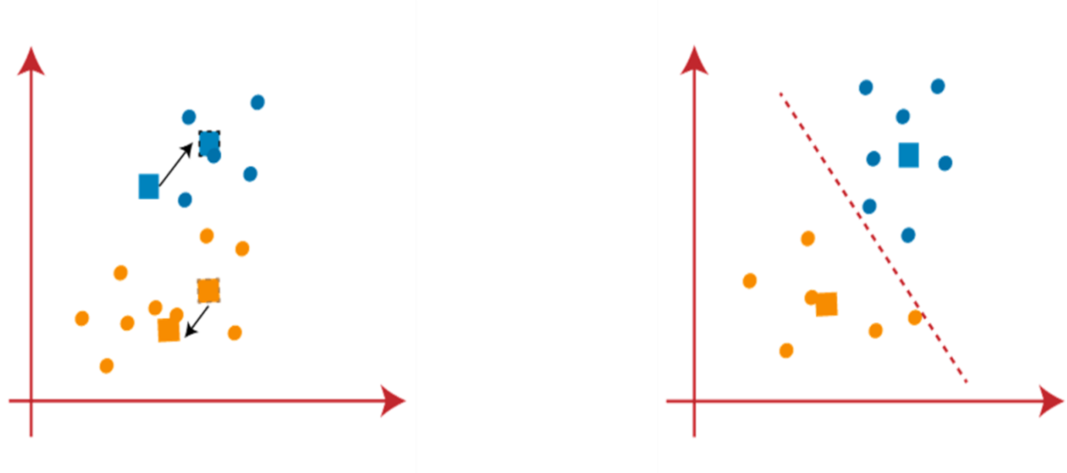


Figure 27. Step 4

Step 5: We repeat this process iteratively until a stopping criterion is met. This could be reaching a maximum number of iterations or until convergence, where the centroids stabilize and do not change significantly between iterations, Figure 28 provides an overview of how data be after k-means

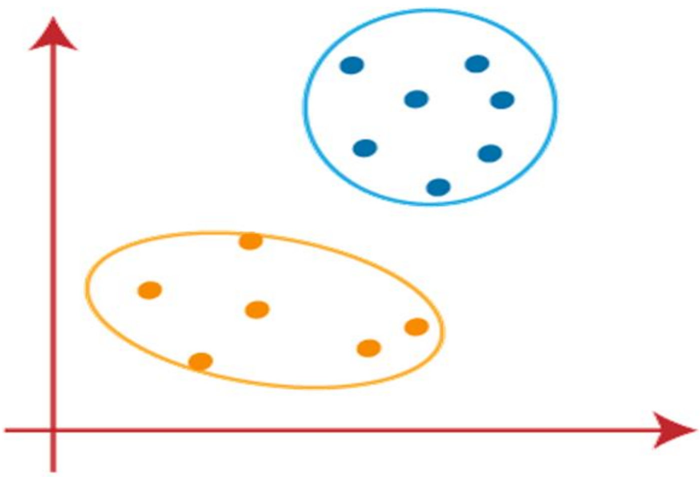


Figure 28. Step 5 data after K-means

How to choose the right K

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS (Within the Cluster Sum of Squares) value [36].

Step 1: It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).

Step 2: For each value of K, calculate the WCSS value.

$$WCSS = \sum P_{i\ in\ cluster1}\, distance(P_iC_i)^2 + \sum P_{i\ in\ cluster2}\, distance(P_iC_i)^2 + \cdots (5)$$

where: $\sum P_{i\ in\ cluster1}\, distance(P_iC_i)^2 (6)$

is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other term.

Step 3: Plot a curve between calculated WCSS values and the number of clusters K, As shown in Figure 29

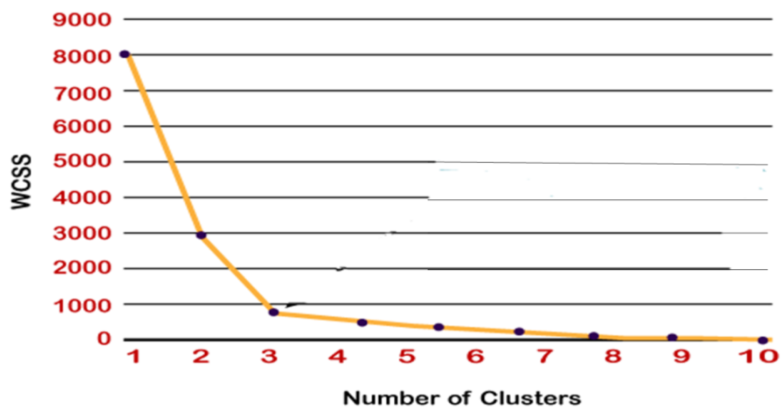


Figure 29. Elbow Curve

Step 4: The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K, like in Figure 30 (k=3).

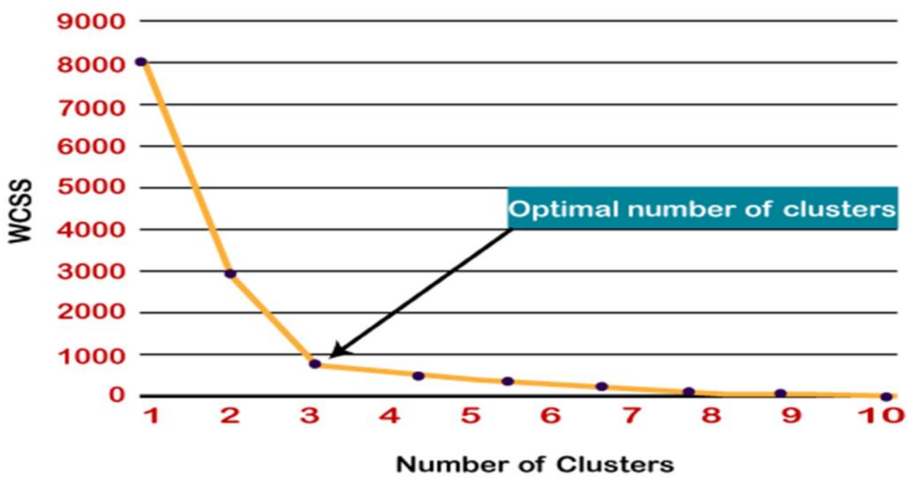


Figure 30. Optimal number of clusters

The objective of k-means clustering

The main objective of k-means clustering is to partition your data into a specific number (k) of groups, where data points within each group are similar and dissimilar to points in other groups [37]. It achieves this by minimizing the distance between data points and their assigned cluster's center, called the centroid [38].

3.1.2.1.2. Association Unsupervised Learning

An association rule is an unsupervised learning method. It is used to find out the relationships between the data in the large dataset. It determines the set of items that occur together in the dataset. The association rule makes marketing strategy more effective [39]. A typical example of the association rule is Market Basket Analysis As shown in Figure 31.

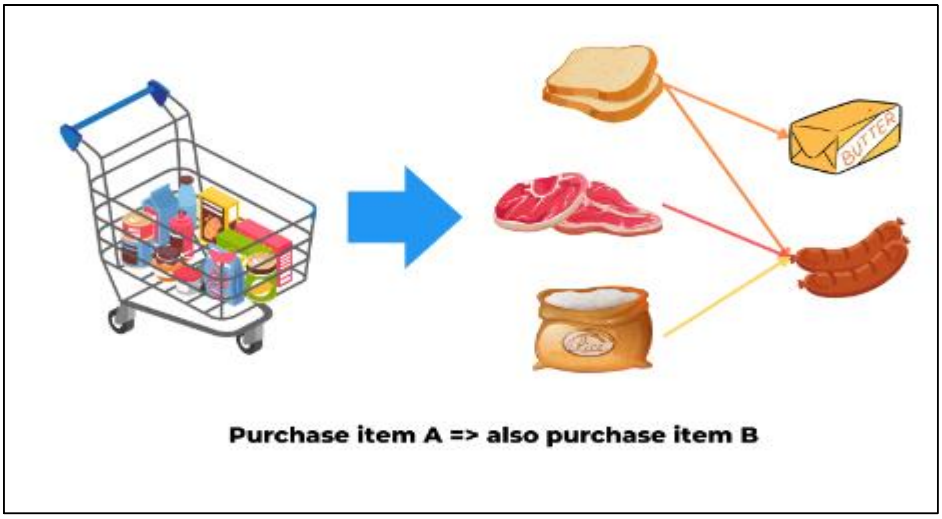


Figure 31. The Association

3.1.3. Reinforcement Learning

In Reinforcement Learning, the machine or agent atomically learns using feedback without any labeled data. Here, the agent learns itself from its experience. In Reinforcement Learning, the agent uses the hit-and-trial process. If the agent tries a successful step, then the agent gets a reward, otherwise, for each mistake, the agent gets a penalty. Its goal is to maximize the total reward [40].

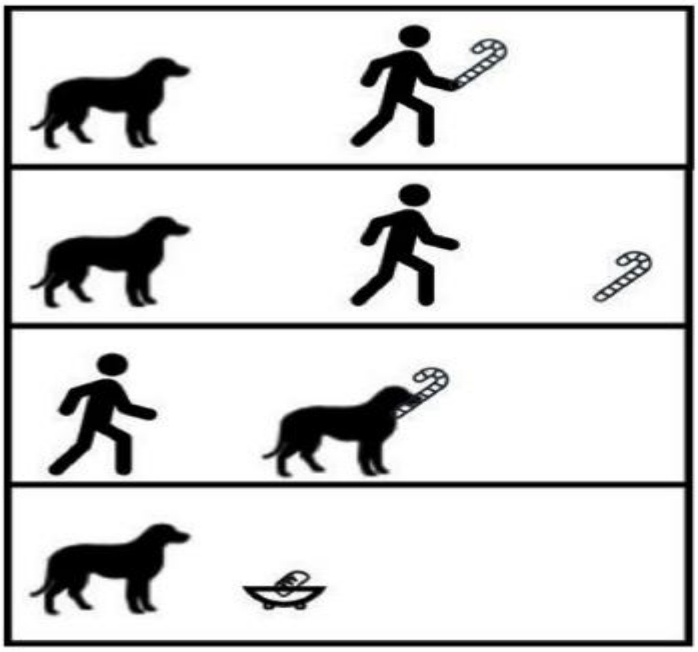


Figure 32. Reinforcement Learning

As shown in Figure 32, consider the example of teaching a dog. Since the dog does not understand any human language, we cannot tell him exactly what to do. We create a situation, and the dog tries to respond in many different ways. If the dog responds desirably, we will give him bread. Now, whenever the dog is presented with the same situation again, the dog does the same action by eagerly anticipating a greater reward (bread). That's like learning that a dog gets from "what to do" from a good experience. At the same time, the dog is learning to do nothing in the face of negative experiences.

3.2. Supervised Learning VS Unsupervised Learning

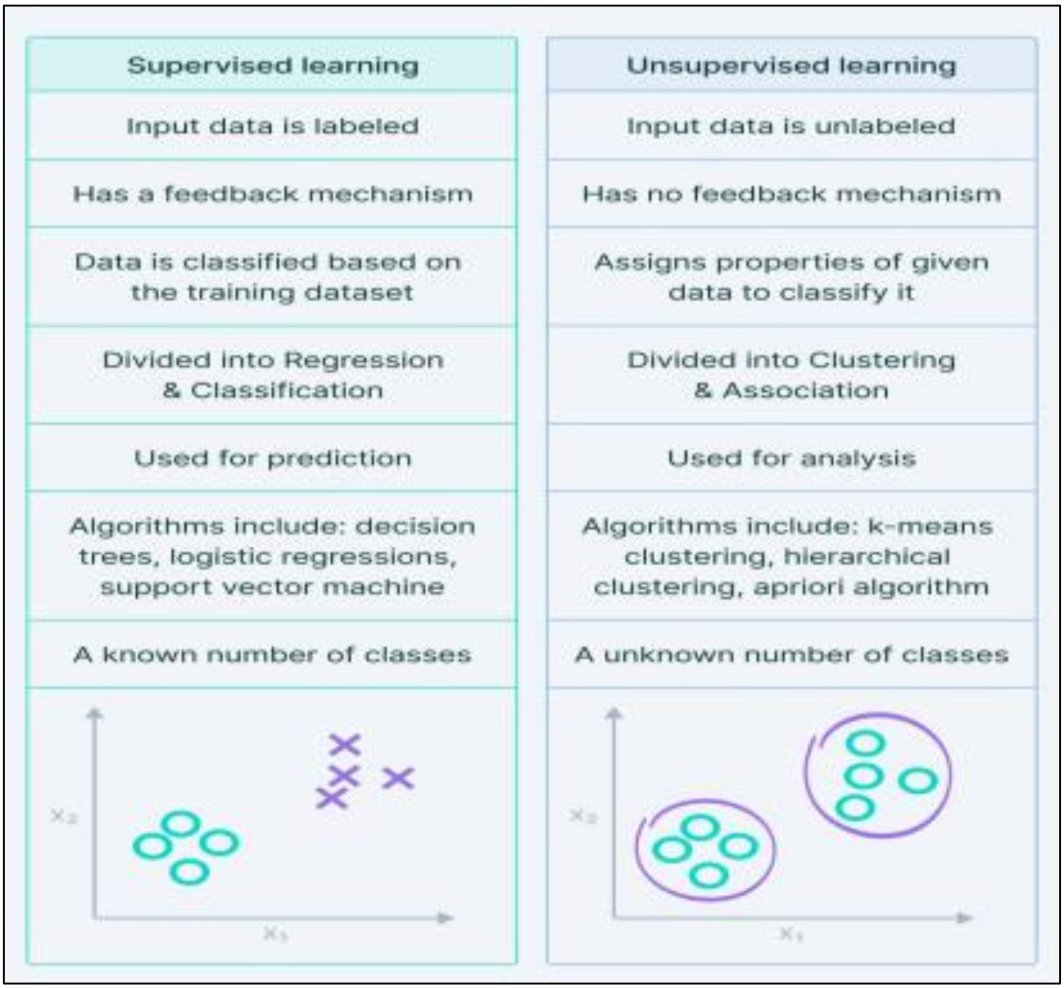


Figure 33. Supervised Learning VS Unsupervised Learning

This is the major difference between supervised and unsupervised learning lies in the presence or absence of classes " Figure 33 ".

4. Deep Learning

Deep learning is a type of machine learning that uses multi-layered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain. Deep learning models can be taught to perform classification tasks and recognize patterns in photos, text, audio, and other various data. It is also used to automate tasks that would normally need human intelligence, such as describing images or transcribing audio files [41].

4.1. Artificial Neural Network (ANN)

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), It is a computational model inspired by the structure and function of the human brain, mimicking the way that biological neurons signal to one another. Artificial neural networks (ANNs) are composed of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and bias [42], the block diagram of Figure 34 shows the mathematical model of a neuron, which forms the basis for designing ANNs.

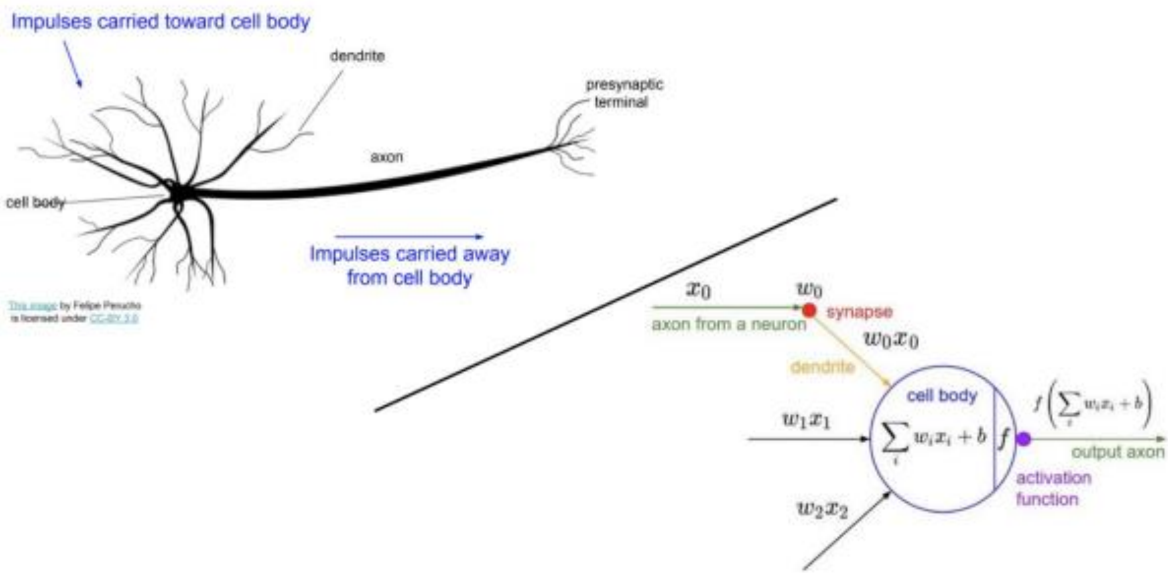


Figure 34.Illustrations of biological and artificial neurons

4.1.1. Structure of Neural Network

4.1.1.1. Layers

A neural network is composed of layers of interconnected nodes that process information As depicted in Figure 35. The three main types of layers are:

1. **Input Layer:** The input layer receives the initial data and passes it on to the next layer. Each node in the input layer represents a feature of the data, such as pixel values in an image or words in a sentence [43].
2. **Hidden Layers:** The hidden layers perform computations on the input data to extract relevant features and patterns. Each node in a hidden layer takes inputs from the previous layer and produces an output that is passed to the next layer [43].
3. **Output Layer:** The output layer produces the final predictions or classifications based on the computations performed in the previous layers. The number of nodes in the output layer depends on the type of problem being solved [43].

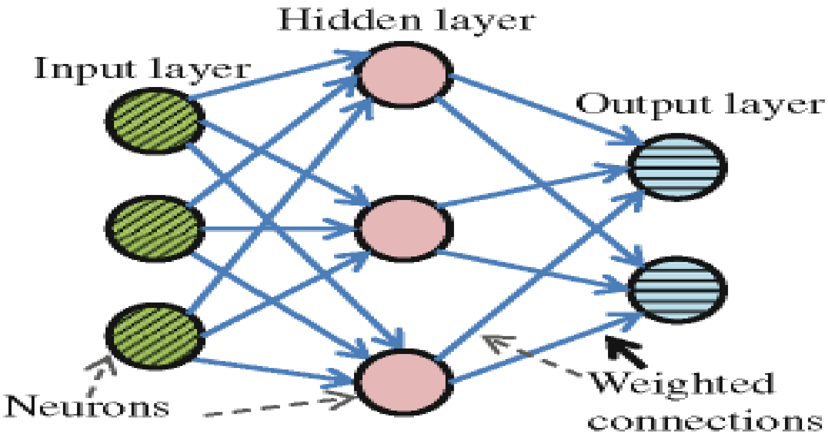


Figure 35. Simple Architecture of ANN

4.1.1.2. Nodes

Artificial neurons are the elementary units of artificial neural networks. The artificial neuron is a function that receives one or more inputs, applies weights to these inputs, and sums them, Usually, each input is separately weighted, and the sum is often added to a term known as a bias before being passed through a nonlinear function known as an activation/transfer function to produce an output Figure 36 illustrates that [44].

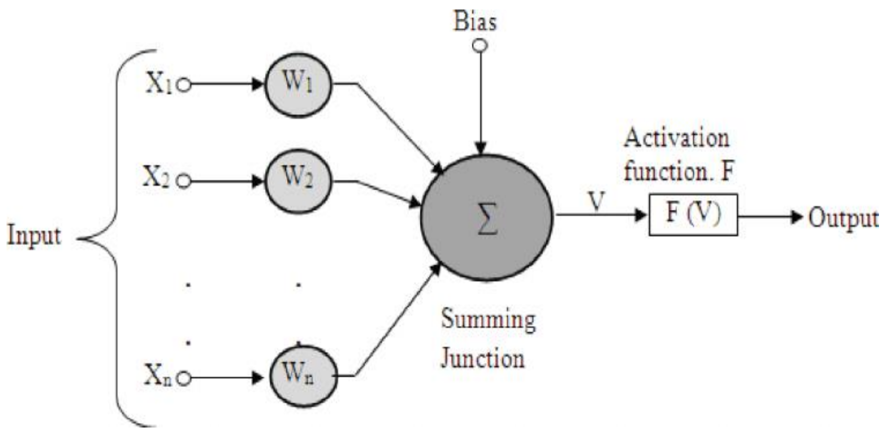


Figure 36. Different nodes of ANN

4.1.2. Training process of ANN

Figure 37 illustrates the Forward and Backward Propagation:

- **Forward Propagation:** This is the simplest type of neural network. Data flows only in the forward direction from the input layer to the hidden layers to the output layer. It may contain one or more hidden layers. All the nodes are fully connected. We do forward propagation to get the output of the model check its accuracy and get the error [45].

- **Backward Propagation:** The backpropagation method is used to train neural networks. If there are a lot of hidden layers, it may be referred to as a deep neural network. Once the forward propagation is completed, we calculate the error. This error is then back-propagated to the network to update the weights [45].

We go backward through the neural network to find the partial derivatives of the error (loss function) concerning the weights. This partial derivative is now multiplied by the learning rate to calculate step size. This step size is added to the original weights to calculate new weights. That is how a neural network learns during the training process [46].

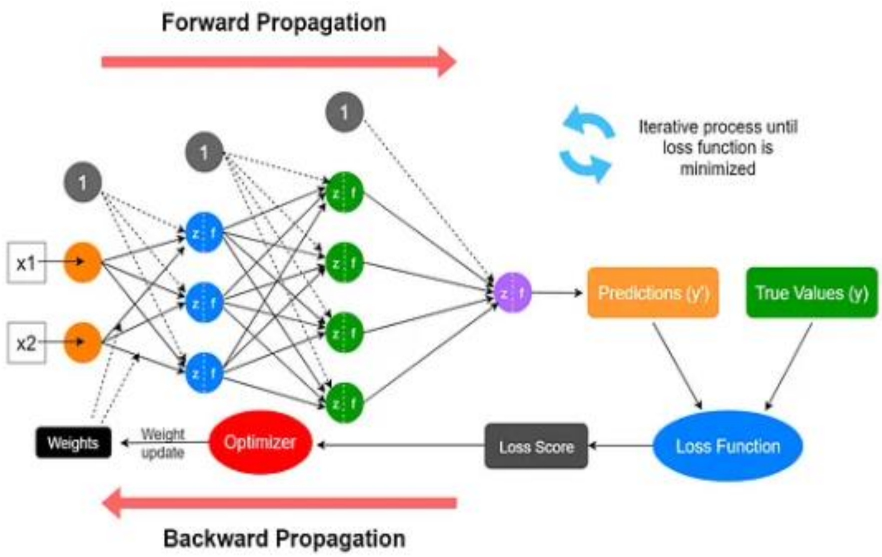


Figure 37. Forward and Backward Propagation

4.1.3. The Activation Function

Activation functions determine whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. It can introduce non-linearity that typically converts the output of a neuron to a value between 0 and 1 or -1 and 1[47].

Common activation functions that can be used are:

- ➔ Sigmoid function
- ➔ Tanh functio

➔ ReLu function

Its graphical representation is shown in ***Figure 38.***

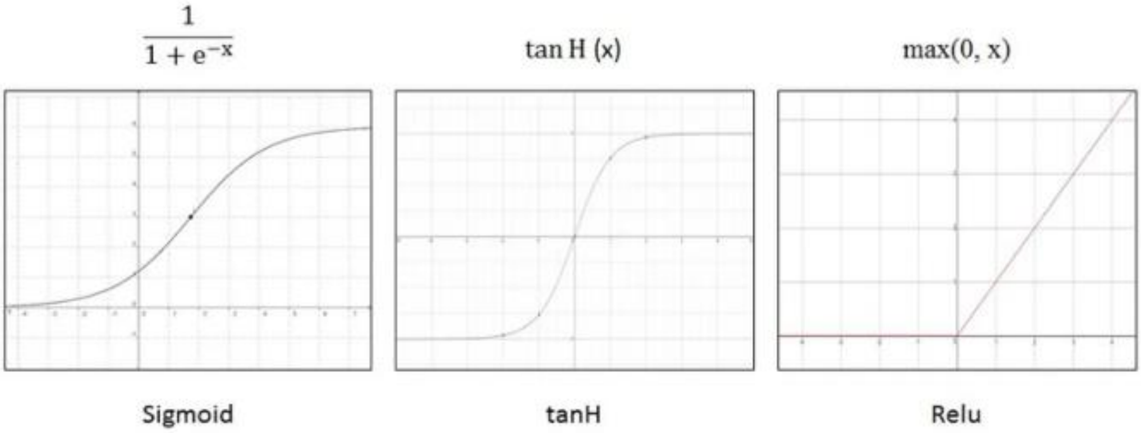


Figure 38. Activation functions.

4.1.4. Types of Neural Networks

Here is the list of the most popular Neural network algorithms [48] :

- ➔ Feedforward Neural Networks (FNN)
- ➔ Convolutional Neural Networks (CNN)
- ➔ Recurrent Neural Networks (RNN)
- ➔ Long Short-Term Memory Networks (LSTM)

4.1.4.1. Recurrent Neural Network

A recurrent Neural Network (RNN) is a type of ANN that can handle sequential data, such as text, speech, or time series. These networks allow machines to “remember” past information and use it to make real-time decisions. While the traditional NN assumes that inputs and outputs are independent of each other, RNN shares parameters across each layer of the network [49]. Figure 39 illustrates the hidden layer of RNN.

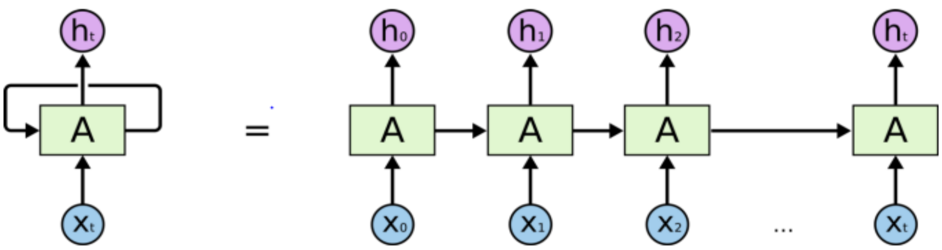


Figure 39. Recurrent Neural Network

4.1.4.1.1. RNN Architecture

we show the architecture of RNN in ***Figure 40.***

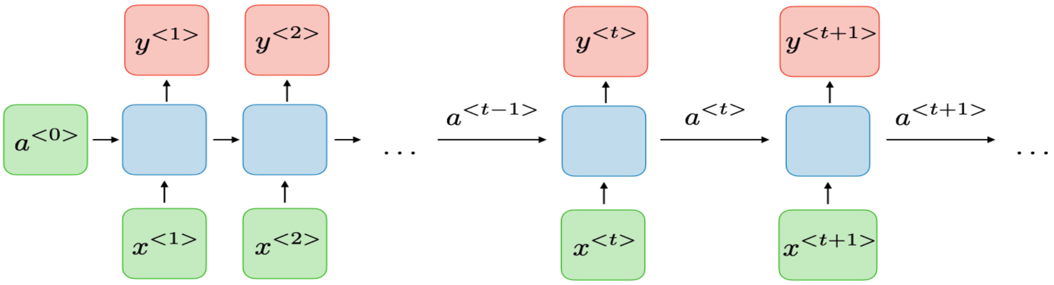


Figure 40. RNN architecture.

4.1.4.1.2. RNN architecture (of one cell)

- **T:** stands for the time step
- **x (t)** is the input
- **y(t)** is the output
- **a(t)** is an activation output
- w_{aa}, w_{ya}, w_{ax} represent the weight matrices (e.g., index 'ya': 'y': the operation result \ output, 'a': the value(s) by which the weight matrix is multiplied.
- b_y, b_a are biases
- g_1, g_2 are activation functions
- Notice the parameters are shared over the different time steps. The role of the activation output is to pass the historical information to the next time steps, which allows the network to make reliable predictions.

$$y^{(t)} = softmax(w_{ya}a^{(t)} + b_y)(7)$$

$$a^{(t)} = activation\_func\ (w_{aa}a^{(t-1)} + w_{ax}x^{(t)} + b_a)(8)$$

As depicted in Figure 40 and Figure 41.

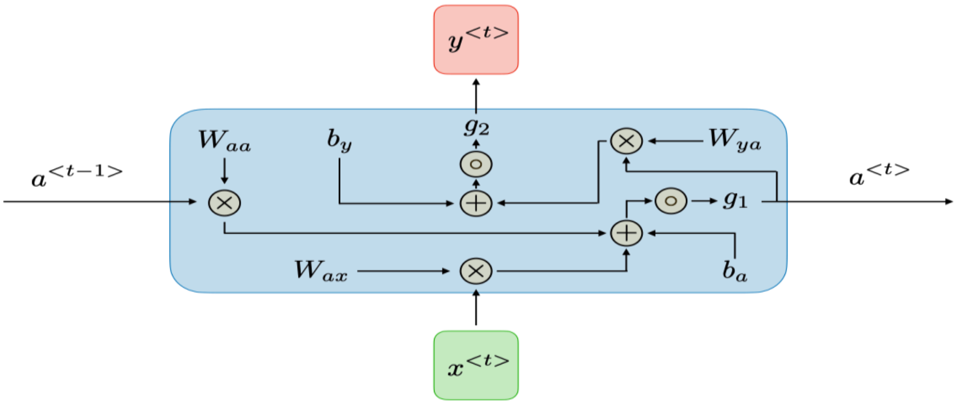


Figure 41. RNN architecture (of one cell).

4.1.4.1.3. RNN Types

- **One to one**

This type of neural network is known as the Vanilla Neural Network. It's used for general machine learning problems, which have a single input and a single output [50]. As shown in Figure 42.

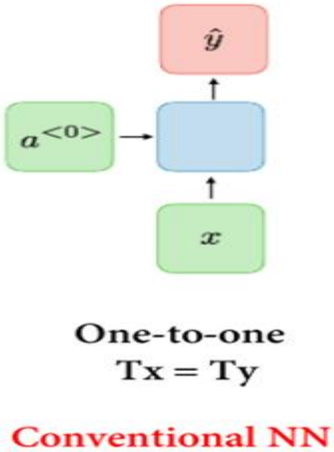


Figure 42. Type One to one

- **One to many**

This type of neural network has a single input and multiple outputs [50] Figure 43 illustrates that. An example of this is the image caption.

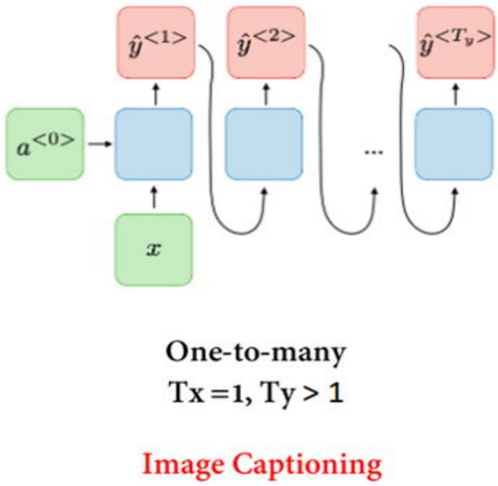


Figure 43.Type One to many

- **Many to one**

This RNN takes a sequence of inputs and generates a single output [50]. Figure 44 illustrates that. Sentiment analysis is a good example of this kind of network where a given sentence can be classified as expressing positive or negative sentiments.

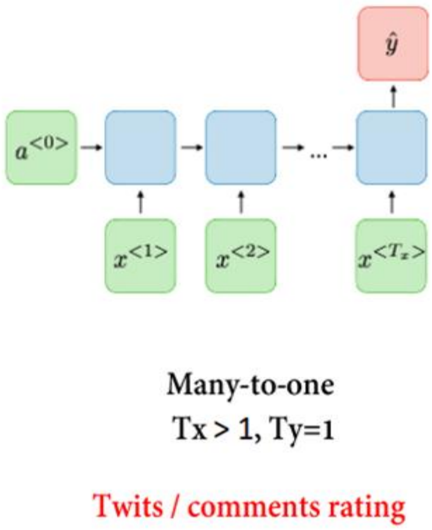


Figure 44. Type Many to one

- **Many to many**

This RNN takes a sequence of inputs and generates a sequence of outputs [50]. Figure 45 illustrates that. Machine translation is one of the examples.

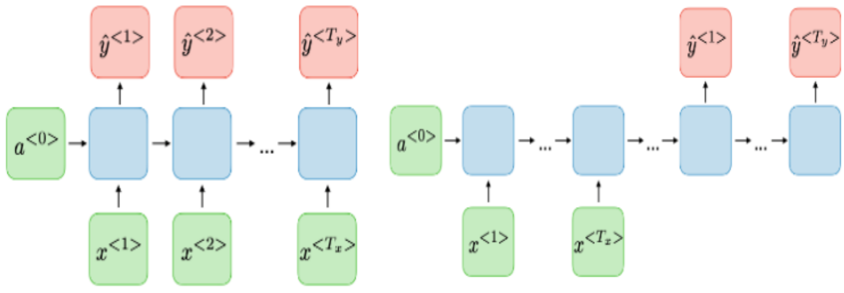


Figure 45. Type Many to many

4.1.4.1.4. Training through RNN

1. A single-time step of the input is provided to the network.
2. Then calculate its current state using a set of current input and the previous state.
3. The current ht becomes ht-1 for the next time step.
4. One can go as many time steps according to the problem and join the information from all the previous states.
5. Once all the time steps are completed the final current state is used to calculate the output.
6. The output is then compared to the actual output i.e. the target output and the error is generated.
7. The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained using Backpropagation through time [51].

4.1.4.1.5. Benefits of RNN

1) Processes sequential data.
2) Can memorize and store previous results.
3) Takes into account both the current and the previous results in the computation of new results.
4) Regardless of the increasing size of the input, the model size remains fixed.
5) It shares weights with other units across time.

4.1.4.1.6. Cons of RNN

1) The computation time is slow as it is recurrent.
2) Unable to process a long sequence of information if using tanh or ReLU activation functions.
3) Cannot process future data in the computation of current data.
4) Training is complicated.

4.1.4.1.7. Two RNN Main Limitations

1. **Vanishing gradient:**with the long sequence of data this problem will be more complicated. The regular gradient will allow appropriate updates of the network parameters, whereas, if the gradient becomes too small, the update of parameters will be insignificant (especially if the size of the data sequence is high, as we will have successive multiplication of small numbers = almost zero) [52].

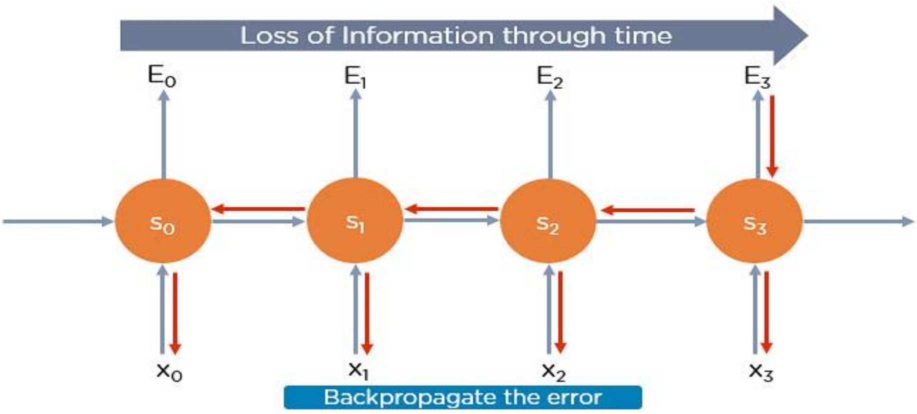


Figure 46. The Vanishing gradient problem.

2. ***Exploding gradient:***With the accumulation of large error gradients, the gradient will explode (especially if the size of the data sequence is high, as we will have successive multiplication of high numbers). This will cause the network to make large updates, which makes it difficult to reach convergence (because updates are done according to the gradient descent rule), and leads to poor performance of the network [53].

4.1.4.1.8. Why LSTM

RNNs are not able to memorize data for a long time and begin to forget their previous inputs. To overcome this problem of vanishing and exploding gradient LSTM is used as a solution for short-term memory learning. Also in RNN when new information is added RNN completely modifies the existing information. RNN is not able to distinguish between important or not important information. Whereas in LSTM there is a small modification in existing information when new information is added because LSTM contains a gate which determines the flow of information.

4.1.4.2. Long short-term memory (LSTM)

LSTM is a special kind of RNN, capable of learning long-term dependencies. LSTM is designed to handle the issue of vanishing or exploding gradients, which can occur when training traditional RNNs on sequences of data. This makes them well-suited for tasks involving sequential data, such as natural language processing (NLP), speech recognition, and time series forecasting [54]. Figure 47 explains the LSTM architecture.

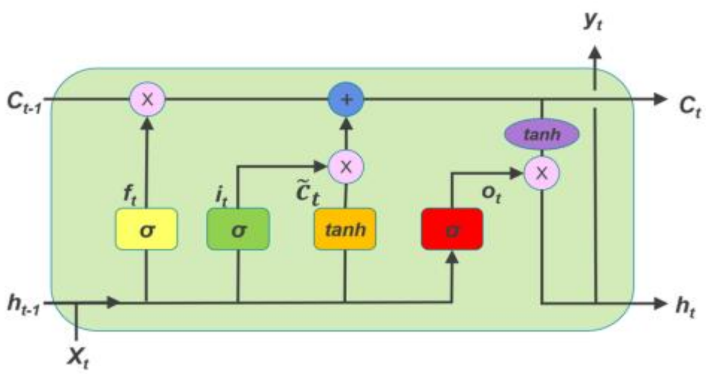


Figure 47.Long short-term memory architecture.

4.1.4.2.1. How LSTM work

The gates decide which data is important and can be useful in the future and which data has to be erased. The three gates are the input gate, output gate and forget gate [55].

Step 1: Decide How Much Past Data It Should Remember

- **Forget Gate:** This gate decides which information is important and should be stored and which information to forget. It removes the non-important information from neuron cells. This results in the optimization of performance. This gate takes 2 inputs one is the output generated by the previous cell and the other is the input of the current cell. Following the required bias, weights are added and multiplied and the sigmoid function is applied to the value. A value between 0 and 1 is generated and based on this we decide which information to keep. If the value is 0 the forget gate will remove that information and if the value is 1 then the information is important and has to be remembered [56].

$$f^{(t)} = \sigma(w_f h^{(t-1)} + w_f x^{(t)} + b_f)(9)$$

Step 2: Decide How Much This Unit Adds to the Current State

- **Input Gate:** This gate is used to add information to neuron cells. It is responsible for what values should be added to a cell by using an activation function like sigmoid. It creates an array of information that has to be added. This is done by using another activation function called tanh. It generates a value between -1 and 1. The sigmoid function acts as a filter and regulates what information has to be added to the cell [57].

$$i^{(t)} = \sigma(w_i h^{(t-1)} + w_i x^{(t)} + b_i)(10)$$

- The candidate value(s) $N^{(t)}$ is given by

$$N^{(t)} = tanh(w_c h^{(t-1)} + w_c x^{(t)} + b_n)(11)$$

Step 3: Decide What Part of the Current Cell State Makes It to the Output

- **Output Gate:** This gate is responsible for selecting important information from the current cell and showing it as output. It creates a vector of values using the tanh function which ranges from -1 to 1. It uses previous output and current input as a regulator which also includes the sigmoid function and decides which values should be shown as output [57].

$$o^{(t)} = \sigma(w_o h^{(t-1)} + w_o x^{(t)} + b_o)(12)$$

$$h^{(t)} = o^{(t)} \times tanh(c_t)(13)$$

$$y^{(t)} = softmax(w_y \times h^{(t)} + b_y)(14)$$

Now, more formally, the forget gate is used to predict the value of $c^{(t)}$

$$c^{(t)} = \left(f^{(t)} \times c^{(t-1)}\right) + \left(i^{(t)} \times N^{(t)}\right)(15)$$

$$\left(f^{(t)} \times c^{(t-1)}\right)(16) \quad \textbf{\textit{What we will forget}}$$

$$\left(i^{(t)} \times N^{(t)}\right)(17) \ \textbf{\textit{What we will add}}$$

$$\left(\mathrm{f}^{(t)} \times \mathrm{c}^{(t-1)}\right) = 0(18) \ \ \textbf{\textit{if}} \ \ \mathrm{f}^{(t)} = 0 \quad \textbf{\textit{(forget everything)}}$$

$$\left(\mathrm{f}^{(t)} \times \mathrm{c}^{(t-1)}\right) = \ \mathrm{c}^{(t-1)}(19) \ \ \textbf{\textit{if}} \ \ \mathrm{f}^{(t)} = 1 \quad \textbf{\textit{(forget nothing)}}$$

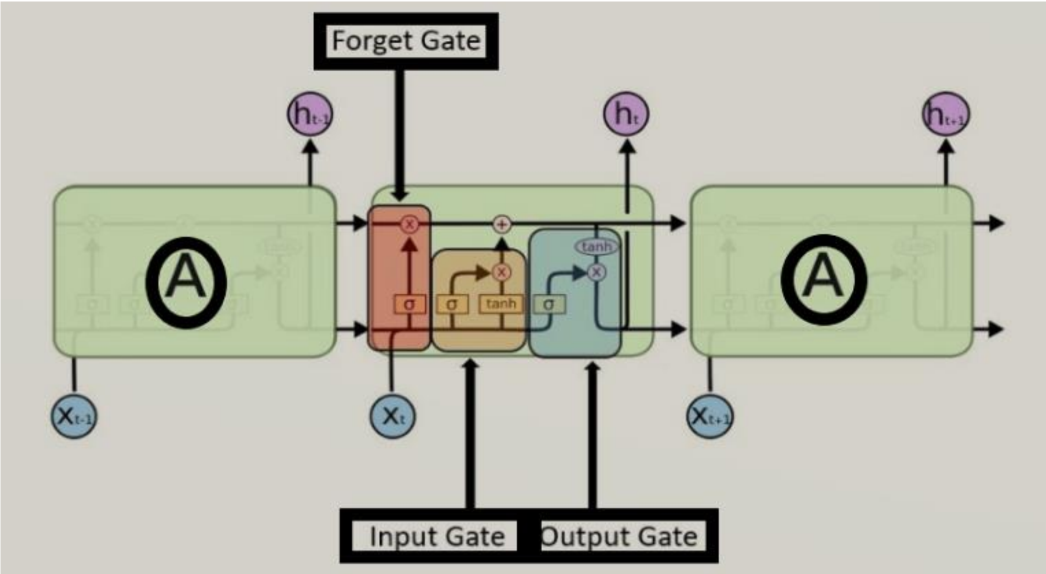


Figure 48. LSTM unit with input and output connections.

4.1.5. Applications of Neural Networks

Artificial neural networks have found widespread applications across various domains, including but not limited to:

- **Computer Vision:** Neural networks are used for image recognition, object detection, and image segmentation in applications such as autonomous vehicles, medical imaging, and surveillance systems.
- **Natural Language Processing:** Neural networks are employed for tasks such as language translation, sentiment analysis, and text generation, enabling advancements in virtual assistants, machine translation, and chatbots.
- **Speech Recognition:** Neural networks are utilized in speech recognition systems to transcribe spoken language into text, enabling applications in voice-controlled devices, speech-to-text software, and voice authentication systems.

- **Financial Forecasting:** Neural networks are applied in predicting stock prices, market trends, and financial risk assessment, aiding in investment decision-making and risk management.
- **Healthcare:** Neural networks are used in medical diagnosis, disease prognosis, and drug discovery, contributing to advancements in personalized medicine and healthcare analytics.

4.1.6. *Challenges and Limitations of Neural Networks*

While artificial neural networks have demonstrated remarkable capabilities, they also present several challenges and limitations that researchers and practitioners continue to address. Some of the key challenges include:

- **Training Data Requirements:** Neural networks often require large amounts of labeled training data to achieve high performance, which can be costly and time-consuming to acquire, especially in domains with limited data availability.
- **Overfitting:** Neural networks are prone to overfitting, where they perform well on the training data but fail to generalize to unseen data, necessitating techniques such as regularization and cross-validation to mitigate this issue.
- **Interpretability:** The inner workings of neural networks can be complex and opaque, making it challenging to interpret their decisions and understand the reasoning behind their predictions, which is crucial in sensitive domains such as healthcare and finance.

5. Conclusion

In conclusion, this chapter has provided a summary of the basic principles of machine learning and deep learning. We talked about the different classifications of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, and the commonly used algorithms in each category. We also introduced the notion of deep learning and its dominant structure, the artificial neural network. At its core, deep learning involves a great deal of field-specific terminology, with many technical concepts coming from other fields of study. It is important to recognize that common terms can have different meanings in the context of deep learning.

chapter V:

V Experimentation and Results

1. Introduction

In this chapter, we detail the experimental setup and results obtained from our predictive modeling of vehicle trajectories using an LSTM model. The experiments were conducted using a dataset of latitude and longitude coordinates, normalized and processed to create time series data. The results include model training, evaluation, prediction, and clustering of predicted positions.

2. Experimentation and Results

We categorize the tools used into two segments: those for constructing the RNN model and those for developing the software.

2.1. Tools and Programming Languages for LSTM Model Development

Python: is a powerful high-level, interpreted, interactive, and object-oriented scripting language created by Guido Van Rossum in the late 1980's. The Python programming language is an object-oriented language, which means that it can model real-world entities. Python codes are compiled line-by-line which makes debugging errors much easier and efficient. Python architecture can be divided into three categories [58].

Google Colab: a multimedia document with different information elements (cells) that can contain i) plain or enriched text using Markdown, HTML code, LaTeX, web resources, pictures, or videos, and ii) Python code that can run within the document. Hence, one can combine descriptive text, commands, and plots in the same environment [59].

Keras: Open-source neural-network Library Written in Python, enabling the creation and training of deep learning models with minimal code, Keras enables developers to quickly build artificial neural networks for tasks such as classification, regression, image recognition, and more[60].

TensorFlow: is an open-source machine learning framework that is widely used for creating, training, and deploying machine learning models, especially deep neural networks [61].

2.2. Tools and Programming Languages for Software Development

HTML: (Hypertext Types and Markup Languages) is the language that is used to design web pages. Normally, a web page is displayed in a formatted style, which means that the browser interprets the HTML page to generate a formatted presentation [62].

CSS: (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML, or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media [63].

JavaScript: is an object-oriented programming language. The language is best known for its use as a scripting language on the web. JavaScript is a key building block of Dynamic HTML (DHTML) , a collection of technologies that are included in nearly all web browsers to support the creation of animated and interactive websites [64].

2.3. Python and Libraries

Python serves as the primary programming language for our model development, while several libraries augment its functionality:

TensorFlow: This library was developed by Google in collaboration with the Brain Team. It is an open-source library used for high-level computations. It is also used in machine learning and deep learning algorithms. It contains a large number of tensor operations. Researchers also use this Python library to solve complex computations in Mathematics and Physics [65].

Matplotlib: This library is responsible for plotting numerical data. And that’s why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc [66].

Pandas: Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc [67].

Numpy: The name "Numpy" stands for "Numerical Python". It is the most commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. The Array Interface is one of the key features of this library [68].

Scikit-learn: It is a famous Python library for working with complex data. Scikit-learn is an open-source library that supports machine learning. It supports various supervised and unsupervised algorithms like linear regression, classification, clustering, etc. This library works in association with Numpy and SciPy [69].

2.4. Our System

Our model uses RNN\LSTM technology to predict the next position for each cab and a machine learning (k means) algorithm to make the predicted position in clusters. In this subsection, we will present the database used for building this model, the architecture of the model, and the results obtained:

2.4.1. System Design

We used HTML, CSS, and JavaScript to construct the user interface for our website. The structural underpinning is provided by HTML, which also defines elements such as headings, paragraphs, and graphics. CSS applies layouts, colors, and styles to improve the visual design. JavaScript enhances functionality and adds interactivity. These technologies work together to guarantee that our website is user-friendly, aesthetically pleasing, and functioning.

2.4.1.1. The Home Page

The homepage is designed to introduce visitors to our service, providing essential information and encouraging them to take the next step.

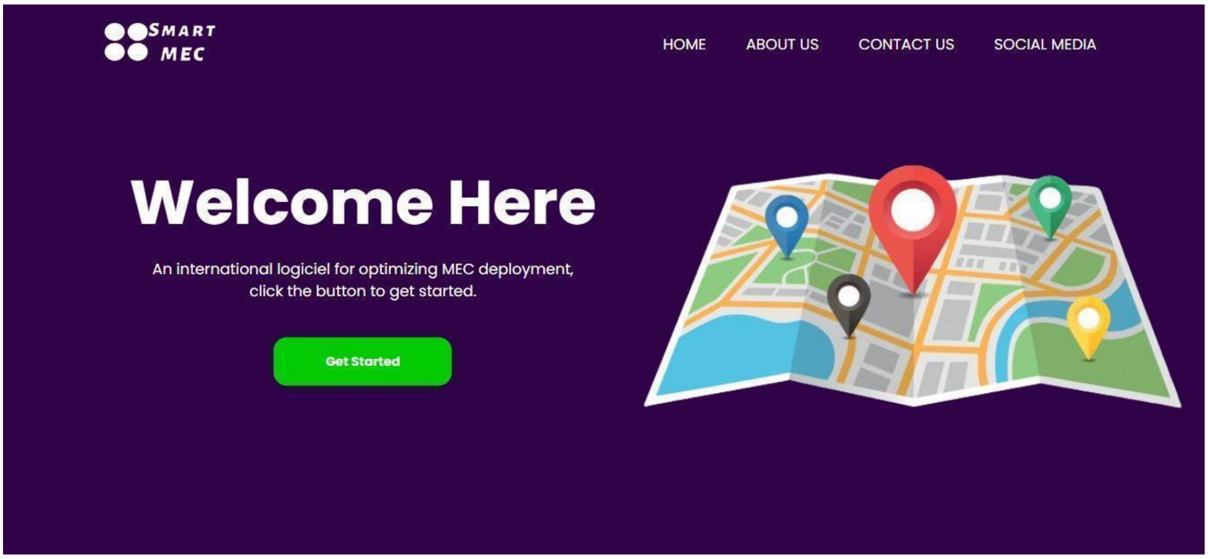


Figure 49. The Home Page

The elements of the home page are:

- **Logo**: Positioned at the top left corner.
- **Welcoming Message**: A warm greeting welcomes visitors to our service.
- **Brief Service Definition**: A concise description of us.
- **Button**: The prominent "Get Started" button invites users to explore further.
- **Photo**: A photo in the theme of our service.
- **Navigation Bar**: To navigate between pages on the website.

User Interaction:

- Users can easily navigate through the website using the navigation bar.
- Clicking on the "Get Started" button guides users to the next step.

2.4.1.2. About us

Our About Us page provides Our team information which is our greatest asset, bringing together diverse expertise outlining our history, values, and achievements. Each member is crucial to our success by having unique skills and forces.

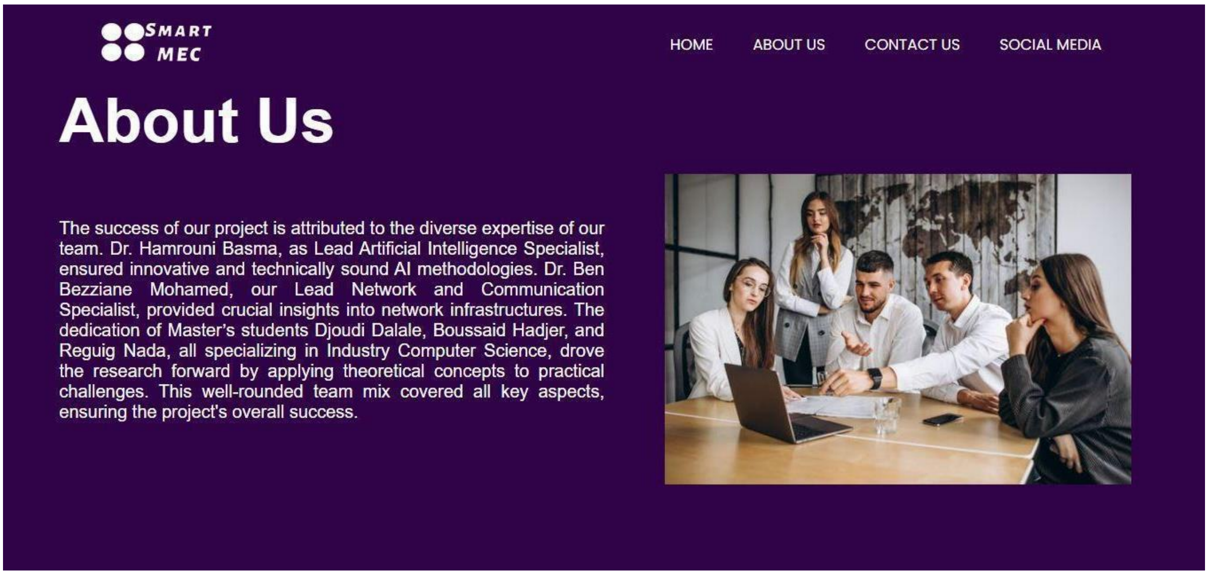


Figure 50. About us

- Dr. Ben Bezziane Mohamed: Lead Network and Communication Specialist
- Dr. Hamrouni Basma: Lead Artificial Intelligence Specialist.
- Djoudi Dalale: Master’s Student in Industry Computer Science.
- BoussaidHadjer: Master’s Student in Industry Computer Science.
- Reguig Nada: Master’s Student in Industry Computer Science.

As a cohesive team, we are a dynamic one that is dedicated to quality and creativity, guaranteeing our projects' overall success.

2.4.1.3. Contact us

Our Contact Us page includes a form for visitors to send emails regarding any problems or questions. A great way to facilitate communication.

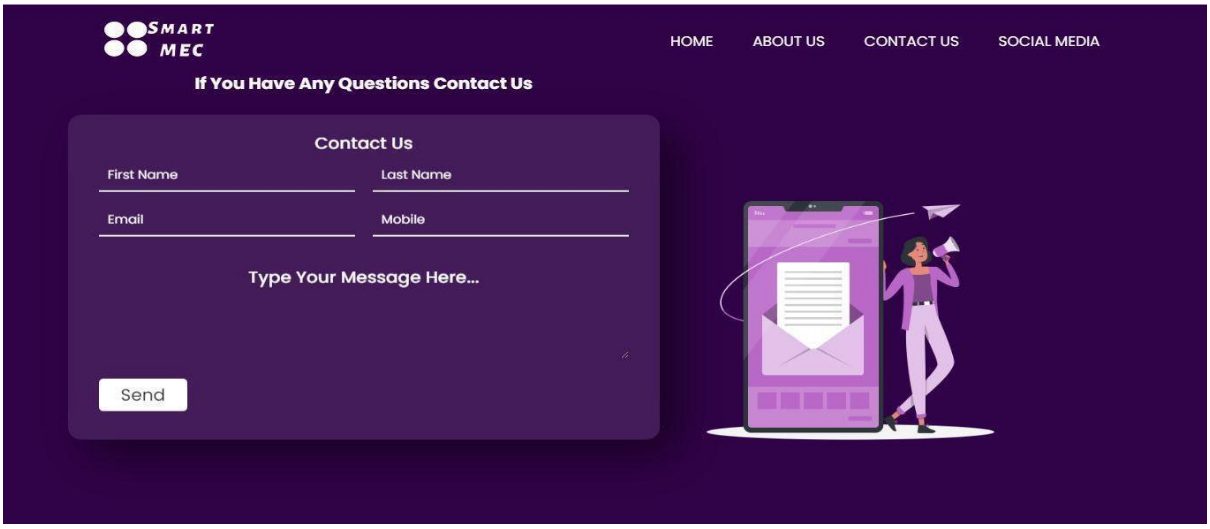


Figure 51. Contact us

Contact Form:

- **First Name:** [Text Field].
- **Last Name:** [Text Field].
- **Email Address:** [Text Field].
- **Mobile:** [Text field].
- **Message:** [Text Area].
- **Send Button:** [Send].

2.4.1.4. Social Media

Recognizing the importance of social media in today's life We added social media links to our website to allow visitors to explore our offering further and keep them regularly updated if they follow us.

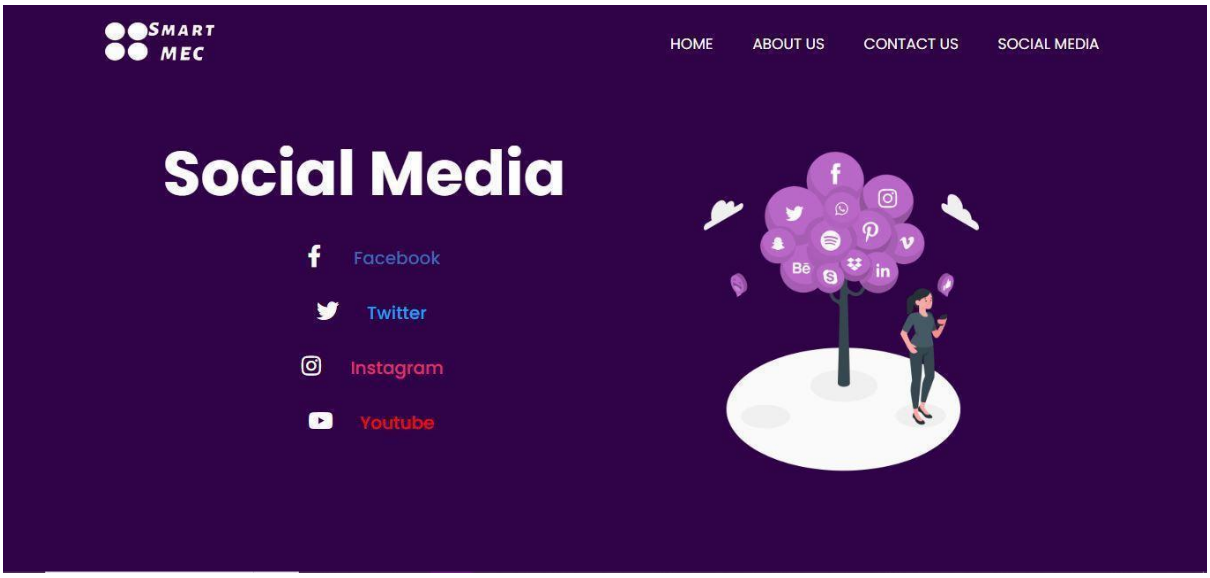


Figure 52. Social Media

2.4.1.5. Getting Started

After getting started with us, it will lead to the next page

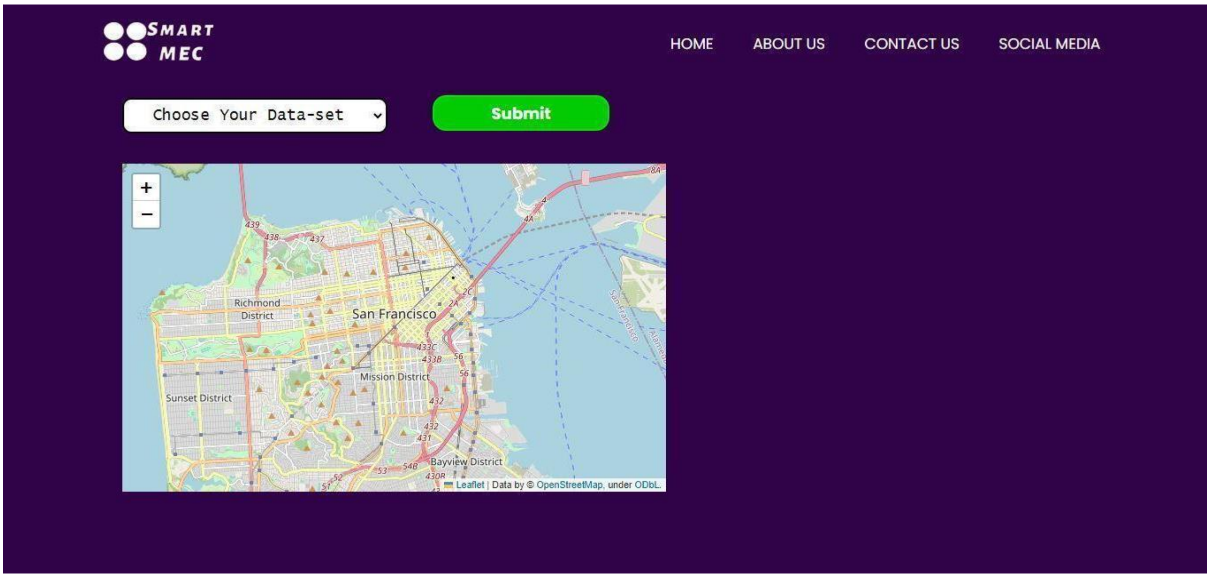


Figure 53. Getting Started

Key elements of this page include:

- Combo Box: At the top of the page is a combo box labeled "Choose Your Data Set." This allows users to select from a list of available datasets.
- Submit Button: A prominent green "Submit" button is next to the combo box. Users click this button after selecting their desired dataset to visualize data.
- Interactive Map: An interactive map is shown below the combo box and submit button. This map is currently empty, awaiting the submission of a dataset. Users can zoom in and out and navigate the map to view different geographic areas.

2.4.1.6. Data Visualization after Submission

The page now includes additional elements that display the results of the data submission:

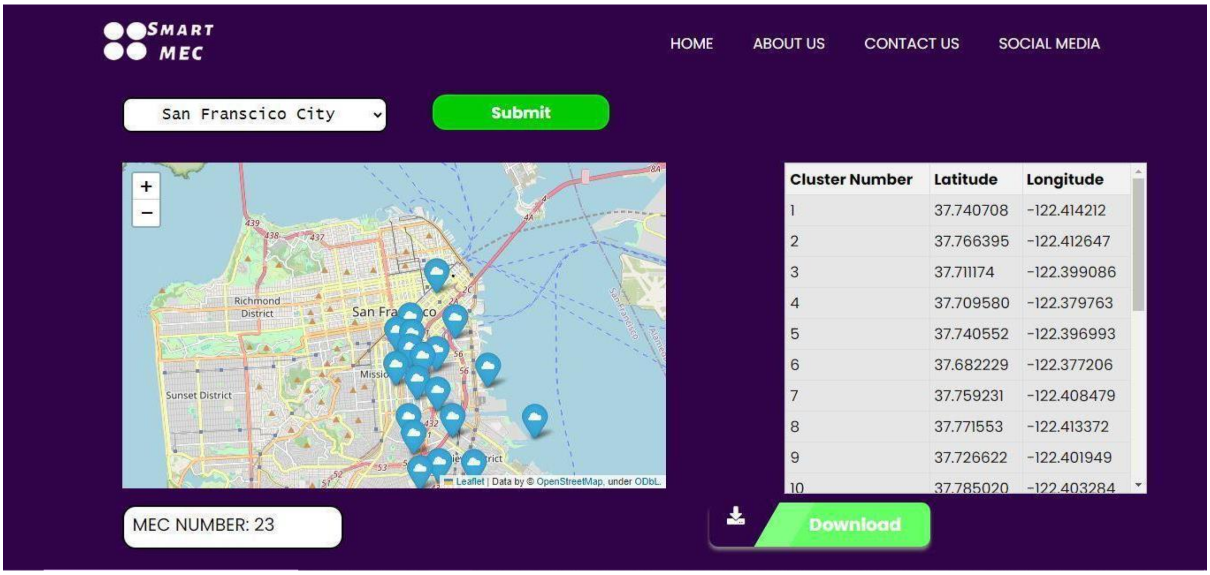


Figure 54. Data Visualization after Submission

- Updated Map: The interactive map now shows multiple data points plotted. Each marker represents a specific location.
- Position Count Label: Below the map, a label displays "MEC NUMBER: 19," indicating the total number of positions (MECs) plotted on the map.
- Data Table: To the right of the map, a detailed table lists the position's cluster numbers, latitude, and longitude values.
- Download Button: At the bottom of the table, a "Download" button allows users to download the table of positions, enabling them to save and further analyze the dataset.

2.4.2. Dataset used

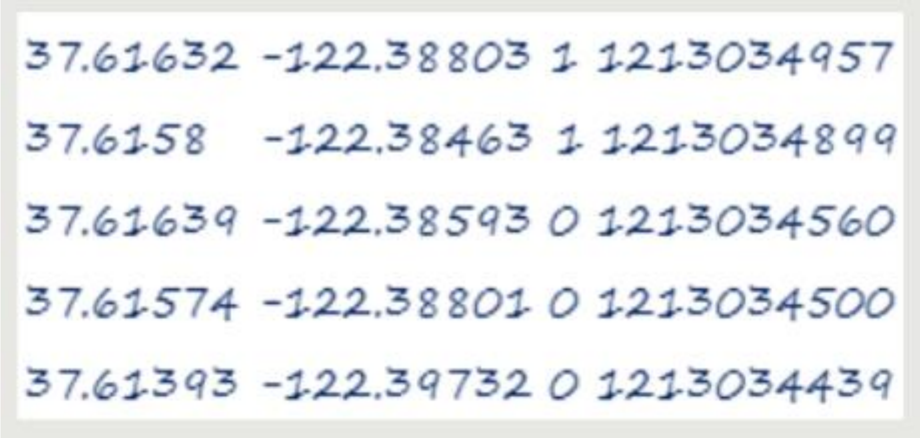
2.4.2.1. Cab Spotting in San Francisco

Cabspotting was a project initiated by the San Francisco Exploratorium and Stamen Design it is among the first uses of real-time data that tracked the movements of taxis in San Francisco using GPS technology from 29 September 2006 till 30 June 2007. The project aimed to visualize the flow and patterns of taxi movement throughout the city.

2.4.2.2. The Data

For this data science challenge, we utilized the **Cabspotting Dataset**. This dataset contains mobility traces of approximately 500 taxi cabs in San Francisco, collected over about 30 days.

The whole dataset is composed of 537 text files, with a total of 11220490 records. Each text file corresponds to a cab. Here's a sample of one of the files in Figure 55 :



37.61632	-122.38803	1	1213034957
37.6158	-122.38463	1	1213034899
37.61639	-122.38593	0	1213034560
37.61574	-122.38801	0	1213034500
37.61393	-122.39732	0	1213034439

Figure 55. cab spotting dataset

Columns are, respectively:

- Latitude
- Longitude
- Occupancy (1 for passengers, 0 for empty)
- Time (number of seconds since the Unix epoch: 00:00:00 UTC)

2.4.3. System Implementation

Our system implementation encompasses data pre-processing and model development phases, executed through the following steps:

2.4.3.1. Importing Necessary Libraries

Python libraries such as Keras, TensorFlow, NumPy, Pandas, Scikit-learn, and Matplotlib are imported to facilitate model development.

2.4.3.2. Load the Dataset

The dataset is loaded into the working environment.

2.4.3.3. Data Preparation

- **Combining Files into CSV:**
    - merged all 536 separate text files in a single CSV file
- **Cleaning Data:**
    - Add a "CarID" Column and assign unique IDs (from 1 to 536) to identify each cab.
    - Removed the "Occupancy" Column.

- ○ transformed the "Timestamp" column from UNIX Epoch to a standard time format. as we show in Figure 56.

	Timestamp	CarID	Latitude	Longitude
0	2008-05-17 15:10:22	1	37.75050	-122.42086
1	2008-05-17 15:51:01	1	37.73494	-122.40670
2	2008-05-17 16:30:44	1	37.78931	-122.42200
3	2008-05-17 17:13:08	1	37.61393	-122.39731
4	2008-05-17 17:51:38	1	37.76632	-122.42462

Figure 56. dataset head

Data Pre-processing

- Position Selection: 500 positions are selected from each cab's data to reduce the dataset size.
- Normalization of Latitude and Longitude: The normalization was done to center the data around a mean value, facilitating better convergence during training.
- Conversion to Time Series Dataset: To create a time series dataset, we used a window size of 3, generating input sequences (X) and corresponding labels (Y).
- Splitting the Data into Training and Testing Sets: The data was split into training and testing sets using an 80-20 ratio.

2.4.3.4. Model Training

2.4.3.4.1. Defining the Model Architecture

- **Input Layer:** Shape (3, 2) representing input data dimensions.
- **LSTM Layer:** Processes sequential data with 100 units and ReLU activation.
- **Dense Hidden Layer:** 200 units with ReLU activation for non-linearity.
- **Dropout Layer:** 20% dropout rate.
- **Output Layer:** 2 units with linear activation for regression predictions.

Table 3. Hyperparameters of the Proposed LSTM Model

Deep learning	**Keras (using TensorFlow backend)**
Deep learning tool	**Tensorflow**
LSTM timestamp Window	**‘timeseries = 3’ specifies the number of time steps considered in each input sequence.**
Activation function	**ReLU for LSTM and Dense layers, Linear for the output layer.**
Optimizer	**Adam optimizer, specified with optimizer='adam' during model compilation.**
Loss function	**Mean Squared Error (MSE), specified with loss='mean_squared_error' during model compilation.**
Droup-out	**Dropout regularization applied with a rate of 0.2 after the first Dense layer.**
Batch size	**32, specified during model training (batch_size=batch_size in lstm_model.fit).**
Number of epoch	**20 epochs, specified in ‘epochs=20’ during model training (lstm_model.fit method).**

2.4.3.4.2. Training the Model

The model was trained for 20 epochs, with the training and validation losses recorded.

2.4.3.4.3. Model Evaluation

We evaluate the performance of the LSTM model on the test dataset using three different metrics:

Loss Function: A loss function, also known as a cost or objective function, quantifies the difference between the predicted values of a machine learning model and the actual target values. The goal during training is to minimize this loss, indicating that the model's predictions are closer to the actual outcomes.

Mean Squared Error (MSE): MSE is one of the most common regression loss functions and an important error metric. In Mean Squared Error, also known as L2 loss, we calculate the error by squaring the difference between the predicted value and actual value and averaging it across the dataset.

MSE formula is:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \quad (20)$$

Where:

- n is the number of data points
- Yi is the actual value of the dependent variable for the i^{th}observation
- $\tilde{y}_i$ is the predicted value of the dependent variable for the i^{th}observation

Mean Absolute Error (MAE): Mean absolute error, or L1 loss, stands out as one of the simplest and easily comprehensible loss functions and evaluation metrics. It computes by averaging the absolute differences between predicted and actual values across the dataset. Mathematically, it represents the arithmetic mean of absolute errors, focusing solely on their magnitude, irrespective of direction. A lower MAE indicates superior model accuracy.

MAE formula is:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (21)$$

Where:

y_i is actual value

$\hat{y}_i$ is predicted value

n is the sample size

In our system, we have achieved

- The loss obtained during evaluation is 0.001814185525290668.
- The Mean Squared Error (MSE) calculated separately is 0.0018141850778880752.
- The Mean Absolute Error (MAE) calculated separately is 0.026066203018259404.

For the test set. “Figure 57” clearly shows the obtained results.

These metrics provide insights into the performance of the LSTM model in predicting the target values. The low values of MSE and MAE indicate that the model performs well in predicting the trajectories of the vehicles.

```
1701/1701 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - loss: 0.0018
LSTM Model Loss: 0.001814185525290668
1701/1701 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step
LSTM Model MSE: 0.0018141850778880752
LSTM Model MAE: 0.026066203018259404
```

Figure 57. The Obtained Results

2.4.3.5. Results

2.4.3.5.1. Loss Curves

Now that we have proven the effectiveness of our model, let’s take a look at some of the outcomes of our model and see how accurate it is. Figure 58 below shows The training and validation losses were plotted to visualize the model's performance over the epochs.

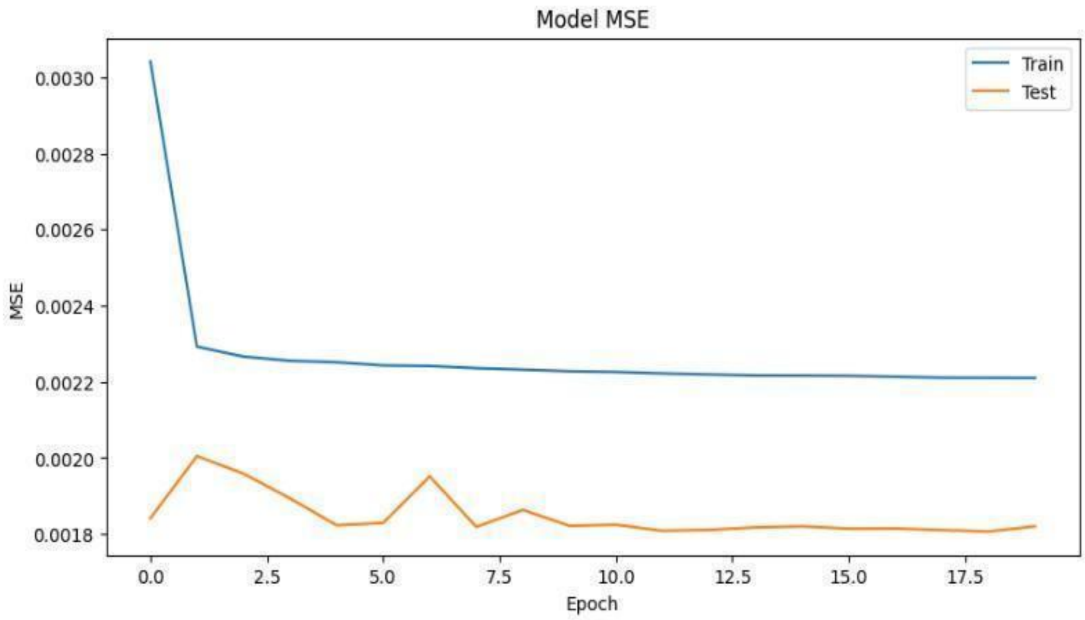


Figure 58. Train and Test Set loss

2.4.5. *Analysis of Predicted Positions for Cab 1*

To verify the validity of the performance of the LSTM model, we focused on the predicted positions of a particular cab (Cab 1). By plotting the actual positions and the predicted position as shown in Figure 59, we found that the predicted position aligns well with the region where the cab is most frequently located, meaning it represents the location of the majority of the cab`s presence.

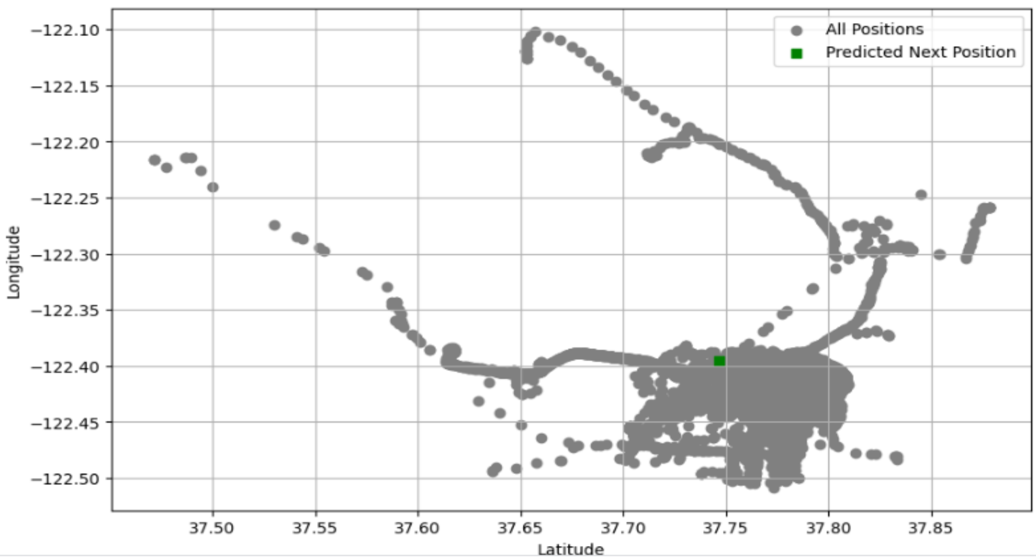


Figure 59. All positions of cab_1 and the predicted point

2.4.6. *Comparison of LSTM and Simple RNN Models*

To compare the performance of the LSTM and Simple RNN models, we calculated the Mean Squared Error (MSE) for both models on the test dataset.

- **Mean Squared Error (MSE) Comparison**: The MSE for both the LSTM and Simple RNN models was calculated and then plotted to evaluate their performance. A lower MSE indicates better prediction accuracy.

  Figures 60,61 below show the loss obtained by the two model

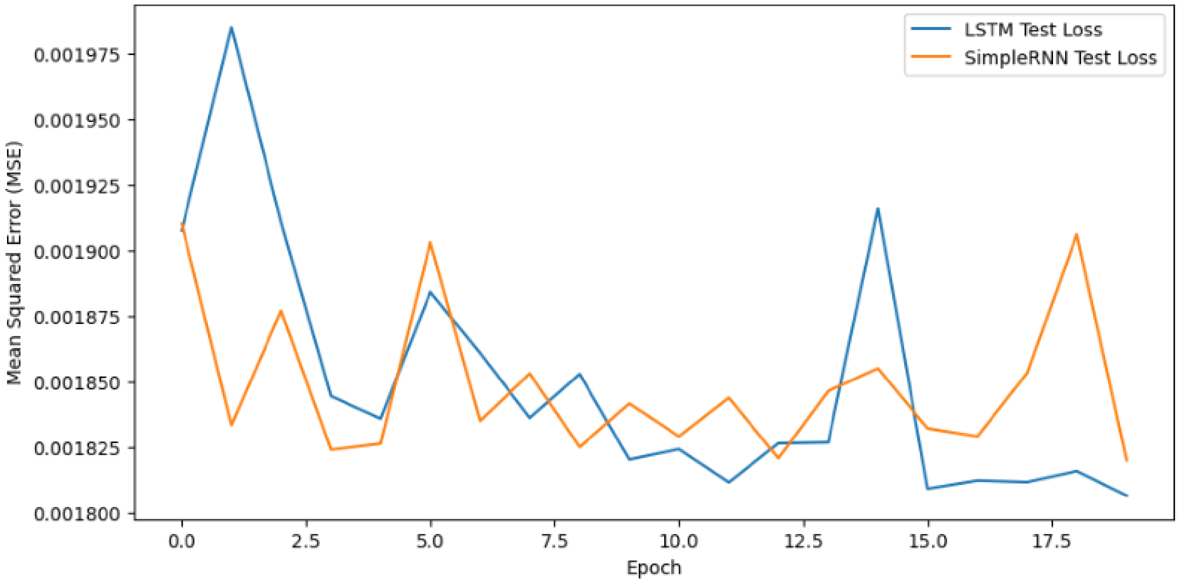


Figure 60. Test Loss comparison: LSTM and simpleRNN

```
1701/1701 ━━━━━━━━━━━━━━━━━━━━ 3s 1ms/step
LSTM Model MSE: 0.0018141850778880752
1701/1701 ━━━━━━━━━━━━━━━━━━━━ 3s 1ms/step
SimpleRNN Model MSE: 0.0018147800234901606
```

Figure 61.The Obtained Results of LSTM and simpleRNN

- **Results**: These MSE values indicate the average squared differences between the actual and predicted values of the target variable (in this case, position) across all instances in the test dataset. Lower MSE values signify better performance, suggesting that the LSTM model slightly outperforms the Simple RNN model in this context, as it has a slightly lower MSE.

2.4.7. *Predicted the next position for each cab*

For each cab in the dataset, we extract the last three positions to forecast its next position using a trained model then I plot the predicted positions as shown in Figure 62. This process provides real-time density and location data for each vehicle, enhancing traffic management and route optimization. The real-time predictions support smart transportation systems by enabling better decision-making and efficient traffic flow management.

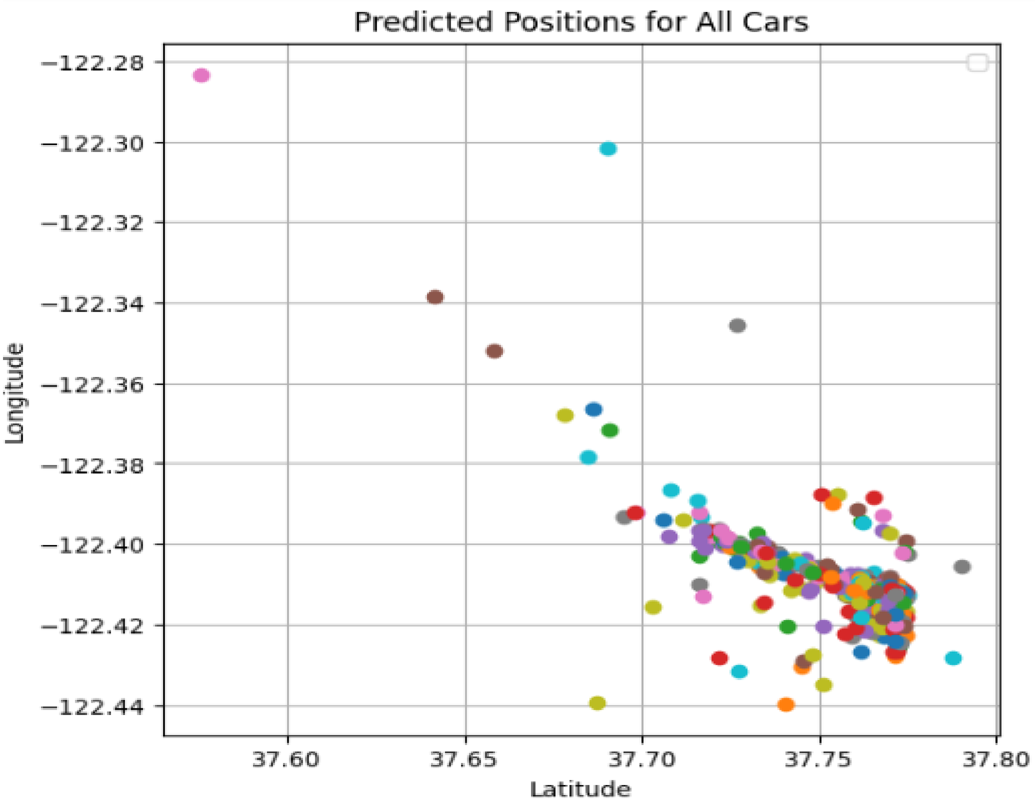


Figure 62. Predicted Positions for All Cars

2.4.8. *Clustering Predicted Positions*

After predicting the next positions for each cab using the trained model, the next step was to group these predicted positions into clusters. Clustering is a technique used to group data points that are similar to each other.

2.4.8.1. Elbow Method for Optimal K

Determining the optimal number of clusters is crucial in clustering analysis to ensure meaningful groupings without unnecessary complexity. We employed the elbow method, a commonly used approach, to find this optimal number. The elbow method works by plotting the inertia values (sum of squared distances of samples to their closest cluster center) against the number of clusters. The point where the inertia begins to decrease more slowly (forming an "elbow" shape in the plot in Figure 63) indicates the optimal number of clusters where adding more clusters does not significantly improve the model's performance.

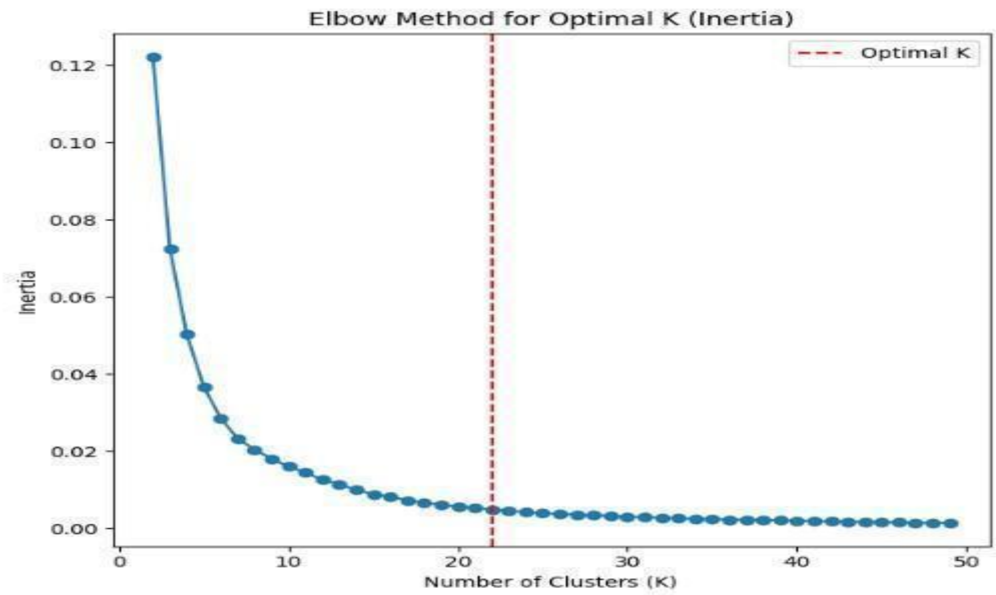


Figure 63. Elbow Method for Optimal K

2.4.8.2. Plotting the Clusters

Once the optimal number of clusters was determined using the elbow method, we applied the K-Means algorithm with this number of clusters to group the predicted positions. Each cluster represents a group of predicted positions that share similar characteristics. Figure 64 illustrates the clusters obtained from this process, providing a visual representation of how the predicted positions are distributed across different groups or clusters.

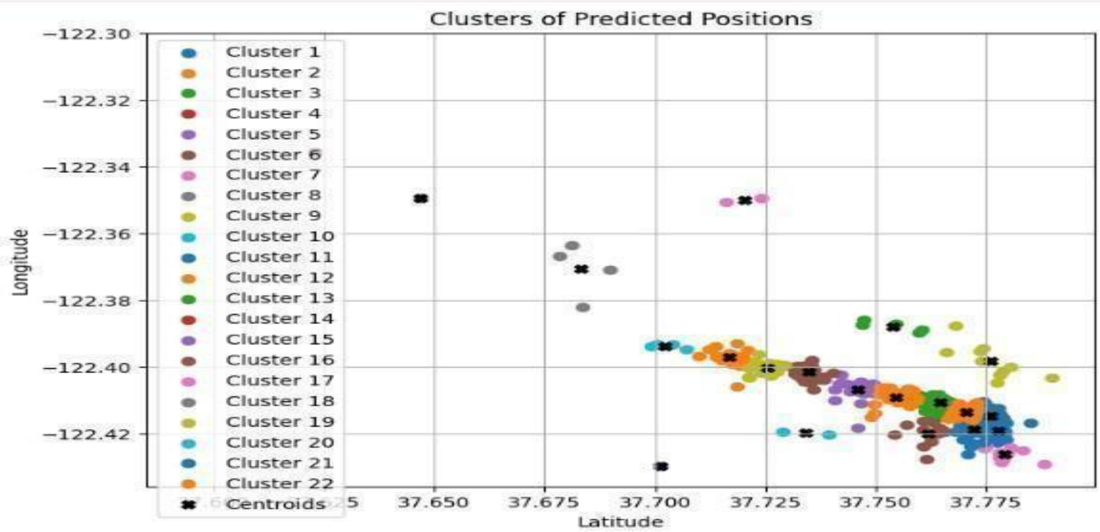


Figure 64. clusters of predicted positions

2.4.8.3. Clustering and Visualization

After clustering predicted positions using the K-Means algorithm, we extract the center of each cluster. These centers are the optimal locations for deploying Mobile Edge Computing (MEC) infrastructure, then we visualize these cluster centers on a geographical map Figure 65.

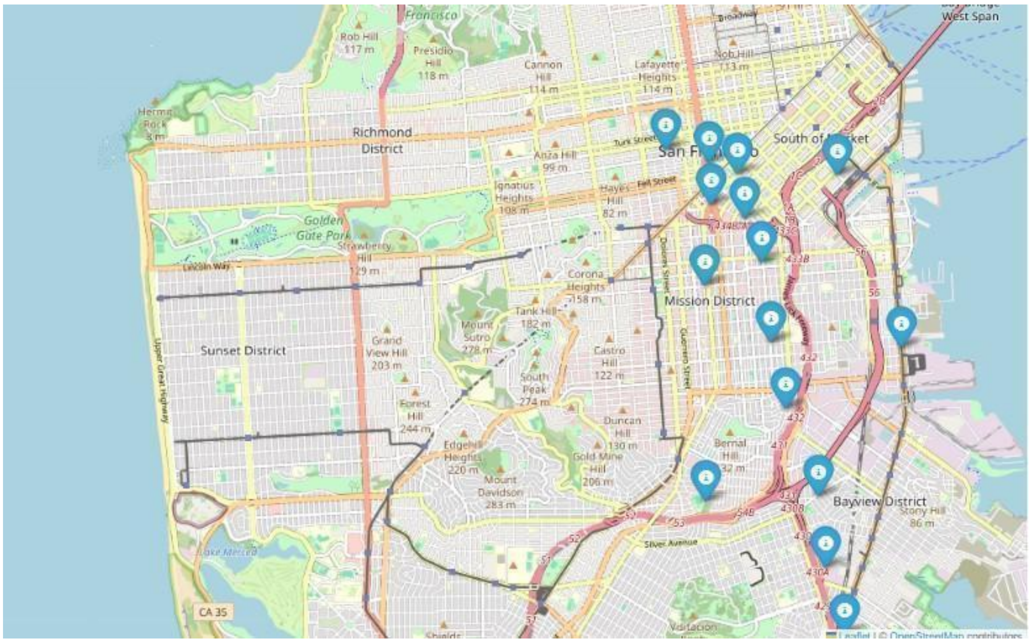


Figure 65. Map of San Francisco with positions of MEC

3. Conclusion

In this chapter, LSTM and RNN models were used to predict vehicle locations and analyze their movements. We observed that the LSTM algorithm showed better capabilities than the simple RNN algorithm for time series prediction. The main objective of this chapter was to leverage the prediction power of LSTM to identify high-density vehicle locations to improve Mobile Edge Computing (MEC) deployment. By using LSTM to monitor vehicle density in real-time and using clustering techniques, we identified the center points of each cluster that indicate the ideal locations for MAC deployment. This integration of LSTM for accurate predictions and clustering for spatial analysis provides a powerful approach to enhance urban infrastructure management and resource allocation efficiency.

General conclusion

This study demonstrates the effectiveness of LSTM models in accurately predicting vehicle trajectories, providing valuable insights for optimizing urban transportation systems. By leveraging advanced programming tools and machine learning algorithms, the research successfully mapped and clustered vehicle positions, facilitating more informed decisions in traffic management and infrastructure development. The predictive capabilities of the LSTM model are particularly significant for the strategic placement of Mobile Edge Computing (MEC) units, which can be hosted on or near 5G antennas to enhance data processing speeds and reduce latency. This integration not only optimizes network performance but also supports the development of smarter, more responsive urban environments. The results underscore the potential of LSTM in enhancing real-time data processing and predictive analytics in smart city applications, paving the way for more efficient and intelligent urban planning.

Bibliography

[1] Dhenakaran, S. S., & Parvathavarthini, A. (2013). An overview of routing protocols in mobile ad-hoc network. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(2).

[2] Dipobagio, M. (2009). An overview on ad hoc networks. *Institute of Computer Science (ICS), Freie Universität Berlin*.

[3]Sood, M., & Kanwar, S. (2014, April). Clustering in MANET and VANET: A survey. In 2014 international conference on circuits, systems, communication and information technology applications (CSCITA) (pp. 375-380). IEEE.

[4] Zaidi, T., & Faisal, S. (2018, December). An overview: Various attacks in VANET. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp. 1-6). IEEE.

[5] Akhtar, N., Ozkasap, O., & Ergen, S. C. (2013, April). VANET topology characteristics under realistic mobility and channel models. In 2013 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1774-1779). IEEE.

[6] Kumar, A., & Bansal, M. (2017, September). A review on VANET security attacks and their countermeasure. In *2017 4th international conference on signal processing, computing and control (ISPCC)* (pp. 580-585). IEEE.

[7] Lu, S. P., Li, K. M., & Hsieh, C. Y. (2010, December). The study of dynamic navigation with WAVE/DSRC in VANETs. In *2010 International Computer Symposium (ICS2010)* (pp. 524-528). IEEE.

[8] Ahamed, A., & Vakilzadian, H. (2018, May). Issues and challenges in VANET routing protocols. In 2018 IEEE international conference on electro/information technology (EIT) (pp. 0723-0728). IEEE.

[9] Zhou, B., Tiwari, A., Zhu, K., Lu, Y., Gerla, M., Ganguli, A., ... & Krzysiak, D. (2009, October). Geo-based inter-domain routing (GIDR) protocol for MANETs. In MILCOM 2009-2009 IEEE Military Communications Conference (pp. 1-7). IEEE.

[10] Nagaraj, U., & Dhamal, P. (2012). Broadcasting routing protocols in VANET. Network and Complex Systems, 1(2), 13-19.

[11] Mishra, S., Raj, A., Kayal, A., Choudhary, V., Verma, P., & Biswal, L. (2012). Study of cluster based routing protocols in wireless sensor networks. International journal of scientific and engineering research, 3(7), 6.

[12] Allal, S., & Boudjit, S. (2012, July). Geocast routing protocols for vanets: Survey and guidelines. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (pp. 323-328). IEEE.

[13] Salman, O., Morcel, R., Al Zoubi, O., Elhajj, I., Kayssi, A., & Chehab, A. (2016, November). Analysis of topology based routing protocols for VANETs in different environments. In 2016 IEEE international multidisciplinary conference on engineering technology (IMCET) (pp. 27-31). IEEE.

[14] Kolte, S. R., & Madankar, M. S. (2014, April). Adaptive congestion control for transmission of safety messages in VANET. In International Conference for Convergence for Technology-2014 (pp. 1-5). IEEE.

[15] Nitesh, G. S., & Kakkar, A. (2016). Generations of mobile communication. *International Journal of Advanced Research in Computer Science and Software Engineering*, *32*.

[16]Beraa, R. N., Silb, S., & Alfred, Q. M. (2022). International Conference on Machine Learning and Data Engineering Achieving Low Latency and High Reliability in 5G and 6G Communication System. NeuroQuantology, 20(9), 4215.

[17]Salih, A. A., Zeebaree, S. R., Abdulraheem, A. S., Zebari, R. R., Sadeeq, M. A., & Ahmed, O. M. (2020). Evolution of mobile wireless communication to 5G revolution. Technology Reports of Kansai University, 62(5), 2139-2151.

[18]Junejo, M. H., Ab Rahman, A. A. H., Shaikh, R. A., & Yusof, K. M. (2021). Location closeness model for VANETs with integration of 5G. Procedia Computer Science, 182, 71-79.

[19] Sunyaev, A., & Sunyaev, A. (2020). Cloud computing. Internet computing: Principles of distributed systems and emerging internet-based technologies, 195-236.

[20] Ali, B., Gregory, M. A., & Li, S. (2021). Multi-access edge computing architecture, data security and privacy: A review. IEEE Access, 9, 18706-18721.

[21] Zhang, Y., Wu, Y., Moustafa, H., Tsang, D. H., Leon-Garcia, A., & Javaid, U. (2018). Multi-access mobile edge computing for heterogeneous IoT. IEEE Communications Magazine, 56(8), 12-13.

[22] Sriram, G. S. (2022). Edge computing vs. Cloud computing: an overview of big data challenges and opportunities for large enterprises. International Research Journal of Modernization in Engineering Technology and Science, 4(1), 1331-1337.

[23] Zhou, S., Netalkar, P. P., Chang, Y., Xu, Y., & Chao, J. (2018, August). The MEC-based architecture design for low-latency and fast hand-off vehicular networking. In 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall) (pp. 1-7). IEEE.

[24] Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. IEEE Communications Surveys & Tutorials, 20(4), 2961-2991.

[25] Filali, A., Abouaomar, A., Cherkaoui, S., Kobbane, A., & Guizani, M. (2020). Multi-access edge computing: A survey. IEEE Access, 8, 197017-197046.

[26] Patra, B. (2023, May). 100x increase in industrial and personal productivity augmenting the state-of-the-art technologies ai/ml, edge computing, and 5g network. In Emerging Electronic Devices, Circuits and Systems: Select Proceedings of EEDCS Workshop Held in Conjunction with ISDCS 2022 (pp. 463-472). Singapore: Springer Nature Singapore.

[27] Dwivedi, Y. K., Pandey, N., Currie, W., & Micu, A. (2024). Leveraging ChatGPT and other generative artificial intelligence (AI)-based applications in the hospitality and tourism industry: practices, challenges and research agenda. *International Journal of Contemporary Hospitality Management*, *36*(1), 1-12.

[28] Paul, T., & Bommu, R. (2024). Strategic Employee Performance Analysis in the USA: Leveraging Intelligent Machine Learning Algorithms. *Unique Endeavor in Business & Social Sciences*, *3*(1), 113-124.

[29] Talukdar, W., & Biswas, A. (2024). Synergizing Unsupervised and Supervised Learning: A Hybrid Approach for Accurate Natural Language Task Modeling. *arXiv preprint arXiv:2406.01096*.

[30] Igual, L., & Seguí, S. (2024). Supervised learning. In *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications* (pp. 67-97). Cham: Springer International Publishing.

[31] Qureshi, M. N., & Lazam, N. A. M. (2024). Optimisation and cross-validation of an e-hailing hybrid pricing algorithm using a supervised learning classification and regression tree model: a heuristic approach. *Journal of King Saud University-Science*, *36*(3), 103107.

[32] Saxena, A., Agarwal, A., Pandey, B. K., & Pandey, D. (2024). Examination of the Criticality of Customer Segmentation Using Unsupervised Learning Methods. *Circular Economy and Sustainability*, 1-14.

[33] Saxena, A., Agarwal, A., Pandey, B. K., & Pandey, D. (2024). Examination of the Criticality of Customer Segmentation Using Unsupervised Learning Methods. *Circular Economy and Sustainability*, 1-14.

[34] Xu, J., Xu, K., Wang, Y., Shen, Q., & Li, R. (2024). A K-means Algorithm for Financial Market Risk Forecasting. *arXiv preprint arXiv:2405.13076*.

[35] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178-210.

[36] Marutho, D., Handaka, S. H., & Wijaya, E. (2018, September). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In *2018 international seminar on application for technology of information and communication* (pp. 533-538). IEEE.

[37] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C., Silverman, R., & Wu, A. Y. (2000, May). The analysis of a simple k-means clustering algorithm. In

Proceedings of the sixteenth annual symposium on Computational geometry (pp. 100-109).

[38] Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*, *9*(1), 15-24.

[39] Tirthyani, D., Kumar, S., & Vats, S. (2024). Clustering and Unsupervised Learning. In *Recent Trends and Future Direction for Data Analytics* (pp. 119-139). IGI Global.

[40] Ernst, D., & Louette, A. (2024). Introduction to reinforcement learning.

[41] Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, *10*(07), 3897-3904.

[42] Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, *22*(5), 717-727.

[43] Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, *2*(1), 189-194.

[44] Guresen, E., & Kayakutlu, G. (2011). Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, *3*, 426-433.

[45] Sazlı, M. H. (2006). A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, *50*(01).

[46] Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, *39*(1), 43-62.

[47] Apicella, A., Donnarumma, F., Isgrò, F., & Prevete, R. (2021). A survey on modern trainable activation functions. *Neural Networks*, *138*, 14-32.

[48] Sharkawy, A. N. (2020). Principle of neural network and its main types. *Journal of Advances in Applied & Computational Mathematics*, *7*, 8-19.

[49] Tyagi, A. K., & Abraham, A. (Eds.). (2022). Recurrent neural networks: Concepts and applications.

[50] Katte, T. (2018). Recurrent neural network and its various architecture types. *International Journal of Research and Scientific Innovation*, *5*(3), 124-129.

[51] Liu, S., Mocanu, D. C., Pei, Y., & Pechenizkiy, M. (2021, July). Selfish sparse rnn training. In *International Conference on Machine Learning* (pp. 6893-6904).

PMLR.

[52] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107-116.

[53] Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020, June). Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. In *International conference on artificial intelligence and statistics* (pp. 2370-2380). PMLR.

[54] Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37-45.

[55] Li, Z., He, D., Tian, F., Chen, W., Qin, T., Wang, L., & Liu, T. (2018, July). Towards binary-valued gates for robust lstm training. In *International Conference on Machine Learning* (pp. 2995-3004). PMLR.

[56] Van Der Westhuizen, J., & Lasenby, J. (2018). The unreasonable effectiveness of the forget gate. *arXiv preprint arXiv:1804.04849*.

[57] Yao, K., Cohn, T., Vylomova, K., Duh, K., & Dyer, C. (2015). Depth-gated LSTM. *arXiv preprint arXiv:1508.03790*.

[58] Saabith, A. S., Fareez, M. M. M., & Vinothraj, T. (2019). Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development*, *6*(10).

[59] Estrada, J. S., Cruz, N. C., Lupión, M., Garzón, E. M., & Ortigosa, P. M. (2023). TEAMWORK USING COLAB NOTEBOOKS IN THE CLOUD. In *EDULEARN23 Proceedings* (pp. 2710-2718). IATED.

[60] Chollet, F. (2015). Keras open-Source Neural-Network Library Written in Python.

[61] Singh, P., & Manure, A. (2019). *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python*. Apress.

[62] Wilde, E., & Wilde, E. (1999). Hypertext Markup Language (HTML). *Wilde's WWW: Technical Foundations of the World Wide Web*, 173-262.

[63] Meyer, E. A. (2004). *Cascading style sheets: The definitive guide*. " O'Reilly Media, Inc.".

[64] Frisbie, M. (2019). *Professional JavaScript for Web Developers*. John Wiley & Sons.

[65] Singh, P., Manure, A., Singh, P., & Manure, A. (2020). Introduction to tensorflow 2.0. *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python*, 1-24.

[66] Tosi, S. (2009). *Matplotlib for Python developers*. Packt Publishing Ltd.

[67] Heydt, M. (2017). *Learning pandas*. Packt Publishing Ltd.

[68] Gupta, P., & Bagchi, A. (2024). Introduction to NumPy. In *Essentials of Python for Artificial Intelligence and Machine Learning* (pp. 127-159). Cham: Springer Nature Switzerland.

[69] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.