Cognitive Virtual Ad Hoc Mobile Cloud-Based Networking Architecture

By

Laith Adeeb Taha Al Saeed

Supervisor**: Professor Hamed Al-Raweshidy**

Professor Maozhen Li

Department of Electronic and Electrical Engineering
College of Engineering, Design and Physical Science
Brunel University London
United Kingdom

This thesis is submitted for the degree of
Doctor of philosophy (PhD) in Telecommunications engineering

March 2021

Abstract

This thesis proposed cognitive techniques and intelligent algorithms that offered adaptive and advanced facilities to cloud-based networking by using Virtual Ad Hoc Mobile Cloud Computing Networks architecture (VAMCCNs). This is presented as a working case to address their global network challenges and to add cognitive support to the network design and implementation for better meeting traffic management and application requirements in mission objectives. The thesis concentrates on three main contributions.

Firstly, an adaptive model, namely: a Heterogeneous Mobile Cloud Computing Network (HMCCN), was proposed to integrate different cloud networks architectures into one workflow. The cognitive data offloading task and the routing decision methods were applied using two different approaches: Fuzzy Analytic Hierarchy system (FAH) as a first approach and cognitive Software Defined Network (SDN) model as a second centralised approach. Experimental results show improvement in network reliability and throughputs, minimised in both nodes' energy consumption and network latency with efficient intelligent data load balance and network resources allocation with best cloud model selection.

Secondly, based on a virtual Ad Hoc cloud network with a realistic Random Waypoint Motion (RWM) model, an innovative cognitive routing algorithm was presented to improve efficient and reliable route selection among multiple possible routes. Routing protocols based on conventional, Fuzzy logic used important parameters with two data collections and decisions techniques and a new adaptive Intelligent Hybrid Fuzzy-Neural routing protocol (IHFN) that included prior knowledge to the network of the underlying motion and energy parameters were all proposed and compared. Results with the new hybrid algorithm shown a significant improvement to solve the network end-to-end performance degradation problem. The new hybrid protocol improved network throughput with an average of 20% higher than traditional Ad Hoc On-Demand Distance Vector (AODV) Routing protocol, improved the usage of network resources and reduced the maintenance process in adynamic topologies network.

Finally, based on datasets collected from a realistic motion RWM model in a virtual Ad Hoc cloud network, the performance behaviour of six selected deep learning algorithms to predict the next steps of positions, speed and residual battery energy values of these mobile nodes have been evaluated and compared. This work goes further by presenting two algorithm's training techniques to predict the next 300-time steps of position, speed, and energy. Results and dissuasion show the differences concerning prediction accuracy between using the single node dataset model or Multiple node's dataset model.

IN MEMORY OF MY AUNTY

TO MY

FATHER, MOTHER, WIFE,

SONS AND DAUGHTER

WITH LOVE AND ETERNAL

APPRECIATION

Declaration

This is to certify that:

I. **This thesis is my original work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed, and reviewed by myself and my supervisors in the Department of Electronic and Electrical Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, UK.**

II. **All the information and material used and derived from other works have been acknowledged and referenced.**

By

Laith Adeeb Taha Al Saeed

March 2021

Acknowledgments

I am grateful to the creator almighty Allah for good health and wellbeing; without his blessing and mercy, this thesis would not have been possible.

I am heartily thankful to my supervisors Professor Hamed Al-Raweshidy and Professor Maozhen Li. The motivation, encouragement, guidance, immense knowledge and support they provided enabled me to develop this work. I am thankful to my colleagues at the WNCC and friends at Brunel University.

I want to express my deepest gratitude to my parents, wife, sons, daughter, and sisters; without their tremendous support, help and prayers, I would not have made it through my PhD degree.

Finally, I would like to show my gratitude to all the staff at Brunel University and those who helped me and supported me in finalising my thesis.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACO	**A**nt **C**olony **O**ptimisation
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
AODV	**A**d Hoc **O**n-Demand **D**istance **V**ector
API	**A**pplication **P**rogram **I**nterface
AR	**A**ugmented **R**eality
CNN-LSTM	**C**onventional **N**eural **N**etwork- **L**ong **S**hort-**T**erm **M**emory
CSV	**C**omma **S**eparated **V**alues
CWC	**C**ognitive **W**ireless **C**loud
DCNs	**D**ata **C**entre **N**etwork**s**
DCNs	**D**ata **C**entre **N**etworks
DL	**D**eep **L**earning
DSR	**D**ynamic **S**ource **R**outing
ELM	**E**xtreme **L**earning **M**achine
ELS	**E**nhanced **L**ocalisation **S**olution
ETSI	**E**uropean **T**elecommunications **S**tandards **I**nstitute
FAH	**F**uzzy **A**nalytic **H**ierarchy
FDD	**F**requency **D**ivision **D**uplexing
FLMC	**F**uzzy **L**ogic **M**obility **C**ontroller
GA	**G**enetic **Al**gorithms
GPS	**G**lobal **P**ositioning **S**ystem
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing **P**rotocol
GRU	**G**ated **R**ecurrent **U**nit
HMCCNs	**H**eterogenous **M**obile **C**loud **C**omputing **N**etwork**s**
IHFN	**I**ntelligent **H**ybrid **F**uzzy -**N**eural
IoT	**I**nternet **o**f **T**hings
LSTM	**L**ong **S**hort-**T**erm **M**emory
LTE	**L**ong-**T**erm **E**volution
MACNs	**M**obile **A**d Hoc **C**loud **N**etworks
MANET	**M**obile **A**d Hoc **N**etwork

MCC	**M**obile **C**loud **C**omputing
MCCNs	**M**obile **C**loud **C**omputing **N**etworks
MEC	**M**obile **E**dge **C**omputing
MFs	**M**embership **F**unction**s**
MLP	**M**ulti-**l**ayer **P**erceptron
MN	**M**obile **N**ode
MS	**M**obile **S**tations
NN	**N**eural **N**etwork
OLSR	**O**ptimized **L**ink **S**tate **R**outing
PDR	**P**acket **D**elivery **R**atio
PSO	**P**article **S**warm **O**ptimization
PUs	**P**rimary **U**sers
RAN	**R**adio **A**ccess **N**etwork
RERR	**R**oute **E**rror
RL	**R**eliability **V**alue
RMSE	**R**oot **M**ean **S**quare **E**rror
RNC	**R**adio **N**etwork **C**ontroller
RNN	**R**ecurrent **N**eural **N**etwork
RREP	**R**oute **R**eply
RREQ	**R**oute **R**equest
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
RTT	**R**ound-**T**rip **T**ime
RWM	**R**andom **W**aypoint **M**obility
SUs	**S**econdary **U**ser**s**
TDD	**T**ime **D**ivision **D**uplexing
VAMCCNs	**V**irtual **A**d Hoc **M**obile Cloud **C**omputing **N**etworks
VANETs	**V**ehicular **A**d Hoc **N**etwork**s**
VR	**V**irtual **R**eality
QoS	**Q**uality **o**f **S**ervice

Chapter 1

INTRODUCTION

1.1 Background

With the Internet of things (IoT) becoming part of human's daily lives and environments, rapid growth has expected in the number of connected devices [1]. Internet of things is expected to connect billions of devices and users to bring promising advantages to society.

With this growth and improvements, fog computing with its related edge computing paradigms was presented as a promising solution for handling the large volume of time-sensitive and security-critical data that cloud-based users and devices produce.

Moreover, over the decade, wireless technologies have increased dramatically, and many mobile devices can now support multiple forms of wireless connectivity. The popularity of cloud services and wireless technology, including wireless devices, has been the motivation to researching and enhancing Mobile Cloud Computing Networks (MCCNs) and Ad Hoc Mobile Cloud Computing Networks (AMCCNs) architectures in this thesis.

Therefore, massive data expected to be traversed cloud network backbones and passes through its nodes and links. Emerging location-aware and time-sensitive applications such as self-driving cars, drones, and patient monitoring through cloud-based networks is a big challenge. Because the network cannot guarantee the fulfilment of the ultra-low latency requirements of these applications, providing accurate location-aware services or being able to scale to the volume of the data that these applications produced [2].

Traditionally cloud-based networks, in general, faces all the challenges related to wireless data communications. Traditional data communication networks are working in reactive approaches, which means that they address network problems after the problem has occurred. New types of cloud architectures, services, and applications drive the researchers to think towards new and innovative communication network designs and new routing methods and techniques.

1.2 Motivation

In recent years, researchers have clearly shown an increased interest in enhancing network performance. The continuous developments of cloud-based applications and mobile devices' ability show that the networking requirements have become significantly much more sophisticated. Today, data traffic is a lot more demanding with complex needs, especially in an age where cloud computing, big data, and IoT reigns supreme. With these complicated patterns, it is too fast progress for traditional networking to handle.

Networks in the mobile cloud must respond and adjust dynamically based on network condition and application needs. Current network management models and routing solutions on mobile networks are only valid under specific assumptions and conditions. Intensive and time-sensitive applications performance have been constrained due to uncertainties associated with network resources utilisation and link status. Virtual Ad Hoc Mobile Cloud Computing Networks (VAMCCNs) can be considered as self-organised, infrastructure-less, multi-hop wireless networks. They are able to be linked to cellular networks and can form a group of devices. Mobile cloud networks in general, require a reliable connection between the cloud and the mobile devices. In VAMCCNs characteristics like node mobility, dynamic topology, and intermittent links make the connections between nodes in some cases unreliable.

As a result, cloud networking needs to be self-aware in order to be able to provide resilient services and applications. Such networks should present cognitive properties where the actions and decisions are based on analysis, automatic operations, adaptive functionality, and self-manageability. The introduction of cognition towards cloud networking can address the complexity of various applications and services provided and its heterogeneous networks by specifying and delivering solutions for the efficient handling of these concepts. For that reason, the intelligence that is needed for the management of cloud networking functionalities is rated to the optimal cloud networks configurations given a specific adaptable solution. The input in such situations consists of information on mobility levels, network resources, data traffic, and interface levels. These challenges motivate the proposed idea of extending cognitive functionalities to the legacy mobile cloud network.

The work in this thesis focused, motivated and taken into consideration the following points:

- The possibility of applying the methodologies found in cognitive networks to cloud networking and particularly to the MCCNs.
- The possibility of using this approach to increase cloud networking's performance to meet its mission objective.

- To find the efficient management of increasing complexity within cloud networking requirements.

1.3 Research Aim and Objectives

The essence of this thesis goal is to present a framework that will combine some of the cognitive network approaches into a cloud networks environment based on VAMCCNs as casework. The aim is addressed by the following:

- To allow to use the principle of cognitive integrations of multiple cloud networking architectures based on application requirements, networks resources and network' state.
- To maintain service availability in cloud-based networks, particularly in MCCNs that are operated over a set of sporadically available nodes and links.
- To better utilise network available resources when establishing route paths, offloading data, and selecting the best cloud architecture based on task requirements to make the network more resilient to changes in its topology and congestion during operations.
- To allow VAMCCNs that operate over unreliable infrastructures to work in a reliable method to execute complex tasks and applications efficiently.
- To allow VAMCCNs to overcome congestion, link breakage and network resources consumptions.
- To allow VAMCCNs to be aware of their state and environment.
- To identify fault patterns to predict future network performance degradation.
- To understand algorithms performance differences in pattern prediction models.
- To understand the best technique to use mobility and energy datasets for training and testing algorithms in pattern prediction methods to get accurate results.

The above aims were achieved by meeting the following objectives of this thesis:

- To present an adaptive Heterogeneous Mobile Cloud Computing Networks model (HMCCNs) using VAMCCNs as an assumption approach.
- To present a cognitive offloading and routing mechanism using The Fuzzy Analytic Hierarchy (FAH) technique for offloading and best optimal cloud model selection in HMCCNs.
- To present a centralised cognitive Software Defined Network (SDN) controller as a second approach for routing and data traffic management in HMCCNs.

- To collect network datasets of useful information such as (position, speed, and residual battery energy) values. The datasets were created based on a simulated mobile Ad Hoc network with realistic movement using the Random Waypoint Mobility model (RWM) and applied three data traffic rates (Low, Medium, and High). The datasets collected will be used for further analysis.
- To simulate and compare the network performance of various conventional and adaptive proposed routing algorithms. An Ad Hoc On-Demand Distance Vector (AODV) as a traditional routing protocol, a fuzzy logic routing algorithm using essential parameters for selecting the reliable routes and with two routing decision and parameters collection techniques. Finally, an Intelligent Hybrid Fuzzy-Neural (IHFN) routing algorithm.
- To improve the reliable path selected by adding prior knowledge of the underlying mobility, battery capacity and speed patterns for the network's nodes.
- To understand and identify the accuracy results of using six selected deep learning models to predict next positioning, speed, and energy based on datasets collected from a virtual Ad Hoc mobile network with realistic RWM model using three different data traffics rates.
- To find better training and testing technique using either Single or Multiple nodes datasets models that will give higher prediction accuracy to predict the next 300-time steps of positioning, speed and energy of the mobile nodes based on a dataset collected from a virtual Ad Hoc mobile network with realistic RWM model using three different data traffics rates by applying six chosen deep learning algorithms (RNN, GRU, Bi-Directional-LSTM, LSTM, CNN-LSTM and Stacked LSTM).

1.4 Contributions

There are three main contributions of this thesis which is summarised as follows:

1- Proposed an adaptive HMCCN model that integrates various cloud architectures, models, and service types into one workflow. Results of the integrated MCCN and VAMCCN show improvements in network throughputs and minimising nodes' power consumptions. Furthermore, two approaches are presented as two cognitive models used to support HMCCN in managing optimal cloud (s) network architecture selection, data flow, and routing. The first approach proposed, based on the FAH model is a cognitive model supported HMCCN, which selects the optimal cloud network architecture (s) based on advance essential criteria. The second approach based on the concept of cognitive SDN controller that is presented as an intelligent centralised method supported HMCCN in terms of routing, data flow balanced and cloud network architecture(s) selections. A comparison between the SDN network and traditional network in terms of its response time performance using Round-Trip Time (RTT) parameter was applied. Results of the measurements' RTT values taken from different scenarios in both networks show a noticeable minimisation in network latency with SDN network compared to traditional network and that differences increased proportionally with the increasing of the number of switches used within the network.

2- Based on the design and the simulation of a realistic RWM model in a virtual Ad Hoc mobile cloud network with three data transmission rates applied (Low, Medium, and High), an innovative routing algorithm was proposed aimed at improvement efficient and reliable route among multiple possible routes. Four experiments proposed with various routing algorithms, AODV as a conventional routing protocol, Fuzzy logic routing protocol with two experiments applied, measured the best routing decision process location and best network parameters collection time. Finally, as a fourth experiment, an adaptive IHFN routing protocol has been proposed. This new routing protocol was designed based on a fuzzy logic system that offered a natural way of reasoning and representing the problems and a neural network system that added an intelligent future awareness capability of movement and resources usage patterns with networks state information. Results of the network performance concerning the four proposed protocols were evaluated and compared. Final results of the proposed IHFN routing protocol have shown improvements in network throughput with an average value of 20% higher than the AODV routing protocol, which has also included improvements in packet delivery ratio. Results are also affected by the time step used for predictions in IHFN. Furthermore, an average of 10% throughput improved when the source node is used as the primary location for routing management and Route Replay (RREP) packet for network parameters collection compared to the other scenario.

3- The performance behaviour of six selected deep learning algorithms (RNN, GRU, Bi-Directional-LSTM, LSTM, CNN-LSTM and Stacked LSTM) is used to predict the next position, speed, and residual battery energy value of mobile nodes have been evaluated and compared. This is based on the datasets collected from the mobile Ad Hoc network simulated with a realistic random waypoint mobility model (RWM) with three data traffic rates scenarios applied. Final results analysed concerning pattern prediction accuracy using the complex information in the datasets have shown high accuracy prediction results using Stack-LSTM, RNN, and LSTM algorithms compared to other algorithms used.

Furthermore, a second experiment is presented to compare two learning and testing datasets techniques by using a single node dataset model or multiple nodes datasets model based on the same six selected deep learning algorithms to predict the next 300-time steps. To get actual and correct results, five nodes were randomly selected and their datasets used for this experiment. Results have been analysed and shown that Stacked LSTM, RNN, and LSTM algorithms gave lower Root Mean Square Error (RMSE) values with the next 300-time steps prediction for x, y, speed, and energy when trained and tested with the multiple nodes dataset model. With Bi-Directional- LSTM algorithm, the RMSE values were lower when it was trained and tested with a single node model. In contrast, the prediction results of other algorithms have shown instability when those complex datasets were used.

1.5 Thesis Outline

The whole work in this thesis is organised into six chapters. Each chapter started with a brief introduction providing an overview and focusing into the main contributions of the chapter. At the end of each chapter a summary is presented.

Chapter 2 Starts with explaining the evolution of cloud computing networks and cloud networking with a brief overview of cloud architectures like MCC, fog, edge, and IoT. Cognitive techniques such as Artificial Intelligence (AI), SDN, and fuzzy logic and their applications in cloud computing networks were discussed. Also, Pattern prediction models with mobility methods and techniques using time series forecasting were fundamentally explained with some examples presented when applied in mobility prediction models.

Chapter 3 Presents a new model, namely HMCCNs. The suggested new cognitive model optimises the utilisation of heterogeneous computing and network resources in cloud networking environments in general and with MCCNs in particular. The proposed model shows an improvement in network throughput of the MCCN when integrated with VAMCCN during congestion and link breakdown conditions. It also improved nodes' energy consumptions during the tasks. Furthermore, two intelligent approaches proposed supported the HMCCN model in terms of optimal cloud architecture(s) selection, data offloading, data management, and routing. The first approach suggested using the FAH system. The second approach suggested using a cognitive SDN controller. The final results of two experiments with multiple scenarios of traditional and SDN based networks have been compared, evaluated and shows a significant improvement in network latency when using an SDN network compared to a traditional network. It is proportional to the scalability related to the increasing number of switches used within the network.

Chapter 4 Presents the design and implementation of an adaptive new IHFN routing protocol applied on Ad Hoc mobile cloud network with realistic RWM model assigned with various energy capacity and used three data traffic rates (low, medium, and high). Four experiments applied using AODV as a conventional routing protocol, Fuzzy logic routing protocol with two experiments were applied, which measured the best routing decision process location and best network parameters. Finally, as a fourth experiment, an adaptive IHFN routing protocol has been proposed. Final results of the network performance concerning the four proposed protocols were evaluated, compared, and shows a significant improvement with the new proposed IHFN routing protocol in network throughput and Packet Delivery Ratio (PDR). Results were also affected by the time step used for predictions in IHFN. Furthermore, there was an average of 10% throughput improvement when the source node was used as the primary location for routing management and RREP for network parameters collection compared to the other experiments that used destination node and Route Request Packet (RREQ).

Chapter 5 Presents the comparison performance of six machine learning algorithms (RNN, GRU, Bi-Directional-LSTM, LSTM, CNN-LSTM, and Stacked-LSTM) that were used to predict the next time steps of nodes positioning, speed and energy using a complex dataset collected from simulation of mobile Ad Hoc network with RWM model with three data rates applied (Low 200 packets/sec, Medium 600 packets/sec, and High 1200 packets/sec). Results showed differences in performance using complex datasets. Furthermore, another experiment presented with two scenarios applied by creating two different training and testing datasets models, namely Single node ‘dataset and Multiple nodes' datasets. The same six selected algorithms have been used to predict the next 300-times steps. To get actual and correct results, five nodes were randomly selected and their datasets used for this experiment. Results were analysed and compared for both scenarios and show the differences in algorithm’s performances when applied both datasets’ models on each one of these algorithms.

Finally, **Chapter 6** presents the study recommendations and conclusions and the suggestion for future work.

Chapter 2

BACKGROUND AND LITRATURE REVIEW

2.1 Introduction

In general, Cloud computing networks allows users to acquire on-demand computational and network resources on a pay-per-use basis and allows vertical and horizontal scalability [3]. The area of cloud computing networks technologies is one of the rapid development areas, with its related applications and services provided that is having a huge and almost direct impact on various aspects of the modern civilisation, including but not limited to military, commerce, education, gaming, and healthcare [4-10]. In this regard, the improvement of a reliable and flexible infrastructure that is secure, real-time, and cost-effective when delivering the data almost anywhere and anytime is very important to provide the best Quality of Service (QoS) required. With complexity increases in current cloud networks, services, and applications, upgrading the network and continuously improving its capabilities and limiting human intervention have become the most important demand. To do that, the networking research community presented a new model for a traditional network called the cognitive network that can think, learn, and decide [11-13].

2.2 Evolution of Cloud Computing Networks

2.2.1 Cloud Computing and Cloud Networking

Although cloud computing concepts were introduced back in the 1950s, the first cloud computing services became available in the early 2000s [14]. It was particularly aimed at large enterprises, then it spread to small and medium businesses and recently to consumers. Nowadays, individual users of mobile devices and PCs are relying on using cloud computing services to sync devices, back up and share data using personal cloud computing. Cloud computing can be defined as the delivery of different services across the Internet. These services include resources and applications such as servers, data storage, databases, software, and networking. In general, the cloud computing model is based on data centres that can manage the processing and storage of the massive scale of data. The connections between data centres are often made over optical networks to form one singular resource called data centre networks (DCNs). Despite that this type of communication will provide low latency between data centres itself, but the communication between DCNs and the end devices still prove to be a bottleneck. The cloud networking concept refers to the network and network management functionality that must be available to enable cloud computing. A good example of cloud networking is the provisioning of high reliability and high performance networking between the cloud provider and the user, which also will include the traffic that passes between them. Traditional cloud networking architectures present imposing challenges for an efficient and effective flow of data traffic through networks. It will be helpful in this regard to consider cognitive cloud networks as an intelligent solution to overcome network challenges.

2.2.2 Mobile Cloud Computing Networks

The importance of mobile cloud computing networks is extremely growing with the increase of cloud applications and services. Day by day, mobile cloud computing-based applications and services are gaining a reputation in different fields such as in military, learning, commerce, and health monitoring [4-10]. Mobile Cloud Computing (MCC) is an integration of mobile device, wireless network, and cloud computing. This concept allows mobile users to access unlimited storage space and computing power inside cloud servers using the internet [15]. Researchers generally use a computational augmentation method to enhance mobile devices' functionality by leveraging computational resources from mobiles to clouds through offloading methods [16]. Offloading computational tasks or intensive applications from the mobile device to the remote cloud can overcome the problems of limited processing power and the limited battery lifetime of mobile devices. Offloading will save mobile device power if heavy computational and light communication are considered. Figure 2-1 illustrates the area of whether to decide to offload the data for computation process or not. From this figure, the data should be offloaded if the amount of computation is high. Partial or full data offload are related to whether the area of computation is moderate or high respectively. The expansion increases of both mobile devices' abilities and the concept of MCC together have raised a potential concept called Mobile Ad Hoc Cloud Networks (MACNs). The main target for this evolution is to provide better availability and quality of the service. This attractive alternative framework presented to transform the non-dedicated and locally available resources capacity from end users' mobile devices into an overlay cloud platform.

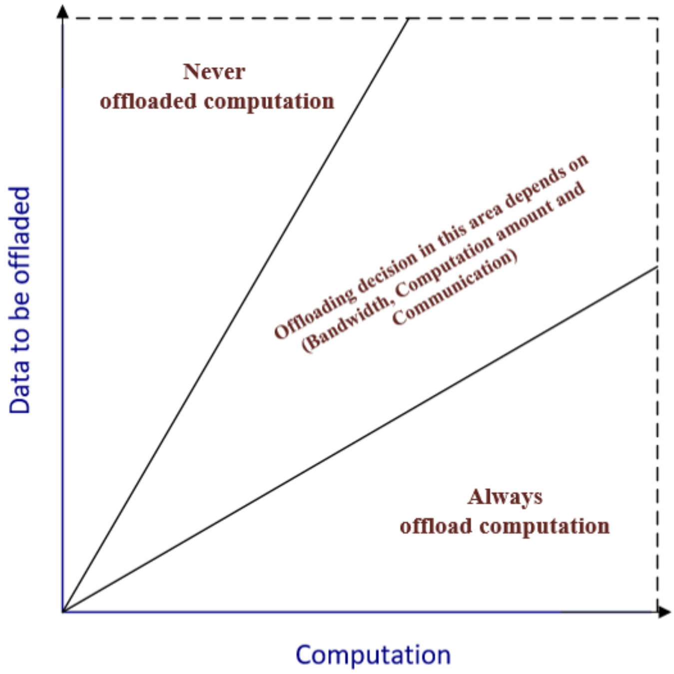

Figure 2-1 Offloading decision with respect to computation and data amount [17].

2.2.3 **Fog Computing**

Despite considering cloud computing as a mature technology that is providing elastic infrastructure in a pay-per-use model, the main limitation of this technology is clearly shown with the connectivity between the cloud servers and the end-user devices [18], since such connectivity is established based on the internet which is mainly associated with noticeable transfer delay that is not suitable with latency-sensitive applications.

Fog computing has emerged to provide elastic resources near the end-user to overcome the unacceptable latency between cloud servers and end-users and overcome the lack of location awareness and mobility support [19]. This paradigm will extend storage, computing and network services provided by the cloud and make it possible to do the required computing or process near to end-user devices allowing not only working with latency-sensitive applications but also will bring rapid innovation, efficiency, and affordable scalability. Fog computing can be considered a layered model that can allow ubiquitous access to shared and scalable computing resources. This model assists in the deployment of latency aware and distributed services and applications. It consists of fog nodes (light servers) located between the end-user devices and the cloud. The Fog nodes (physical or virtual) support communication system and data management and are also context aware. The nodes can be arranged into clusters either vertically or horizontally. Fog computing models provide local computing resources to end-user devices and allow connectivity to public servers (cloud servers) when needed. It achieves the minimum time required for request or respond to or from working applications. Fog nodes are the core module of the fog computing part, which can be either physical modules like routers, switches, gateways or be virtual modules like virtual machines, virtual switches, cloudlets that are closely connected with the end-user devices or access networks so that computing resources will be able to be provided to these end devices. Fog node is aware of its logical location within its cluster and also its geographical distribution. Furthermore, fog nodes can operate in a centralised and decentralised way.

Fog computing characteristics can be summarised as:

1- Low latency with contextual location awareness: Each fog node will be aware of its logical location in the context of the whole system. It will also be aware of the latency communication cost with other nodes(s). Unlike the datacentre or centralised cloud service, fog nodes generate data and analysis it much faster.

2- Heterogeneity: Fog computing supports various types of cloud network capabilities.

3- Scalability: Fog computing supports data load changes, resource pooling and network condition changing.

4- Real-time interaction: Unlike application with patching processing, fog computing works with real-time interactions applications.

Fog nodes also can work independently by making local decisions at the same node or within the cluster level. The fog computing concept that Cisco has announced was extended to a new idea called edge computing that works in the cellular networks [20]. It is also clear that cloud computing takes the main part to build 5G system and also Tactile Internet system [21] that can be considered the next evolution of the (IoT) that enables human-to-machine and machine-to-machine interaction in real-time with ultra-low latency and high availability.

The tactile internet will merge multiple technologies in the networking level, data and contents will be transmitted over 5G network. At the same time, all intelligent processes will be close to the users at the edge (mobile edge computing), in the application level, Artificial Intelligence (AI), Virtual Reality (VR), robotics, and Augmented Reality (AR) will all take part on it.

2.2.4 Edge Computing

Edge computing is a technology that transfer cloud computing to be located in cellular network boundary such that it will be one hop from the end-user devices [22] [23][24][25]. This shifting will unload the core network because all computing process will be performed at the edge of the cellular network. It will also minimise network latency because it is one hop from the user's devices, which will increase the bandwidth and will be capable of introducing new services and applications [26]. In the 5G cellular system, the cloud moves to the edge of the mobile network to reduce round trip latency by using one or two communication hops away from cellular mobile devices. In general, 5G cellular system architecture can be considered a combination of mobile users, cloud unit, and core mobile network supported with the internet and integrated with the remote cloud. Based on European Telecommunications Standards Institute (ETSI) that is considered as the leading organisation concerned with edge computing [27], researchers are searching and proposing the best place of cloud unit to be located in 5G network where multiple scenarios can be used, such as connecting cloud servers to (eNB) - LTE macro base station or placing cloud unit either in 3G/4G – RNC (Radio Network Controller) or at the edge of the core network. Some researchers also introduced a small cloud units called Nabula [28] and micro cloud [29]. Figure 2-2 shows the layers of the cloud, Fog, and Edge architectures.

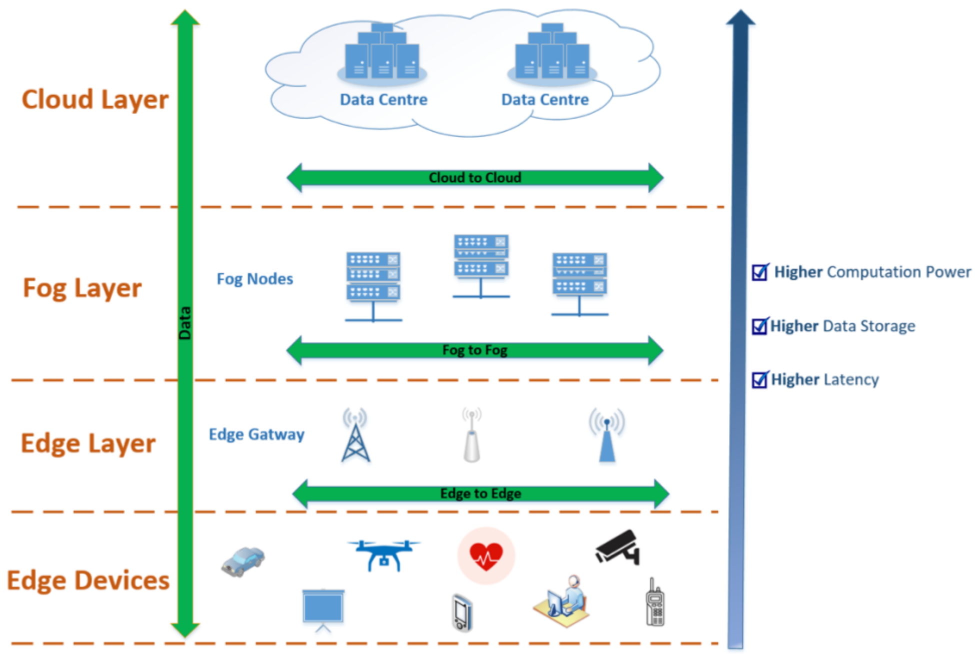

Figure 2-2 Cloud vs Fog vs Edge layers.

2.2.5 Internet of Things (IoT)

The internet of things refers to the expanding of the interconnection of smart devices. It is the latest development revolution of communication and computing that enables new communication forms between the people and things, and between things themselves. Through the cloud systems, the internet now supports the interconnections of huge numbers of personal and industrial objects that deliver sensor(s) information. IoT is mainly driven by deeply embedded devices. These devices are low bandwidth that communicates to each other and provides data through user interfaces. But with other high-resolution video security systems as an example, it requires high bandwidth streaming capabilities. In general, researchers focus on two IoT elements, the things that are connected and the networking that interconnects them. IoT consists of five main layers:

- The sensors and actuators: It represents the things. Sensors send back all observed information related to their environment such as humidity and temperature and all absence or presence of some observable process. While actuators work on their environment, such as operating a valve.
- Connectivity: Network connected to the devices through either wireless or wired links to send collected data to the datacentre or receive command data from the controller (actuator).
- Capacity: The working network supporting the devices should be able to handle huge flow of data.
- Storage: Through the cloud capability, it can provide the storage facility for storing and maintaining the collected data.
- Data analytics: Due to the huge information collected, data analytics capability is required to process the data flow.

2.3 Challenges of Cloud Networks and MCCNs

Many cloud applications require some guaranteed bandwidth between the end-user as a client and the cloud servers to comply with the necessary quality of service and be within the acceptable time frame. Insufficient bandwidths between the client-server’s model will impose significant latency on user tasks or interactions. The network topology of the cloud networks can be the main reason for the limitation of cloud services provided, so it should be tuned to match a predefined traffic requirement. In the MCC, the two different technologies namely cloud computing and mobile computing, have several issues and challenges [17].

- Bandwidth: The main concern for MCC is the low bandwidth. In general, wireless networks have much lower bandwidth compared to wired networks [30]. Also, mobile devices have a problem of high variation in network bandwidth.
- Energy resources: According to the hardware specifications, there is a limitation on the battery life of mobile devices. Power consumption should be taken into consideration when running any local computation or offloading tasks.
- Mobile resources: Mobile devices computational and storage resources are limited compared to other static elements [31]. Mobile computational resources such as memory size, disk capacity, and processor speed are limited due to weight limitations and power consumption.
- Data availability: Unreliability or failure in network connectivity is a significant risk when running applications in the cloud computing environment [32], particularly in unnatural circumstances like a disaster.
- Wireless communications: In wireless communication, the packet's path might be broken, or the packets might be dropped or get congested causing more obstacles in the network.
- Security and privacy: The process of keeping the data secure when using the cloud for computational and storage is essential.
- Fault tolerance: Faults occur in MCCNs as a result of the mobility of the device, running out of battery or network failure with the cloud.

2.3.1 Traditional Routing in Wireless Networks

In the routing protocol, user traffic is directed and transported through the network from the source node to the destination node. The main objectives are to consider maximising network performance from the application aspects (application requirements) and minimising the network's cost according to its capacity. The network application's main requirements are throughput, hop count, stability, jitter, delay, cost, etc. In contrast, the network capacity is related to the measurements of each node's resources, density (nodes number in the network), rate of occurrence End-to-End connection (number of radio access) and the occurrence of the topology changes (mobility rate) [33]. As a result, the routing concept is restricted by traffic requirements and network capacity, as illustrated in Figure 2-3.

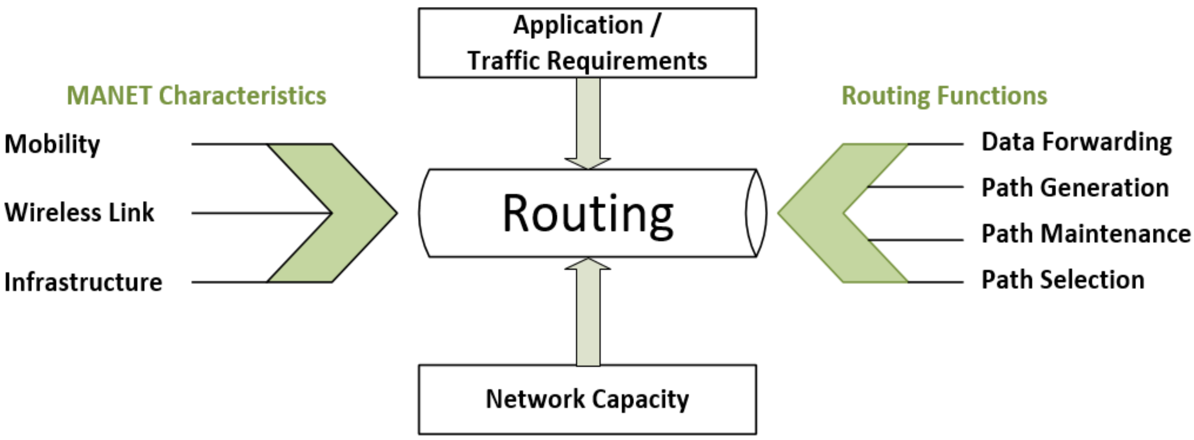

Figure 2-3 Routing restrictions [33].

From this figure, Data forwarding refers to forwarding user traffic across the selected route. Path generation indicates generating paths with respect to distribution and assembled state information of the application and the network. Path maintenance relates to maintaining the chosen route. Path selection refers to selecting appropriate paths based on application and network state. Routing in a mobile network is complex because of the bandwidth limitation, dynamic topology, and energy constraints. Each mobile node acts separately or corporately as a router. Sometimes routing is multi-hop such that packets are forwarded from the source node to the destination node via several intermediate nodes.

Routing protocols can be classified into different groups based on their properties [34]:

- Centralised versus Distributed.
- Static versus Dynamic.
- Proactive versus Reactive.

In centralised routing algorithms, the central node will be responsible for all routing choices. In general, centralised routing algorithms can be used in individual cases and for limited size network. In distributed routing algorithms, route computation in network nodes are shared routes among them, and they also exchanged information. The distributed model is currently used in most network systems. Routing protocols have another classification that relates to change the routes in response to traffic input patterns. In static algorithms, the routes used between source and destination pairs stay fixed no matter the traffic conditions. Changes happen only when there is link or node failure. This type of routing algorithm cannot accomplish high throughput under wide differences in traffic input patterns. In adaptive routing algorithms, the routes between source and destination pairs have some changes in response to congestion. The third type of classification that is more associated to Ad Hoc networks is to classify the routing algorithms into either proactive or reactive. In proactive protocols, the routes' evolution is continuing within the networks so that the route is already known in advance when any packet needs to be forwarded. The group of Distance Vector protocols is a good example of proactive protocols type. Reactive protocols, on the other hand, will initiate the route discovery procedure on-demand only. Thus, only when the route is needed, a sort of discovery procedure and search is employed. The group of classical flooding algorithms is related to reactive type.

2.4 Cognitive Routing in Cloud Networks

It is not always possible to expect the availability of fully functional network and computing infrastructure, assuming unrestricted power sources to handle the required data processing needs. In some exceptional circumstances like tornadoes, hurricanes, earthquakes and wildfires, edge infrastructure might be destroyed. Therefore, constrained mobile devices will be an option proposed to provide networking, computation and power resources that will be used for disaster incident decisions and responses. Usually, applications used in such circumstances like facial recognition techniques to help find a missing person or bad performance [35] need to be managed in real-time and with high streaming of images with fluctuating resolutions on a limited power edge network. In general, Cognitive networks [36] can be described as an intelligent, proactive technique that enables networks to provide a higher quality of service to the user and increase network performance and efficiency. Unlike traditional networks, a cognitive network is aware of usage pattern and network’s states. It can predict and enhance data exchange and routing based on users' QoS needs, data history, and the current state of the network. Methods like Artificial Intelligence and Game Theory are considered a good solution to predicting what will happen next inside the cognitive network. Cognitive Radio is also a method that can be applied to manage how to utilise radio frequencies in a better way for mobile communication cognitive networks.

2.5 Artificial Intelligence (AI)

Artificial intelligence was started in the early 1950s by the famous mathematician Alan Turing. AI is part of the area in computer science that deals with machine intelligence. Unlike computers, which solves problems based on instructions provided by humans in the form of coding (programs), Artificial intelligence makes computer learn itself to solve problems. The combination of cloud computing applications and artificial intelligence has been considering the most important driving force to many researchers. Advance technology like automated document classification self-driving vehicles and voice recognition would have been impossible without the collaboration of both cloud computing and AI technologies. The power of AI can be used for helping the cloud networking to look for patterns and insights in network's information to manage cloud networking routing and optimising the workflows inside it.

2.5.1 Machine Learning (ML)

Machine learning is the subset of artificial intelligence that allows computers to learn and improve automatically form the past experience without explicitly being programmed. ML was initially introduced in the late 1950s as a special technique of Artificial Intelligence [37].

In machine learning, there are three categories that can be classified, these are [31]:

- Supervised Learning: In this system, both input data and the output data are being provided and labelled for classification purposes providing learning features to additional data processing.

  Supervised learning can also be divided into two categories:

  (a) Regression: This algorithm predicts when the output data is a real value, for example when predicting the prices in the stock market.

  (b) Classification: In this algorithm, the output data is used as part of a specific category, for example when classifying the colour "BLACK" or "WHITE".

- Unsupervised Learning: In this system, only the input data is provided without the output labelled data.

  Unsupervised learning can also be divided into two categories:

  (a) Association: It works by learning the rules that are explaining a large portion of data. This rule is called association rule, for example when customers buy "Toys" they tend to buy "Batteries".

  (b) Clustering: It works by discovering the inherent relationships within groupings of data, for example, when customers being classified into groups based on the type of products they purchased.

- Reinforcement Learning: Actions are taken in this learning based on the situation so that the main goal is to increase the reward.

2.5.2 Artificial Neural Networks (ANNs)

ANNs have been trained to achieve complex functions in different fields, including identification, pattern recognition, speech, classification, control system and vision [38]. Neural networks are including of simple elements that are operating in parallel. The biological nervous system inspires these simple elements. ANN can be trained to perform a specific function by adjusting the weights which represent the values of connections between elements. Usually, neural networks are trained or adjust so that a specific input to the system leads to the specific target output. The network is adjusted based on a comparison between the values of the outputs and the targets until the output values match the target values. Normally, many input/targets' sets are needed to train the network.

2.5.3 Deep Learning (DL)

Deep learning is considered a subset of machine learning; it is also called deep learning neural network because it follows the neural network architecture. Even though deep learning was introduced in the 1980s [46], it is still a technique that attracts and expands rapidly because of its higher accuracy than human intelligence and the training time that has been reduced because of using GPUs and cloud computing resources.

2.5.4 AI Applications in Routing in Cloud Networking

In recent years, the combination of artificial intelligence technology and cloud computing has become a popular research topic and has been integrated into large number of applications. This combination helps to improve significant optimisation effects and managements for cloud applications and services by knowing in advance how to utilise the communication and the computation capabilities efficiently. This section presents some AI contributions as the leading solution to improve routing in cloud networks.

2.5.4.1 AI in Cognitive Radio Network

This technology has appeared as a promising solution to overcome the limitation of frequency resources (spectrum) underutilisation by licensed users (Primary Users PUs) while on the other side it is overcrowded by unlicensed users (Secondary Users SUs) [39]. In any case, if the primary user wants to utilise its licensed spectrum, the secondary user(s) should leave the spectrum without disturbing the primary user activity. In a Cognitive Radio Network (CRN), a mobile terminal that is supported with cognitive radio capabilities will sense – discover communication environments like geographical location, spectrum holes, available service, and available wire/wireless networks, then data will be analysed, information from the environments with user's demands and preferences will be learned, self-configurations and adjustment for system parameters will take place to confirm specific regulations and policies. Figure 2-4 shows the cognitive spectrum management framework [40]. Heterogeneity of cognitive radio networks adds more complexity to its topology and routing information and introduces many security issues. Heterogeneity occurs in user terminals, wireless access technologies, services providers, and applications [41]. The new approach of designing CRN architecture is towards improving the whole network utilisation instead of just considering link spectral efficiency. From the end-user perspective, the concept of network utilisation means that the users can always be accessing CRNs anytime and anywhere with all demands being fulfilled. While from an operator's perspective, it provides better radio allocation and network resources to deliver an efficiently large number of packets per unit bandwidth and provide better services to the user. CRN can be deployed in distributed mesh and Ad Hoc architectures so that it can be of benefit to both unlicensed and licensed applications. The main basic components in CRN are the Mobile station, access point/base station and core networks/backbone.

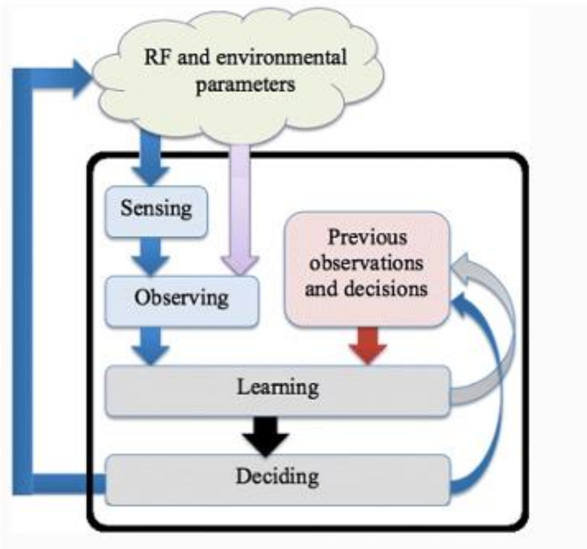

Figure 2-4 Cognitive spectrum management framework [40].

Research has proliferated in CR technologies as a concept and has been based on cloud realisations during the past few years. However, little progress has been made in introducing cognitive/intelligence capabilities into cognitive radios. Early works and efforts demonstrated that using artificial intelligence and machine learning, if suitably applied, could enhance cognitive radios adaptations [42][43]. However, further advances in this approach face huge, if not insurmountable, challenges that CR researcher faces to define what intelligence is adequately addressed challenges in CRs and which AI can achieve it with respect to radio node or networks.

2.5.4.2 Artificial Intelligence applications in MCCNs

Artificial intelligence and Machine learning provide a modern approach to mobile cloud computing for optimizing computation time and energy consumption. In [44], the author formulates a joint radio and cloud resource allocation problem for heterogeneous mobile cloud computing networks to achieve users' maximum utility and satisfy the QoS requirements such as latency. Genetic algorithm, ant colony optimization with a genetic algorithm, and quantum genetic algorithm have been used to solve the formulated problem. In [45], the author investigates a resource management problem of virtual machine placement in a physical machine for cloud computing data centres and figuring out which virtual machines have high data rates for communication to assign them in the same physical machine for minimizing network traffic within virtual machines.

2.6 Software Defined Network (SDN)

Programmable networks have been proposed as a new approach to facilitate network evolution. SDN technology has emerged as a means based on that approach by upgrading innovation in network management and its deployment services through the programmability of the essential network entities. This section introduces the concept, architecture and features of software defined network that can be used to address the complexity and congestion in traditional networks’ environments.

2.6.1 Introduction

SDN is an emerging network technology that has been proposed to replace traditional networking by providing an enhanced level of customisation and flexibility to meet the needs of newer network communications and mobility. The two elements taking part in forwarding packets via routers are control elements, which decide the priority and route the traffic should take, and a data element, which forwards the data based on control element policy. Prior to SDN's appearance, these two elements' functions were achieved in an integrated form at each network device (router, switch, bridge, and so on). Control in traditional networks is trained by means of a routing protocol implemented in each node in the network. This approach was inflexible and required all nodes in the network to apply the same protocol. With SDN, another approach is presented using a central controller that performs all routing and complex functionality like policy declaration and security checks. Figure 2-5 illustrates SDN architecture and its Application Control and Data planes [46].

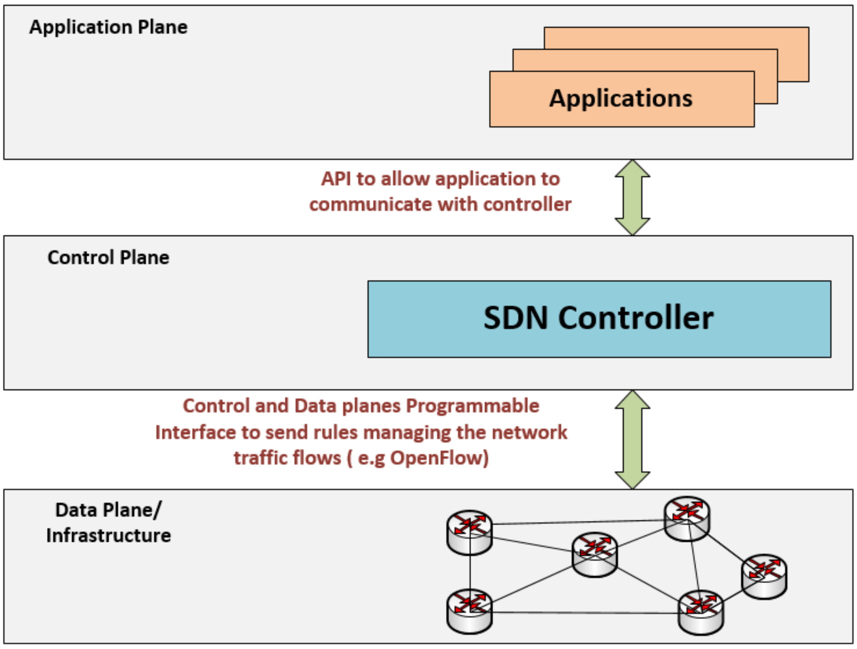

Figure 2-5 SDN networking architecture [46].

As shown from the SDN networking architecture figure, SDN application plane is the upper layer that consists of various applications and services to define, monitor, and control network resources. SDN control plane comprises one or more controllers. The controller is responsible for defining the data flows in the SDN data plane and is accountable for configuring each flow through the network. When the flow requested by the end system is allowed by the controller (depends on the policy), the controller starts to calculate a route for the flow to use and adds an entry for the flow in each of the switches selected along the path. The switches (physical or virtual) or also called OpenFlow switches here contain the data plane. Communication between the OpenFlow switches and the controller uses a standard protocol like OpenFlow protocol. API (Application Program Interface) provides the link between the services and applications running over the network and SDN controller.

With the increasing variety and volume of network traffic generated by high demand sources as mobile cloud traffic and cloud computing, it becomes very difficult to meet the QoS requirements. Networks need to be more scalable and adaptable. SDN can be presented as an adaptable intelligent solution to provide the required network scalability and adaptability.

2.6.2 SDN Applications in Routing in Cloud Networking

Recently, SDN architectures in which the control plane is separated from the data plane is becoming popular by researchers who have presented different intelligent control routing, data traffic monument and resources usage mechanisms. Based on the concept that the SDN enables Open flow switches [47] and forwards traffic through the data plane based on the control plane rules which is running on a separated controller [48] it will reduce the complexity of the hardware and let the networks to be controlled by standalone software that can be easily developed and tuned for the developer's or user's needs. Started with addresses the problem of routing control in traditional networks, the author in [49] proposed a dynamic routing framework using SDN that worked based on machine learning- Neural Network. In this method named NeuRoute, a default routing protocol based on Dijkstra's algorithm used by SDN/OpenFlow controller was presented for obtaining the shortest path and to provide APIs to develop custom routing for applications. NeuRoute achieves efficient routing with less execution time by applying real-time traffic matrix prediction, by learning traffic characteristics using a neural network, and finally generates the required forwarding rules to enhance network throughput. Several different approaches based on SDN architecture have been proposed for controlling routing. In [50], a new source routing implementation is proposed in a multi-hop wireless network based on SDN. In this novel routing protocol, OPNET is used to build the model and compare the proposed protocol with other traditional algorithms (including AODV, OLSR and GPSR). Results show that with SDN based algorithm, network lifetime is extended compared to conventional algorithms and also though SDN centralised controller, shortest path routing of nodes can be provided.

SDN is also presented as a solution to overcome the problems related to cloud network congestion. In [51], the author proposed a local rerouting mechanism using SDN based on datacentre networks to effectively manage congestion in the event of link failure or link congestion. Unlike traditional network reaction during congestion by notifying the source to resend the flow, this new mechanism reforwards flows at the point of congestion or one hop before to another available path based on the flow classification scheme. Results show improvement with respect to load balancing control and link utilisation. The emergence of edge computing as a complement of cloud computing can reduce energy consumption and maintain QoS for various applications. However, congestion in the underlying networks is still possible. In [52] author presented SDN based edge cloud interplay that deals with flow scheduling between edge and cloud devices by using multi-objective algorithms namely (Techebycheff decomposition). The new scheme makes trade-off between energy efficiency and bandwidth and energy efficiency and latency. Results are compared with existing methods. Results show improvements in flow scheduling in IoT environments. Further works were presented using the SDN concept with edge computing to improve processing time [53], with load balancing approach [54][55][56][57][58] using OpenFlow protocol, and also with mobility prediction [59].

2.7 Fuzzy Logic System

The fields of fuzzy logic systems are expanding rapidly with new applications and results. Because of its capability to handle uncertainties, Fuzzy logic has greatly supported cloud computing’s applications and services by performing logical operations.

This section presents the concept of fuzzy logic systems and briefly addresses some important researches on using fuzzy logic in routing and cloud networks.

2.7.1 Introduction

Fuzzy logic is the type of reasoning that helps make rational decisions in an environment of uncertainty and imprecision [60]. Its concept was formulated in the 1960s by Zadeh who worked at the University of California. Since that time, the fuzzy theory was rapidly evolved and applied. Unlike computers that use precise figures that have been either converted to zeros and ones or true and false, human brains can reason vague situations. Humans have common sense, which makes them able to reason in those things that are partially true. Fuzzy logic is the branch of machine intelligence that helps computers to picture the uncertain world. With the help of fuzzy logic, it can make the computers understand the vague concepts that can lead to developing technologies to judge the situation that is hard to define. When specific algorithms cannot dictate the system how to respond, the fuzzy logic can control the system by using the common-sense like feature. Figure 2-6 shows the architecture of a Fuzzy logic system.

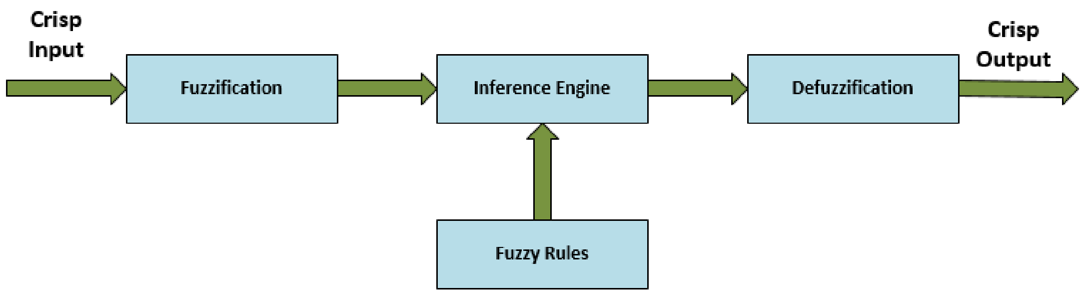

Figure 2-6 Fuzzy logic system architecture [61].

As shown from the fuzzy logic system architecture figure, the architecture of the fuzzy logic system consists of four different components [61] as follows:

- Rule Base: This component is basically the set of if /then conditions or the set of rules that is given by the user for controlling the decision making.
- Fuzzification: This model converts the crisp inputs from the measured sensors into fuzzy sets. The converted inputs are passed to the control system within the fuzzy logic system for further processing.
- Inference Engine: This component can be considered the central part of the Fuzzy Logic system because it maps the results to the input set and decides which rules to be applied for a given input. This is performed by calculating the percentage match of the rules for the given input.
- Defuzzification: This model is the opposite of the Fuzzification process. Here the fuzzy sets generated from the interface engine are converted to crisp values. These obtained crisp values are the outputs of the fuzzy logic system.

The Fuzzy Logic Systems structure is accessible and understandable. The algorithms can be described with little data and so little memory is required. All these advantages make using fuzzy logic system techniques in the decision-making process a perfect option.

2.7.2 **Fuzzy Logic Applications in Routing and Cloud Networks**

Fuzzy logic concept has been used in various technologies and research from a wide range of areas like engineering, aerospace, agriculture medical, environmental sciences, industrial, geological, mathematics and natural sciences [62]. In intelligent fuzzy routing solutions, the author in [63] proposed an intelligent routing - policing mechanism based on fuzzy logic and genetic algorithms. In Asynchronous Transfer Mode networks, it is important to design a proper traffic control because of the challenges shown in networks related to various services supported and the needs for effective network resources utilisation. To guarantee the required quality of the service, two important functions were used in Asynchronous Transfer Mode networks, which are policing and routing. The proposed system used the combination mechanism approach to get better traffic control. Results have shown that the proposed mechanism performance was higher than conventional routing algorithm and policing mechanism separately. In [64], the author proposed a new dynamic scheme of mobile Ad Hoc network routing protocol using fuzzy logic. This protocol intelligently selects the best route based on parameters (bandwidth, signal power, packet forwarding ratio). Route ranking as an output of the system provided for decision making, made the proposed protocol functional and effective. In [65], the author proposed expanding vehicular cloud services by using a fuzzy logic algorithm that was used to select the most proper cluster head. In addition, genetic algorithms were presented with this model to prevent more data loss at crossroads. The newly proposed method has shown promising results when compared to two other algorithms used before. In [66], the author presented a fuzzy-based power system to monitor and adjust electrical appliances' working time automatically. From the room's temperature and humidity measurements, the system will calculate the suitable working time for the electrical appliances. The current implemented Vehicular Ad Hoc Networks (VANETs) architectures faces several challenges due to poor flexibility, scalability, lack of Intelligence, and connectivity. For that reason, the author in [67], presented, compared, and evaluated two intelligent fuzzy-based systems for Resource Management called FSRM1 and FSRM2 for coordinating and managing Cloud, Fog, and Edge resources in SDN-VANETs. These systems are used to make decisions on the processing layer of the applications 'data in a Cloud, Fog, and Edge architecture and to come in handy when a vehicle requires additional resources to run its application(s). The decisions in this model are made by prioritising the vehicle's application(s) requirements and the availability of connections. FSRM1 considers the relative vehicle speed with the neighboring vehicles, the application data size, and its time sensitivity when deciding the processing layer. FSRM2 uses all these parameters together with the number of calculated neighboring vehicles as an input parameter when making the decision. Results from the simulations show an improvement in network management resources. In [68], the author proposed a novel fuzzy-based Adaptive green and reliable routing scheme. In this scheme, a fuzzy logic system was proposed to decide the number of reproduced packet copies to obtain an efficient data gathering. The fuzzy system interface takes the remaining energy and the distance to the sink of the reproducing node or source as its inputs. Therefore, it can tune the number of reproducing packet copies in an adaptable way. Simulation results show improved energy efficiency and extended network lifetime under the guarantee of transmission reliability.

Three main factors make implementing a fuzzy system in network routing decisions and data management more effective, efficient, and accurate. The first is to know and choose the best technique to collect the data required that will be used as an input for the fuzzy logic system. The second is to maintain updating of all these data during task duration, which includes assuring the accuracy of the data values. Finally, it is to know the best location to decide the best route. Different approaches and algorithms are proposed, but no attention is considered to combine all these factors into one system.

2.8 Intelligence in Load balancing

Recently, the increasing complexity of cloud applications, communications networks, and the rapid growth of data traffic has created significant challenges for service management and network. The traditional load balancing model has not been integrated into the cloud network platform; it works as an independent model. In this case, random mobility, distribution of terminals, and various quality of service requirements for the mobile user services may cause an unbalanced distribution for the network traffic, leading to overloading and congestion on some nodes with a heavy load that causes an increase in packets lost and increase in service latency, while other nodes with idle resources have a light load and poor utilisation. An integration of cognitive network balancing and cloud networking has been offered. The workload with predictive frameworks based on artificial intelligence were proposed that offer the same service as resource manager by providing an estimated measure of the future workload. It anticipates future traffic trends in terms of user requests and resource utilisation patterns based on previous information. These predictive frameworks consist of intelligent resources management systems based on algorithms used to scale up the growth of the service provider by maximising the throughput and minimising energy consumption. The author of [69] proposes a self-directed workload forecasting method (SDWF) that determines the workload on cloud servers by detecting its past estimation error trend and improving its future prediction accuracy. The proposed SDWF uses the optimisation approach of a black hole algorithm and learns the network load efficiently, providing better performance results as compared to deep learning, differential evolution, and backpropagation algorithms. The author in [70] uses a combination of ant colony optimisation (ACO) algorithm with a fuzzy logic approach to propose a hybrid algorithm that improves load balancing, processing time and response time in a cloud computing network. The fuzzy algorithm used improved the computation duration and enhanced the ACO algorithm performance.

2.9 Pattern Prediction Analysis

Nowadays, the network's data traffic has highly increased due to the appearance of new technologies and applications. Networks supporting these services and applications have to cope with growing traffic demands, and they must provide the best quality of service to the user. Therefore, efficient utilisation of networking resources and the information that can be extracted from the networks have become crucial.

Many new intelligence-based techniques have been proposed that uses the information extracted from the network for pattern predictions system to overcome the traditional network routing and data analyses challenges. The aims to improve the network can be achieved effectively and efficiently only when the predicted data are measured accurately. Pattern prediction models have been proposed as a solution in various applications such as in network resources management [71], energy saving [72], and wireless sensor networks [73].

This section started with discovering mobility pattern prediction by knowing mobility models. Also, an explorer of some predictors with different techniques and complexity has been explained, and some related works were covered.

2.9.1 Mobility Pattern Prediction- Models

In this section, an overview of some mobility models proposed and used by researchers will be discussed and presented. There are many different models for realistic mobility patterns that are related to object taxonomy, such as Pedestrians Mobility, Marine and Submarine Mobility, Earthbound Vehicles, Aerial Mobility, Medium based Mobility, Mobility in outer space, and Robot Motion [12], which have been classified based on applications, motion speeds and working dimensions. This section will focus only on Pedestrians Mobility. This type can be considered as the oldest and the most common way to mimic mobility in the walk. Unlike other modern mobility patterns, this model has the slowest velocity values and draws people's walking. In wireless networks, pedestrian mobility can be applied as people holding cellular phones walking in a mall or street. Limited mobile resources like battery charge capacity can be considered the main side effect of this mobility type.

In mobility models, that have been approved, namely: Random Waypoint Mobility Model and Random Walk Mobility Model, are the most commonly used in research than others. Types of mobility models can be categorised as follows:

- **Random Waypoint Mobility Model**: In Random Waypoint Mobility Model [74], the motion begins by selection of the initial positioning for the node and staying in that location for a period of time called (pause time), once this time has come to an end, the mobile node chooses another random destination within the simulation area and a random speed that is distributed between [minimum speed, maximum speed]. The mobile node then will travel to the new selected random position at the selected random speed. When the node reaches the new position, it will pause for a specified random time period and then start again to move to another new position. The Random Waypoint Mobility Model is a common model that is used in mobility model research [75][76][77][78]. In some cases, this model can be simplified by using the same concept of motions without pause time [79].

  Figure 2-7 illustrates an example of mobile node motion using Random waypoint Mobility model [80]. In this scenario, the speed was selected between 0 and 10 ms^{-1}, new positioning has been chosen randomly after the pause time period has come to an end.

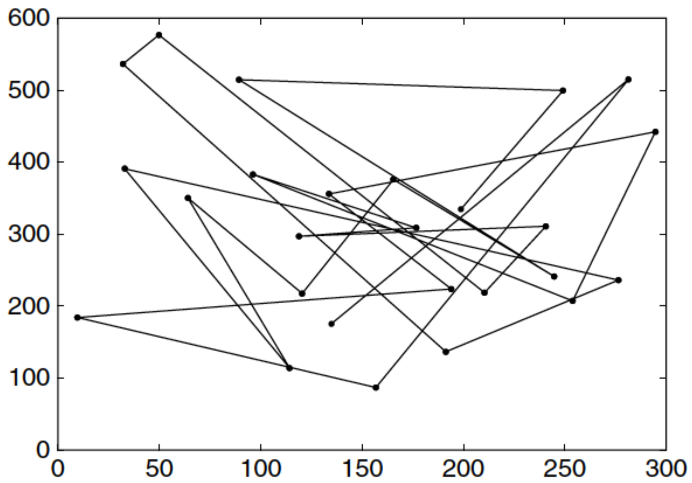

Figure 2-7 Movement pattern of mobile nodes using Random Waypoint Mobility Model [80].

- **Random Walk Mobility Model**: This mobility model was mathematically explained in 1926 by Einstein [81]. It was presented to mimic an extremely unpredictable way of motion for many entities in nature. In this model, the nodes move from their current position to a new position with a specific speed, both values of the new positioning (direction) and the speed of movements are selected randomly in the predefined range [0, 2π], [speedmin, speedmax] respectively. The movements in this model occur in either constant distance travelled or constant time interval, at the end of each new position the speed and the direction are calculated. Suppose the nodes reach the simulation's platform boundary. In this case, it will hit and move back from boundary with an angle determined by its incoming direction and then it will continue to its new direction. There are many derivatives within this model that has been developed like 1D, 2D, and 3D. The Random Walk mobility model is widely used [82][83][84][85]. However, this model in a memoryless mobility pattern, which means it keeps no knowledge regarding its previous speed and location values [86], In other words, the current direction and speed values are independent of its past values [87]. In this model, if the time and distance are specified, mobile nodes motion in this model will be short, making the movement pattern random roaming but restricted to a small territory in the simulation area. In this case, a larger value should be used to assign the movements of steps before changing directions. Figure 2-8 illustrates an example of this 2D movement model [80], where mobile nodes begin their movements form the centre (150x300) m of the simulation area (300x600) m, each node will randomly choose its new direction between [0, 2π] and a speed between 10 ms^{-1} , all nodes are allowed to travel for 60s before changing direction and speed. The movement pattern of Random Walk Mobility Model is similar to the movement pattern of Random Waypoint Mobility Model when the pause time is zero.

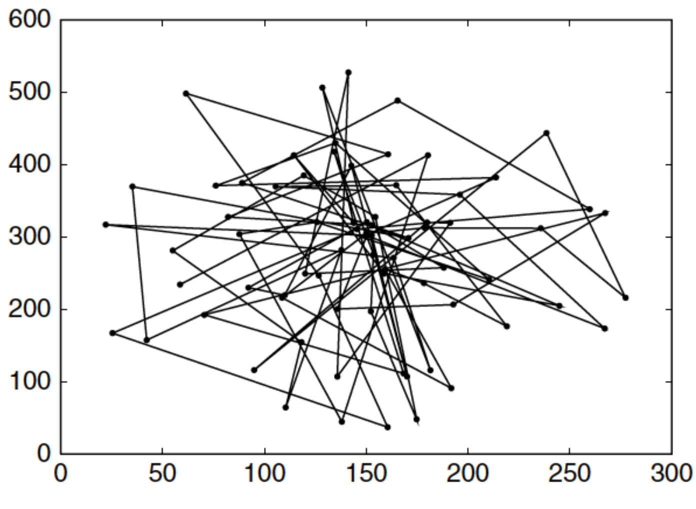

Figure 2-8 Movement pattern of 2D mobile nodes using Random Walk Mobility Model (Timing) [80].

- **Random Direction Model:** This model [88] was created to overcome density waves phenomenon that is shown mostly in Random Waypoint mobility model. A density wave can be described as a gathering of nodes in one part of the simulation area. In the Random Waypoint mobility model clustering happens close to the middle of the simulation because the probability of each node passing through the middle of the simulation area each time it is assigned its new randomly next positioning is high. Thus, nodes are shown to be clustered, then dispersed and then again clustered to each other again. To alleviate this phenomenon, Random Direction model was developed, where mobile nodes select their direction randomly like Random Walk mobility model then when they reach the border of the simulation area, they are paused for a specific time then choose another direction with an angle between [0, 2π] and move toward their new direction which is selected randomly. Figure 2-9 illustrates an example of this movement model. From the figure, the average hop count of packets using this movement model will be higher than the previous models described. Furthermore, this model is most likely to be used with network partitioning compared to other models. Few more mobility models have been presented and described [80], such as boundless Simulation Area Model [89], Gauss-Markov Model [86], Probabilistic Version of Random Walk [90] and City Section Mobility Model [91]. In addition, there are other mobility models for multiple mobile nodes that are dependent on each other with respect to their movements like Exponential Correlated Random Mobility Model [87], Column Mobility Model [92], Nomadic Community Mobility Model [93], Pursue Mobility Model [94] and Reference Point Group Mobility Model [95].

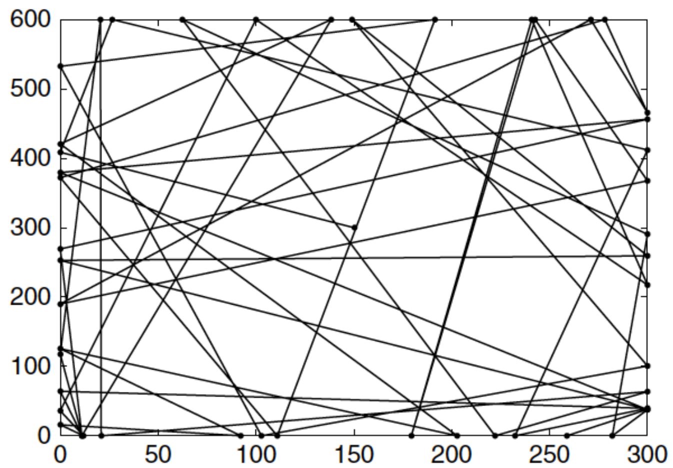

Figure 2-9 Movement pattern of mobile nodes using Random Direction Mobility Model [80].

2.9.2 Mobility Pattern Prediction Methods

The mobility prediction methods can be classified into three categories [96]:

- Prediction based on movement history: in this method, the network's future topology is predicted from the previous movement pattern. In a highly mobile network model, this method might fail to give accurate results.
- Prediction based on topology: in this model, prediction depends on the physical characteristics of the mobile user nodes, which includes for example using GPS embedded on the device to get an estimation of the future location of the mobile node.
- Prediction based on the logical topologies: this model uses the logical topology of the network to calculate the node movements based on different parameters such as line cluster position, which represents the distance from the cluster head, or neighbour's movements.

2.9.3 Prediction Techniques- Time Series forecasting

Time series forecasting has been considered an important research area in many fields because different types of data are collected and saved as a time series. A lot of time series data can be found in weather forecasting, medicine, biology, stock prices forecasting, and it can be used as well in decision making applications [97]. Considering the growing availability of computing power and data in recent years, the deep learning concept has become an essential part of the presented new generation of time series forecasting models that, obtain stunning results.

Time series consists of main components that can be explained as follows:

- Long term trend: It represents the overall direction and approach of the data achieved, while ignoring any short-term turbulence that might affect it such as noise.
- Seasonality: it is related to the periodic fluctuation of the data that are repeated during the time series period.
- Stationary: it is one of the important characteristics of time series. It refers to time series as stationary when covariance and variance do not have significant changes over the considered time period.
- Autocorrelation: It is used to identify the seasonality characteristic and trend in time series data. It refers to the correlation between the time series and a lagged version of itself.
- Noise: It refers to all random fluctuations or variations of data because of uncontrolled factors. Every set of data includes noise.

2.9.3.1 Traditional and Deep Learning Time Series Forecasting

Time series with traditional machine learning like ARIMA models and Exponential smoothing forecasts have many limitations, so that they are not suitable for long term forecast (steps), not suitable for recognizing complex patterns in the data, and any missing values can affect the performance of the model. To overcome the traditional machine learning disadvantages, deep learning for time series forecasting has been used with different approaches.

In this chapter, six deep learning architectures for time series forecasting are presented [98]:

- **Recurrent Neural Networks (RNN)**:

  Recurrent Neural Networks can be considered as a development to the conventional feedforward neural network. It consists of a neuron that looks like a node, and these nodes are organised into layers whose architecture is similar to the standard Neural Networks. The neurons are divided to form an input layer, hidden layers, and output layers. The connection between each neuron has a trainable weight. Every neuron in this model is assigned to a fixed time step. That also includes the neurons in the hidden layer that forward data from the input in the forward direction with time dependency. In this model, all neurons are fully connected only with the hidden layer's neurons with the same assigned time step and connected through a one-way connection to every neuron assigned to its next time step. Hidden layers are connected to both the input and output neurons only to the same assigned time step [99].

  The activation of the neurons can be considered here as time ordered, because the one-time step of the output of the hidden layer is part of the input of the next-time step. The RNN architecture shown in figure 2-10, shows that the input vector is x(t) at time step t.

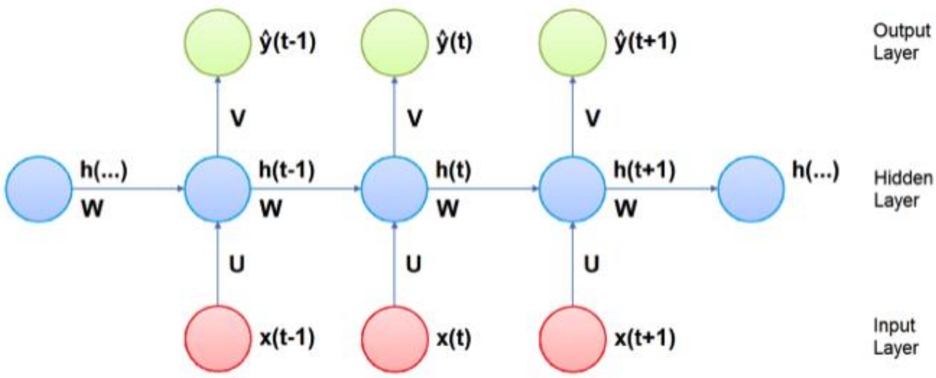

Figure 2-10 RNN Architecture [98].

This input vector is connected to the hidden layer neurons of time t with a weight Matrix (U). That hidden layer neurons are connected to neurons of time t+1 and t-1 with a weight matrix of (W), and at the end that hidden layer of neurons are connected to the output vector of time t with weight matrix (V). Weight matrices are all constant for each time step t. The vector h(t) represents the hidden state at time t, and it is a type of network memory, which can be calculated from the current input and the previous time step of the previous hidden step.

$$h(t)=\sigma_h(W_h(\ t\text{-}1)+Ux(t)+b_h) \tag{2.1}$$

$$y^{\wedge}(t) = \sigma_y\ (\ V_h(\ t)+\ b_y) \tag{2.2}$$

σ_h, σ_y are the activation functions, b_h, b_h are the bias, $y^{\wedge}(t)$ is the output vector at time t.

In general, recurrent neural network cannot be affected by any missing values. It can detect complex patterns within the time series input. It can also give more good results using fewer steps. However, the training of the RNN is computationally expensive and it suffers from a weak memory that is unable to take several values in the past for the prediction of the future.

- **Long Short-Term Memory (LSTM)**:

In the mid of 90s, LSTM has been developed by placing LSTM unit in place to a hidden layer of RNN; this was proposed to overcome the vanishing gradient problem in the standard RNN by enhancing the gradient flow in the network. In Figure 2-11 below, the LSTM unit consists of a cell state that represents the network's memory and is responsible for bringing information along the whole sequence. A forget gate decides what is applicable to keep from the previous time steps. An input gate decides what is applicable to add from the current time step. An output gate decides the value of the output at the present time step. Same as RNN, with LSTM unit, the input vector at time (t) is connected to the LSTM-cell of time (t) with a weight matrix of (U), the LSTM- cell is connected to the LSTM- cell of time (t-1) and (t+1) with weight matrix (W), and the LSTM-cell is connected to the output vector of time (t) with wight matrix (V). In this unit the matrices (W) and (V) are divided into submatrices (W_f, W_i, W_g and W_o) and (U_f, U_i, U_g, and U_o) that connected and shared across time to different elements in the LSTM unit.

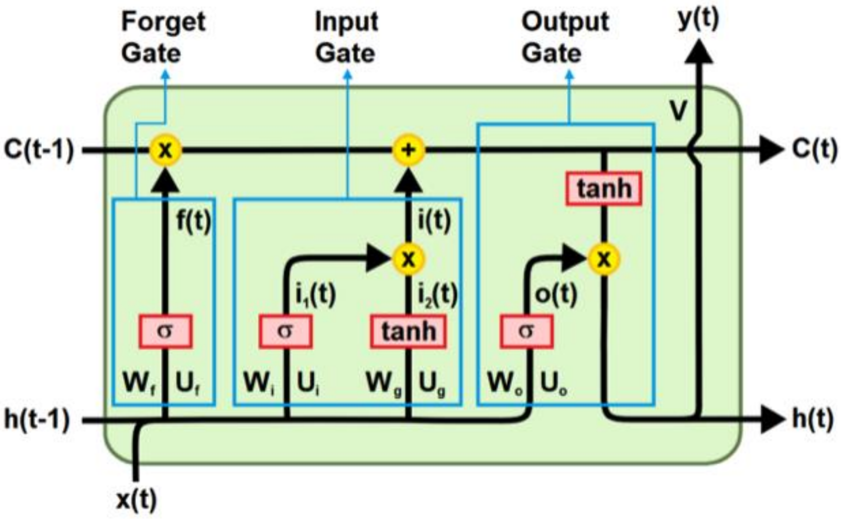

Figure 2-11 LSTM architecture [98].

The effect of short-time memory is reduced by making the relevant information transfer from the cell state during the processing, also the information that comes from the previous time steps arrive at each time step. The gate learns during the training overall time, which information is important to forget or to keep, and it will add them or remove them from the cell state. This process enables LSTM to recover the data transferred into the memory.

The forget gate represents the first gate. This gate will work as a filter to decide which information should be deleted or saved. The information from the current input and the information from the previous hidden state are passed through a sigmoid function. If the output is close to (0) it means that the information can be forgotten, while if the output close to (1) then it means that information must be saved.

$$f(t)= \sigma(x(t)U_f+h(t+1)W_f \tag{2.3}$$

In the input gate, which represents the second gate that will be used to update the cell state. Initially, the current input and the previous hidden state will be given as inputs to the sigmoid function, where the closer the output to value (1), the more important the information. It also passes the current input and the hidden state to a tanh function to squeeze the values between (-1) and (1) to improve tuning of the network. The output of the sigmoid and the tanh are multiplied element by element. The sigmoid is the factor that decides what to keep of important information from the tanh output.

$$i_1(t)=\sigma(x(t)U_i+h(t-1)W_i) \tag{2.4}$$

$$i_2(t)=\tanh(x(t)U_g+h(t-1)W_g) \tag{2.5}$$

$$i(t)=i_1(t)*\ i_2(t) \tag{2.6}$$

Cell state can be calculated after the activation of the input gate, where the call state of the previous time step will get elementwise that is multiplied by the output of the forget gate. This will help to provide the possibility to ignore the values in the cell state when multiplying it by values close to (0). Then input gates' output is elementwise added to the cell state. The output will be the new cell state.

$$C(t)=\sigma\ (\ f(\ t)*\ \ C(\ t-1)+i(t)) \tag{2.7}$$

Finally, at the final gates which is the output gate, it will decide the value of the next hidden state, which includes information about the previous inputs. At first, the current inputs and the previous hidden state are summed and passed to a sigmoid function. After that, the new cell state is moving to tanh function in which its output with the sigmoid function output will be multiplied to decide what information should be contained within the hidden state. The output represents the new hidden state. The new hidden state and the new cell state are then moving to the next time step.

$$o(t)=\sigma\ (\ x(\ t)U_0+\ h(\ t-1)W_o) \tag{2.8}$$

$$h(t)=\tanh\ (\ C_t)*\ o(t) \tag{2.9}$$

- **Gated Recurrent Unit (GRU)**:

Gated Recurrent Unit was developed as a new generation of RNN with similarity to LSTM to overcome the vanishing gradient problem of the standard RNN. In GRU, it uses the reset gate and the update gate. These gates are responsible for deciding what information should be acceptable to pass to the output. Information can be kept from many times steps before the current time step are trained by these two gates, which can be done without washing it through time by removing information that is not related to the prediction. With good training, it will be possible for GRU to perform extremely well, even in a complex scenario. As shown in figure 2-12, the GRU unit consists of a reset gate responsible for deciding how much information that comes from the previous time steps can be forgotten. An update gate that is responsible for deciding how much information that has come from the previous time steps can be saved.

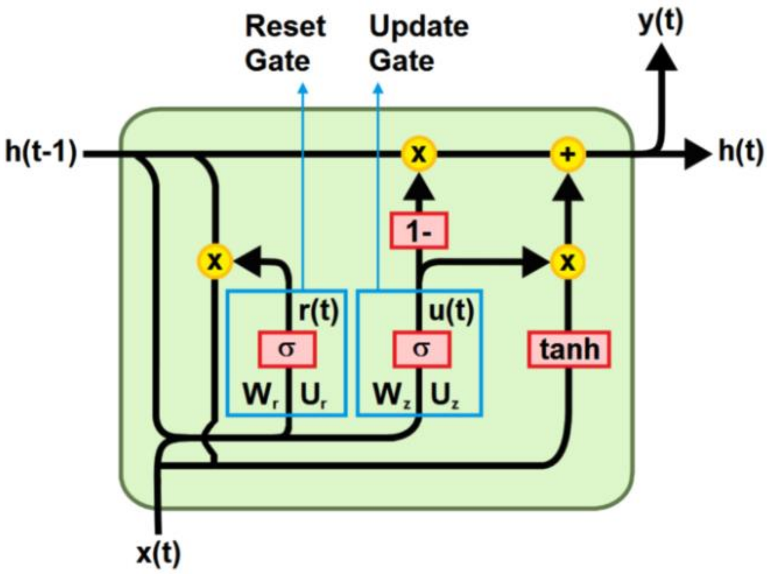

Figure 2-12 GRU architecture [98].

The first gate in GRU is the reset gate, which determines how to do a combination between the new and the previous memory and decides how much information to be forgotten that comes from previous time steps. Weighted sum between the memory $h(t\text{-}1)$ and the input $x(t)$ will occur while the information for the previous $t\text{-}1$ steps are held, then to squeeze the results between (0) and (1), a sigmoid application function is applied.

$$r(t) = \sigma\,(x(t)U_r + h(t-1)W_r) \tag{2.10}$$

The next step is the update gate that helps to determine how much information comes from the previous time steps to be passed to the future. This feature is a powerful one because it eliminates the risk of vanishing gradient problem by making the model to decide to copy all the information that comes from the past.

$$Z(t)= \sigma (x(t)U_z+ h(t-1)W_z) \tag{2.11}$$

The relevant information from the past that has come from the rest gate is stored in the memory. At first, element by element multiplication is applied between the final memory at the previous time step h(t-1) and the output of the reset gate r(t) is computed. Then the weighted sum between the input x(t) and the result is completed. Then at the end, the activation function tanh is applied

$$h(t)= \tanh (x(t)U_h+(r(t)* h(t-1)W_h) \tag{2.12}$$

Next, to calculate h(t), which represents the vector that has the information of the current unit to send it to the next time step. It discovers what to collect from the previous steps h(t-1) and the current memory content $h^{\sim}(t)$. Element by element are multiplied and computed between 1- (1- z(t)) and h(t) and the update gate z(t) and h(t-1). At the end, weighted sum calculated between two results.

$$h(t)= \left(1- z(t)\right)*h(t-1)+z(t)*h^{\sim}(t) \tag{2.13}$$

- **Stack LSTM**:

  Unlike the standard LSTM model which consists of a single hidden LSTM layer followed by a feedforward output layer, Stack LSTM is a development version with multiple hidden LSTM layers, where each layer consists of multiple memory cells. Figure 2-13 illustrates Stack LSTM architecture. The hidden layers make the model more accurate and deeper. Stack LSTM is considered as a reliable model in solving sequence prediction problems.

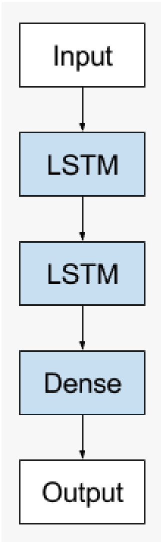

Figure 2-13 Stack LSTM architecture [100].

- **Bi-directional LSTM**:

  This model is a combination of LSTM and RNN [101]. This structure provides the networks to have both backwards and forward information related to the sequence at every time step. By using bidirectional neurons, it will allow the model to run the inputs in two ways, from the past to the future and from the future to the past. Input sequence time steps are proceeding one at a time, while the input sequence network steps are run simultaneously in both directions. However, Bi-directional-LSTM is not a suitable solution for all sequence prediction problems. Figure (2-14) illustrates Bi-directional LSTM architecture.

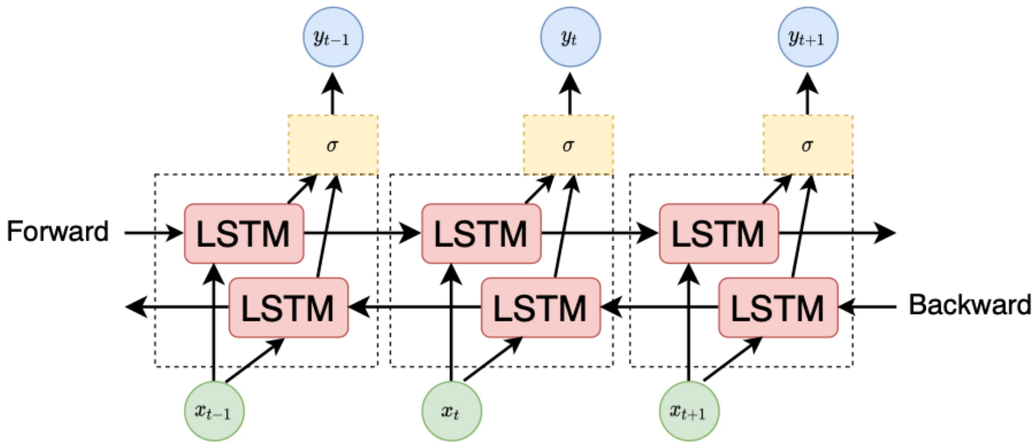

Figure 2-14 Bi-directional LSTM architecture [102].

- **CNN-LSTM**:

  This model is a combination of LSTM, which is efficient in learning and extracting the long-term dependencies and the traditional neural network which is efficient at one- dimensional data [103], Figure 2-15 shows CNN-LSTM architecture.

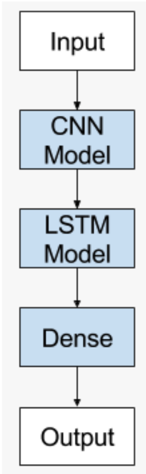

Figure 2-15 CNN-LSTM architecture [100].

2.9.4 **Intelligent Prediction Related work**

Many Researchers have used time series and deep learning models and Artificial intelligence methods for predicting networks. Based on real trajectory data, the author of [104] presented a method to predict pedestrian's future position. The results show a good performance of this method on various types of trajectories. This technique works on pedestrian traced data, and its performance improved when working in high mobility scenarios such that the error will change from 3 meters to less than 1 meter. In [105], a Markov model was proposed as a mobility prediction method. To use this technique, the area of simulation that has been used for mobile moves are geographically divided into cells to shape the Markov Chain. Increasing partitioning of the geographical area of movements into a larger number of smaller sizes, will improve the timing required to calculate mobility behaviour (accuracy rate). In a new model proposed in [106] using encoder-recurrent-decoder networks based on LSTM, the author designed a non-linear transformation model for prediction and recognition of human body pose in motion and video capture. The recurrent unit constrained human motion prediction using the history information passed through it. Another approach for short-term human motion prediction has focused on neural network architecture [107][108][109]. Recurrent neural networks (RNNs) have been introduced because they are able to store relevant recurrent information in their hidden state.

2.10 Summary

In this chapter, the fundamental basic technologies of cloud computing networks, concepts and evolutions were reviewed and presented. The chapter also gives a brief explanation of cloud networking and particularly on mobile cloud computing network’s challenges by concentrating on routing that raises the question of whether existing traditional routing and data management solutions can be ported to mobile cloud networks. A study on using cognitive data routing approaches was presented by showing various intelligent techniques that research has offered such as artificial intelligence algorithms, software defined networks and fuzzy logic systems that can be used to overcome challenges of mobile cloud networks. Also, in this chapter, a survey of pattern prediction models, mobility pattern prediction methods and prediction techniques of six time-series forecasting algorithms were discussed. In summary, mobile cloud networking raises some unique design challenges related to its services and network heterogeneity, requiring a cognitive architecture tailored for modern cloud applications and dynamic topologies environments. It should be aware that while the aim of this chapter has been to give details and cover as many various aspects on cognitive solutions in cloud networking as possible, it is not the most comprehensive of the solutions available. There are plenty of other resources and research, and many investigated different issues.

Chapter 3

COGNITIVE DATA ROUTING IN HETEROGENEOUS MCCNs

3.1 Introduction

Nowadays, using cloud computing becomes a significant computing paradigm in utilising resources over the Internet. This new computing paradigm that builds upon advanced research in resources virtualisation, distributed, grid and utility computing [110] provides flexible, abstracted, and virtualised services on demand. The basic concept of cloud computing is that computing resources provided as a service, such as computing power and storage, are not stored locally but are stored in the cloud's datacentre. One of the cloud's advantages is that it provides a variety of services to the cloud users, such as infrastructure, platform, and software as a service (IaaS, PaaS, and SaaS) [111]. The cloud users will access services through the Internet so that maintenance and the installation cost decreases while the flexibility and scalability increase.

Recently, the popularity of mobile devices and the demand for applications are rapidly increasing. Moreover, the worldwide shipments of tablets and mobiles phones are rising to 2.16 billion units in 2020. Also, increasing availability of 5G handsets will replace the old devices, accounting for over 50% of the mobile phones shipped in 2023 [112]. However, the constrained resources of mobile devices, such as the limitation of their computing power and storage capacity in addition to mobility, causes an impedance to the running of resource-intensive mobile applications. To overcome such challenges, a new concept has been proposed by integrating mobile devices and cloud computing to obtain the concept of mobile cloud computing (MCC). In this concept, all or some of the resources-intensive applications will be performed outside mobile devices using cloud-based resources by offloading intensive computational tasks and most of the storage and processing related to mobile applications from limited capability mobile devices to powerful, centralised remote processing device(s)/server(s) on the cloud. In this scheme, all the communication between mobile devices and remote cloud servers are carried through cloud networks [113]. The availability of a good and high-speed network connection will ensure the advantages of mobile cloud-based concept without experiencing any significant delays associated with mobile communication technologies. The most apparent challenge with the traditional networks is how to fully leverage mobile cloud network's capabilities with respect to conventional network capacity and reliability as in some applications like those related to healthcare [114], disasters [115], interactive applications and real-time media content analysis [116], where it is always important to keep those applications working in real-time streaming, high QoS, using efficient resources and the connectivity to backend should always be available anytime and anywhere. It will be helpful to consider cognitive cloud networks as an intelligent solution to overcome mobile cloud network challenges.

3.2 Research Gap

Using the mobile cloud concept is being widely addressed in today's research. In this section, the popular mobile cloud networking architecture, as shown in Figure 3-1, was introduced and categorised with respect to how it provides services for the users. The architecture can be classified into three types: -

- Client-Server Architecture (C-S): In this model, some or all of the tasks or complex applications are offloaded from the mobile device (Client) to a computational infrastructure hosted a cloud (server), which remains static and provisioning services to the mobile users [117].
- Cloudlet Architecture: - Is a nearby resources-rich computer or server with one hop, low latency, high bandwidth accessing from mobile device [118].
- Virtual AD Hoc Architecture: - A group of mobile devices that formed a virtual cloud by sharing its resources to provide special tasks to other mobile devices [119].

A number of researchers have presented so many improvements in mobile cloud networks and mobile computing concepts from different perspectives, such as offloading [120], allocation, scheduling, and virtual machine deployment in cloud computing [121]. Some other researchers presented the concept of using multiple cloud computing providers to host user's data [122], [123]. However, up to date, it is evident that not too many of the research provided any mechanism of extending new intelligent functionalities to legacy cloud network architectures and presented it as one framework.

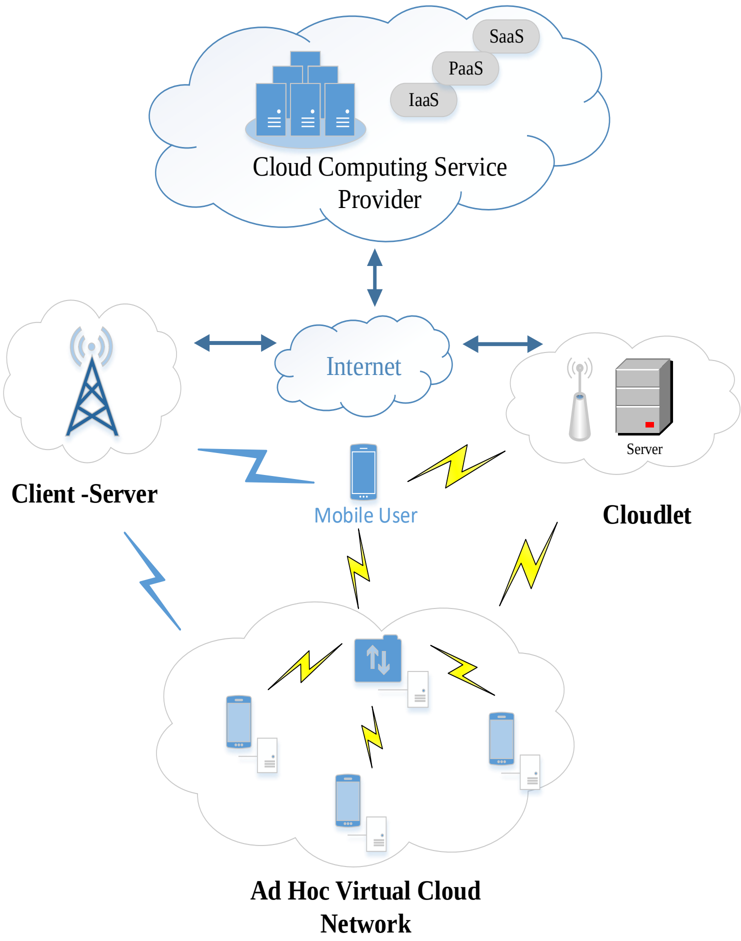

Figure 3-1 Heterogeneous Mobile Cloud Networking Architectures.

Various topics have been covered by researchers, such as energy consumption [124], [125]. Some research proved that a mobile node using the 3G/4G network consumes energy higher than a node using Wi-Fi. The author in [126] presented the concept of a software defined cognitive radio system: Cognitive Wireless Cloud (CWC) from a network's viewpoint. However, this solution does not consider mobile local resources during the design for the appropriate tasks, making the system incapable of working efficiently if there is a lack of mobile resources. In [127], the author proposed a new concept in constructing a mobile cloud computing system by integrating P2P networks and heterogeneous wireless networks to perform services and resources exchange among peer nodes. However, this design only solved the information querying problem, which is generally looked at as the classic problem of P2P networks. Mobile edge cloud computing is a new concept proposed to overcome the delay in using traditional mobile computing systems. In this new concept, multiple stationary and mobile devices are interconnected to create a small cloud infrastructure for several IoT that require a huge amount of storage and computing resources with minimum time delay. Node's mobility in edge computing and the poor design of its network [128] can significantly decrease its lifetime due to an increase in energy and communication consumption cost. Many cloud applications require a rapid response and a long working duration. Fog networks also face challenges related to providing the required QoS for latency-sensitive applications and dealing with the various fog node resources and network links [129].

In general, every cloud computing network can be considered as a combination of the deployment model and service. Nevertheless, of the type of cloud computing, however, one crucial fact that is always true, no network indicates no cloud.

3.3 Integrated Service Routing Oriented Architecture in HMCCNs

In this work, an adaptive solution is proposed based on cognitive methods to overcome the situation when mobile channel connectivity or capacity has a problem with the initial working model formed, such as in mobile cloud computing network with (Client-Server Architecture). The adaptive model proposed, namely: Heterogeneous Mobile Cloud Computing Networks (HMCCNs), create and integrate different cloud network architectures to work as one structural unit that helps achieve the required task when it is impossible to accomplish it using only traditional networking methods. In the simulation, different assumptions and scenarios have been implemented and proposed. The working case started by assuming that a variety of resources and alternative wireless connectivity(s) were available in the network that will be simulated. Then by optimising data routing decision, the data is passed across the proposed heterogeneous network model in a cognitive way to ensure reliability, scalability, economy and in real-time. In this work, a Long-Term Evolution (LTE) cellular network that consists of a cellular base station (eNodeB) and several mobile devices connected to it has been used and simulated. The LTE cellular network was connected to the back haul with a server that worked as a mobile cloud services provider. The LTE is a radio technology designed to increase cellular networks' capacity and speed. In LTE, the downlink peak rates of at least 100Mbps and 50Mbps for Uplink were used. The Radio Access Network (RAN) round-trip times in this network less than 10(ms). This network is supported with Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD) and scalable carrier bandwidths with a range from 20 MHz down to 1.4 MHz. The main reasons for selecting the LTE network in this work are its low latency and high throughput capabilities.

In general, multimedia data consists of image, audio, and video that requires high mobile resources and wide bandwidth capacity in the network for transferring data tasks, as in the many cases where many mobile users are uploading multimedia files in cloud storage services to share them with a friend(s). In other cases, some intensive and time-sensitive applications need to upload a huge stream of information to analyse it and use it for real-time decisions. In this work, a special case is presented that was started by assumed that MobA is a mobile user device connected to an LTE cellular network. The user has a special cloud task to upload real-time multimedia data (video streaming) for a certain duration of time to store it in the server for real-time analysis. During the data traffic uploaded process, the LTE cellular link capacity (bandwidth to deliver data) after a while will become a bottleneck and could not handle real-time transmission so that the network latency increases to be above the acceptable limit, also in the case when accessing to the cellular network is not guaranteed (connection to the cloud) broken. Unlike traditional mobile cloud network architecture as an individual cloud network model, when dealing with such networks' link failure issues, in the proposed integrated model, MobA user device that was disconnected or suffered from channel congestion in the LTE channel network will be connected to another neighbour mobile devices. A free high-bandwidth WI-FI network will establish these connections to form a virtual Ad Hoc mobile cloud network that will become a core component cooperated in sharing the cellular wireless channels of these mobile devices shown in Figure 3-2.

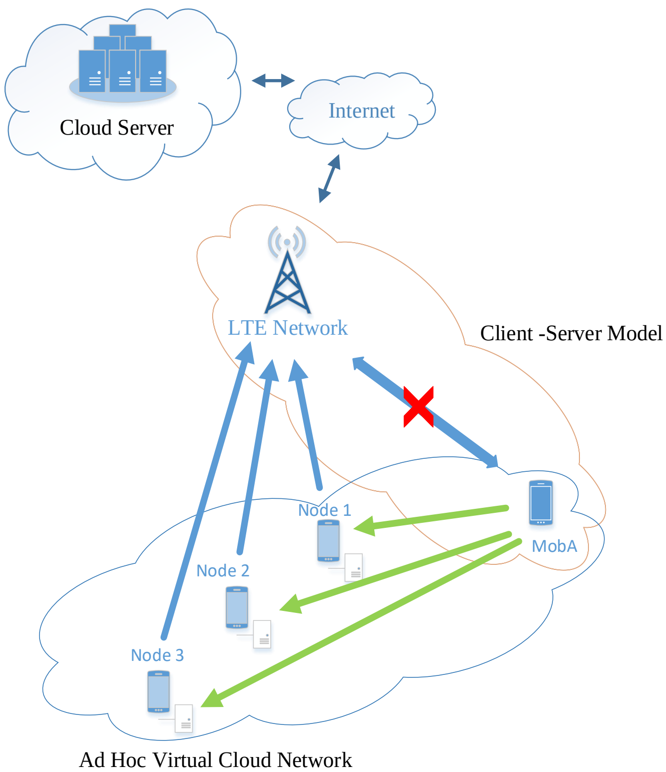

Figure 3-2 Cognitive data routing in heterogeneous mobile cloud networks.

This migration helped to do the necessary task in such cases anytime and anywhere, meeting efficiency, reliability, and high performance in uploading the required task to the cloud server. Achieving reliability data routing in the proposed heterogeneous mobile cloud computing network model required using a novel cognitive routing algorithm and intelligent data management methods to enhance network resources utilisation and increase capacity link problems of mobile user networks.

In this experiment, the virtual Ad Hoc mobile cloud network was created and chosen for the appropriate task and integrated with the original client-server model to improve the network performance by increasing its throughput and reliability. In order to achieve that, two approaches have been presented in this work as follows:

- **Local Node Management System**:

  This presented approach is based on the assumptions that routing management was built inside the user mobile MobA device. It locally decides the adaptive cloud architecture network used. It also decides when to switch from one cloud network architecture model to another or, in some cases, multiple models can be used to form the heterogeneous mobile cloud network as per tasks' requirements and availability of resources. Data migration or offloading was used to outsource intensive tasks partially or entirely. In this approach, user node MobA has a command-and-control management function that is used to initiate the following procedures:

  a) Resources Scanner: This function is responsible for searching and collecting information about neighbour terminal(s) location, available connectivity and its types and costs of the interfaces to the mobile user device such as cellular link channel, Wi-Fi, and Bluetooth. The resources information mentioned is the primary criteria for allocating or establishing the required cloud network model(s) to perform the task. The searching process for collecting resources information of other neighbouring mobile nodes and connections continues. All the collected data discovered is sent back to the mobile user to save it inside the database unit. The database unit is the memory part of the user node management structure that will save the required local and public network resources, such as node(s) energy, CPU, and memory size values.

  b) Cognitive routing algorithm: a cognitive routing mechanism is used in this approach that is able to be adaptable to the network topologies changes with respect to the network information collected and applications or tasks requirements.

  c) Decision and execution: The mobile user node MobA is offloading the task based on the HMCCNs concept proposed.

As mentioned before, there are different types of cloud networks models with various links, resources, and conditions. Offloading the same task to each one of these models produces different network performance results. Therefore, a cognitive method to select the optimal model(s) is needed especially when using the new proposed HMCCNs model. In this experiment, within MobA node, is selecting optimal cloud network model from the whole integrated HMCCNs for offloading the task required new additional routing parameters and criteria that should be independent of the routing algorithms. This is achieved by presented the Fuzzy Analytic Hierarchy (FAH) system. This system was proposed by Saaty [130]. The aim of using this method is to break down an unstructured and complex situation into a hierarchical arrangement of parts and to determine the highest priority variable that should be the outcome of the situation based on informed judgment. It is one of the useful methodologies for decision making and path selection.

In this work, there are three hierarchical levels used and listed as shown in Figure 3-3. The first level is called target hierarchy, which describes what the object is, the second level is called criteria hierarchy, within this level, there are five criteria to be considered simultaneously as a scenario for this task. The last level is called decision hierarchy, within this level there are only five models used to be selected as a final decision based on the analysis provided in the criteria hierarchy level.

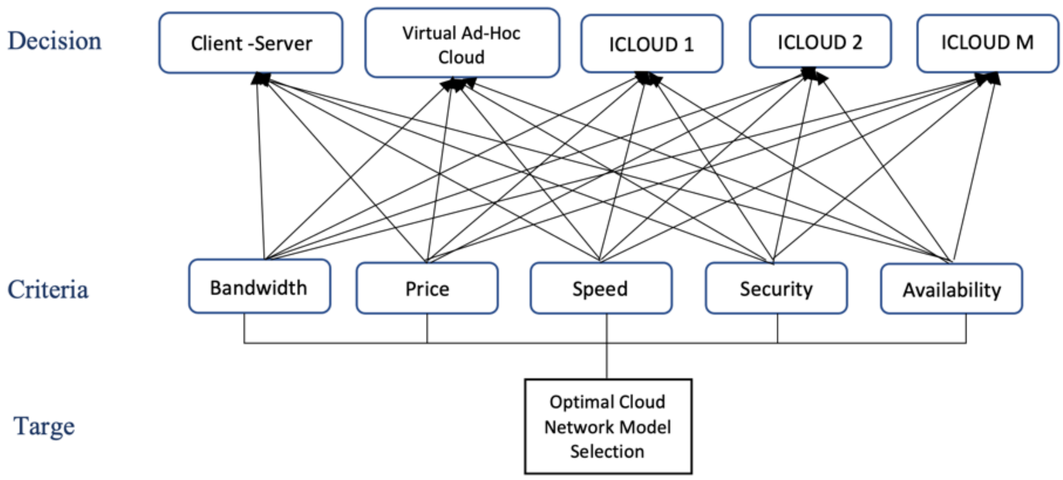

Figure 3-3 Cognitive decision hierarchy of HMCCNs model selection.

Bandwidth, price, speed, security, and availability have been chosen for this experiment as criteria to be considered in the offloading task to make the final routing decision. In Bandwidth, it depends on the type of wireless link between the mobile user and the cloud model established or selected. When the connection is very good, then the amount of data and computation that needs to be offloaded will be large, but if the connection is weak, then the amount of data computation is small [131]. The price is related to differences in costs for the same amount of computing. The measurement values are different from one cloud to another [132]. Speed is related to the cloud or the server's execution speed value for computational processes, sometimes it is measured with speedup factor (F) that represents the comparison between the execution speed of the cloud to the mobile device. For security, it is related to the level of security of how data are protected [133]. Availability is associated with the link status (failure) during the offloading process because of devices' mobility condition [134].

Six steps have been used to evaluate the weights of these criteria. The first step was to define the problem objectives. The second step was the decomposition process, which breaks the unstructured problem into a clear hierarchical structure. Then the third step was a pairwise comparison and formed the required comparison matrix. The fourth step used the eigen value method to determine the relative weights then checked consistency property, which was the fifth step. Finally, in the last step, an aggregation of relative weights is applied to measure the total performance of all the alternatives [135].

- **SDN Cognitive Controller System**:

The second approach proposed for cognitive network management and routing decision in HMCCNs is based on using a cognitive SDN controller as an intelligent system. This system presented optimizes the integration of cloud networks models. The system architecture proposed for cognitive SDN controller with HMCCN is shown in Figure 3-4.

In this proposed model, the cognitive SDN controller will work in the perceptions action cycle and provide input to the controller. In addition, based on the learned knowledge of the network over time, the controller will take action (decision) regarding routing and network data management.

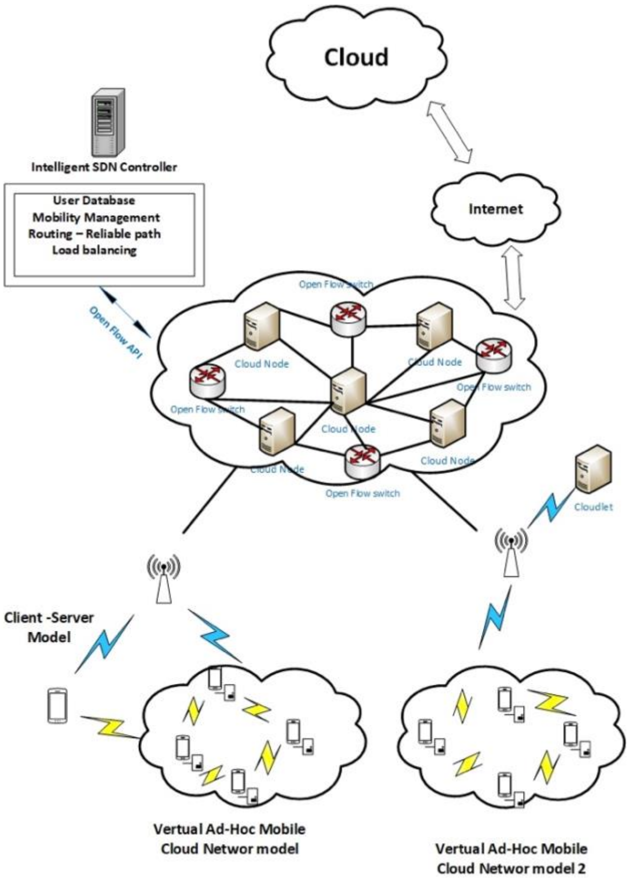

Figure 3-4 HMCCNs with Intelligent SDN controller.

A cognitive inspired system, based on cognitive control [136] for cognitive SDN-HMCCNs, is shown in Figure 3-5.

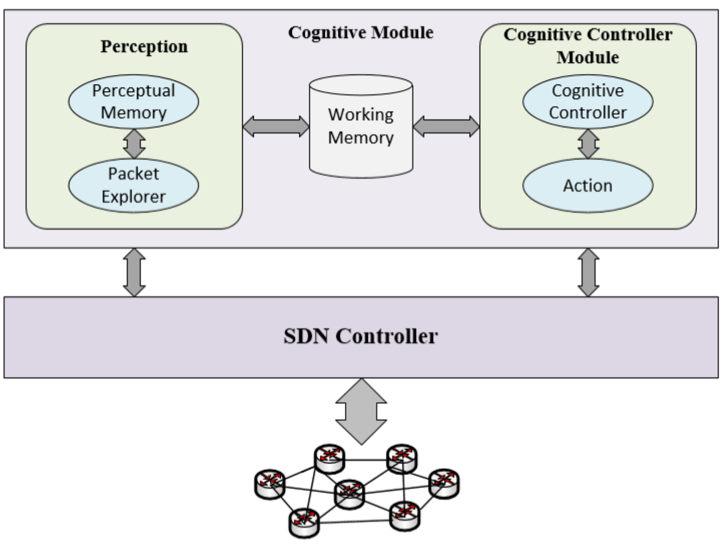

Figure 3-5 Cognitive System architecture for SDN -HMCCNs.

From the controller system illustrated above, there are three main components: The perception unit, which is responsible for perceiving and learning about the network's environment. The second part is a cognitive controller unit, which is responsible for taking actions. The last unit is the working memory that is supporting both previous units. These units form a closed-loop feedback mechanism (perception-action cycle).

In perception module, the SDN controller is connected and communicates to SDN switches using OpenFlow API [137]. The controller receives the control packets and the data from SDN switches. The control packets carries information like link status of link failure that may frequently occur in the network to cause latency and of information about network resources and of node mobility information. The perceptual memory extracts and keeps the learnt information that comes from incoming packets. The working memory sorts and analyses its meaningful information for decision making in the cognitive controller. A good example of useful information is knowing which link has frequently failed over time. Finally, the cognitive controller is responsible for making a decision based on the working memory unit's information, for example, finding a reliable path. With its centralised controller, this system also determines applications' requirements based on Quality of Service (QoS). Resending the packets from source to destination might be due to packets dropping (path congestion); thus, the cognitive controller can find delay on all paths using this mechanism.

In general, traditional network devices like routers, firewalls, switches, and other devices are made up of control plane and data plane. As a result, a huge effort is needed with a large network when performing devices configurations because it needs access through the command line for each device within the network. However, in an SDN network, the control plane and the data plane are separated. Thus, a control plane is managed using a centralized, programmable controller where all the policies are configured for each device within the network. In this work, a comparison of performance between SDN networks with traditional networks has been implemented based on calculating the average Round-Trip Time (RTT) with different numbers of switches used.

3.4 Simulation and Results

3.4.1 Integrated HMCCN Model Experiment

Using an OPNET version 17.5 as a toolkit to simulate HMCCN and based on LTE Network with bandwidth of 3MHz FDD. Different scenarios are implemented with the assumption that all cellular nodes within the network get similar resources such as (CPU speed, memory size and network interfaces). Figure 3-6 illustrates the result of an average throughput when sending uplink video streaming of 1 Gbps of data for 10 seconds of time from the source node MobA to the cloud server as a destination (Client-Server model) through the LTE network channel.

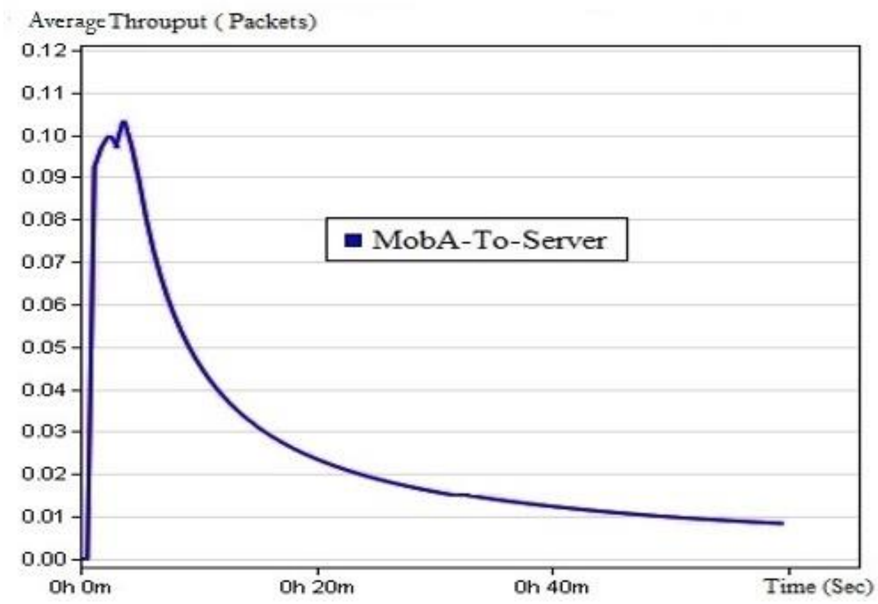

Figure 3-6 The average throughput in Client–Server cloud model with video streaming.

In the graph, it shows clearly that with the Client-Server model, the average network throughput value measured was low because of the congestion of data traffic between the source and destination, and also because of the limited capacity of the uplink channel used. This is considered a serious problem in real life especially when working with sensitive real-time applications. While in Figure. 3-7. it shows different scenarios with a worse case when MobA is disconnected from LTE base station, and the original task of video streaming has been partitioned in equal sizes and routed through (1, 2, 3, 4, 8) neighbour mobile nodes respectively to the server using virtual Ad Hoc network technique. In these scenarios, all neighbour mobile devices share their capacity links to upload the task. The new network performance results indicate that the average throughput increases each time increasing the number of mobile devices within the Ad Hoc network to transfer the data over Wi-Fi channels to the server through the LTE network. This new proposed model also improved energy consumption because the power and time required to send the data to each neighbour node using Wi-Fi links is less than the power and time needed to send the same data to the far cellular base station using only one link. Results also show that the value of the average throughput for uploading video streaming from MobA directly to the server through LTE network is almost identical to the value of average throughput of uploading the same video streaming from MobA using one hop Wi-Fi technique directed to other neighbour mobile nodes than to the server through LTE network. This is because the time spent on mobile devices for uploading process over LTE cellular network channel is much higher than the time spent when using Wi-Fi network channel. Finally, integrated a client-server model to an Ad Hoc virtual cloud model has minimised the utilisation of each node resources and its link capacity. Values are related to the number of nodes participating in forming an Ad Hoc virtual cloud network.

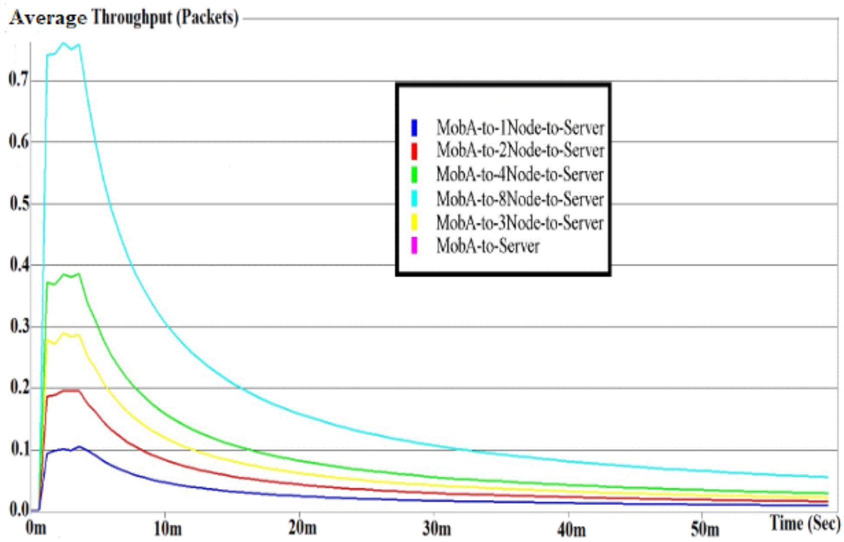

Figure 3-7 The average throughput in Heterogeneous MCN with high video streaming.

3.4.2 **SDN vs Traditional Networks Experiment**

These two experiments with all scenarios included were aimed to measure and compare the network performance in terms of network’s latency between SDN networks and traditional networks when their connected switches are gradually increased between the source node and the destination node using RTT values. In this experiment, two open-source simulation platforms have been selected. For the traditional network, Cisco Packet Tracer version 7.2.2.0418 was used, while for the SDN network, the Mininet platform was used and was installed on Ubuntu 18.4 platform.

- **Traditional Network Simulation**:

  In this experiment, all scenarios have been implemented using Cisco Packet Tracer platform. The network has a first scenario, which consists of two nodes, the source node host1 and the destination node host2. These two nodes connected through 4 switches and one router in the middle. When all system configurations are completed, the ping command has been used to send packets from host1 to host2 and the average value of RTT was calculated. In networking, the round-trip time represents the duration in millisecond measured for the signal to be sent from the source, plus the time it takes the acknowledging of that signal to be received. Further scenarios were implemented by increasing the number of switches gradually to (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) respectively, and the average values of RTT have been measured per each scenario. Figure 3- 8 shows the simulation scenario of 100 switches connected between host1 and host2.

Figure 3-8 Traditional network design with 100 switches.

- **SDN Network Simulation.**

  In this experiment, the same network concept previously used but with OpenFlow switches and SDN controller. Mininet was installed into the Ubuntu operating system. Unlike the Cisco Packet Tracer platform that can be described as a graphical user interface, the Mininet platform is a command control interface.

  The first scenario was implemented using two nodes, the source node host1 and destination node host2, these two nodes connected through 4 OpenFlow switches and one SDN controller. When all system configurations were completed, the ping command has been used to send packets from host1 to host2, and the average value of RTT was calculated. Further scenarios were implemented by increasing the number of OpenFlow switches gradually to (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) respectively, the average values of RTT have been measured per each scenario.

Figure 3-9 shows the scenario of 100 OpenFlow switches connected between host1 and host2 with the SDN controller in the middle.

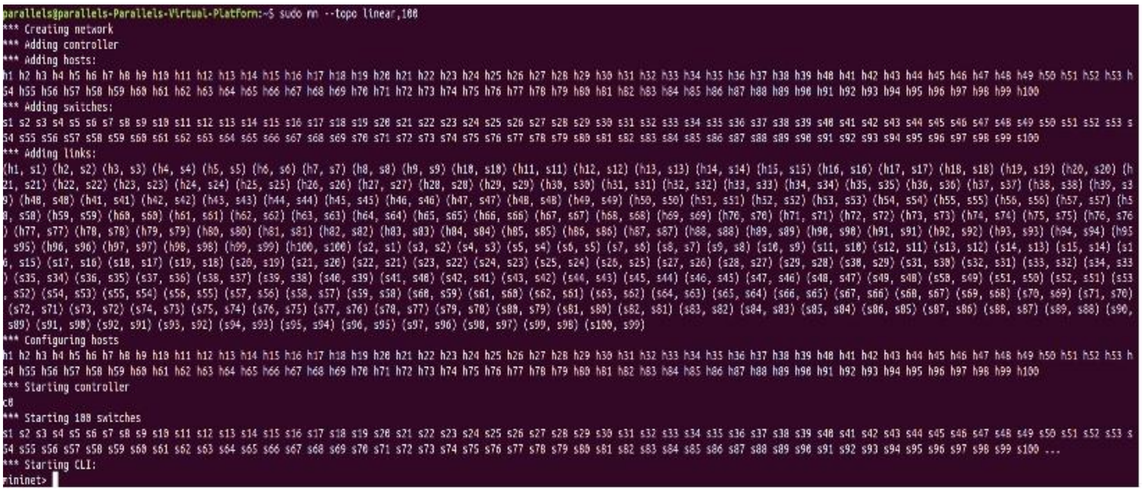

Figure 3-9 Mininet with 100 OpenFlow switches and SDN controller configurations.

All average RTT values for both experiments with all scenarios implemented were measured as shown in Table 3-1, and the bar chart of Figure 3-10 shows the average RTT values measured and differences for both experiments. Multiple readings (four readings) have been taken per each scenario to increase the accuracy of measurements taken from all scenarios. The mean value has been calculated as the actual average value per each scenario such that the measurement's errors will cancel each other out.

Number of Switches	Traditional Network Avg. RTT (ms)	SDN Avg. RTT (ms)
4	0	0.066
10	0	0.089
20	2	0.103
30	6	0.318
40	11	0.565
50	18	0.644
60	23	0.699
70	27	0.854
80	29	0.912
90	33	1.01
100	39	1.272

Table 3-1 Average RTT values for Traditional and SDN networks.

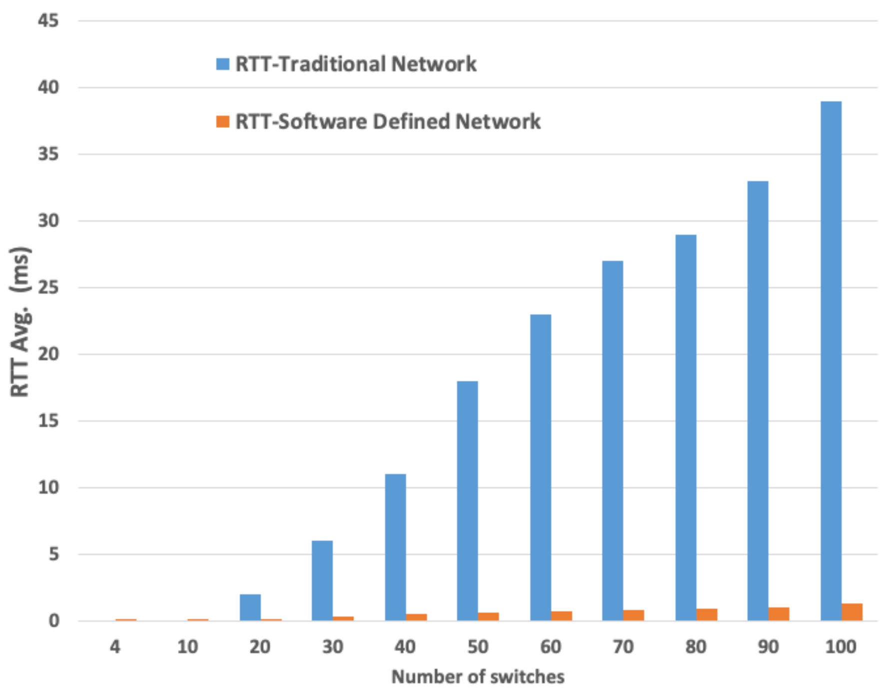

Figure 3-10 Comparison chart between traditional network and SDN network with respect to RTT.

The bar chart in the figure above illustrated that with the first two scenarios (4 and 10) switches, the average RTT values for the SDN network were almost similar or slightly higher than the average RTT values calculated for the same scenarios in a traditional network. The results show noticeable changes when the number of switches keeps increasing. Starting from the scenario with 20 switches, the average RTT values for the SDN network provided better results than the traditional network which its RTT values keep increasing till the last scenario with 100 switches.

As a result, the SDN network provided better RTT results (lower values) than the traditional network, especially with an extensive scale network that comprises too many switches. In the SDN experiment, when host1 sent the first packet to host2, it routed through OpenFlow switches that established a communication link with the SDN controller through OpenFlow protocol to check and decide how the current input flow should be processed. All other packets sent after that were routed directly to the source using the best route selected before. In traditional network experiment, the technique is different, each switch is forwarding packets based on the forwarding table. With the SDN concept, all managements of data packets are software-based, while the traditional network is hardware-based. In this case, SDN more flexible and can easily manage resources through the control plane.

3.5 Summary

In this chapter, an adaptive HMCCN model has been proposed to integrate different cloud network topologies and services in one workflow.

A task of high-rate video streaming has been simulated passing through an LTE MCCN from the source node to the server, the channel used suffered from low bandwidth, congestion, and link breakdown. The new adaptive model was established by integrating MCCN with VAMCCN to form the new model HMCCN.

The cognitive data offloading task and routing management methods were applied using two approaches: An FAH system as the first approach and the centralised cognitive SDN model as a second approach. Final results of the HMCCN model used in an experiment with different scenarios that were scalable in terms of the number of mobile Ad Hoc nodes, which shared their resources, shows an improvement in network reliability and minimising in mobile nodes energy consumption. The throughputs measured were increased each time more mobile nodes were added to the network. Finally, to prove the enhancement in network performance with

HMCCN that worked based on the SDN cognitive control network that was proposed as a second approach for routing management, two experiments were simulated to compare and analyse the differences between the SDN network and the traditional network RTT values in terms of network's latency. Final results with different scenarios, which were scalable in terms of the number of switches used within the two networks, have improved network latency when using an SDN network with higher number of switches than a traditional network.

Chapter 4

INTELLIGENT HYBRID ROUTING PROTOCOL WITH PATTERNS AWARENESS IN VIRTUAL MOBILE AD HOC NETWORKS

4.1 Introduction

Cloud computing networks is a modern networking generation that enables convenient, ubiquitous, and on-demand access to computing resources such as networks, servers, applications, storage and services with high scalability and less management effort. The great advantage and improvement of cloud computing to cloud users is the issue of mobile services. Mobile devices are becoming more efficient because of the contentious improvements in computing capabilities due to these devices' influence almost in every human's daily life. Despite that improvement, they are still constrained mostly due to the limited storage capacities, processing capabilities and short battery lives of the mobile devices, which compromise the Quality of Service (QoS) provided. To overcome these limitations, researchers have presented two new types of mobile cloud concepts, Mobile Cloud Computing Networks (MCCNs) [138][139][11], and Ad Hoc Mobile Cloud Computing Networks (AMCCNs) [140][141]. In mobile cloud computing, a cloud system is integrated with mobile devices based on an infrastructure communication network, of which a cellular network is a good example of such a network. The integration between the cloud system and mobile devices enables these devices to access huge storage space and processing power. It helps mobile devices to run intensive computational applications such as video and image processing on the same mobile devices. Furthermore, using cloud systems for data storage and execution for such applications improves reliability and minimises battery power consumption on mobile devices. In mobile Ad Hoc cloud computing, or sometimes called a virtual mobile Ad Hoc cloud computing network, it consists of multiple devices intercommunicating through mobile Ad Hoc networks to form a virtual supercomputing node model.

4.2 Virtual Mobile Ad Hoc Cloud Computing Networks (VMACCNs)

Virtual mobile Ad Hoc cloud computing networks can be considered as a collection of wireless mobile nodes that dynamically create a one-hop connection to the cellular network and also can work as a wireless network connected amongst themselves without using any infrastructure. Mobile cloud nodes are no longer just the end-user node. Each of them must also be able to function as a router to relay packets generated from another previous node(s). The growth of data traffic has witnessed a phenomenal increase in a mobile network.

Figure 4-1 shows the mobile network data traffic growth chart around the world made by the Ericsson Mobility Report Data and Forecasts. Growth increased sevenfold between the year 2016 to the expectation of growth in 2027 [142].

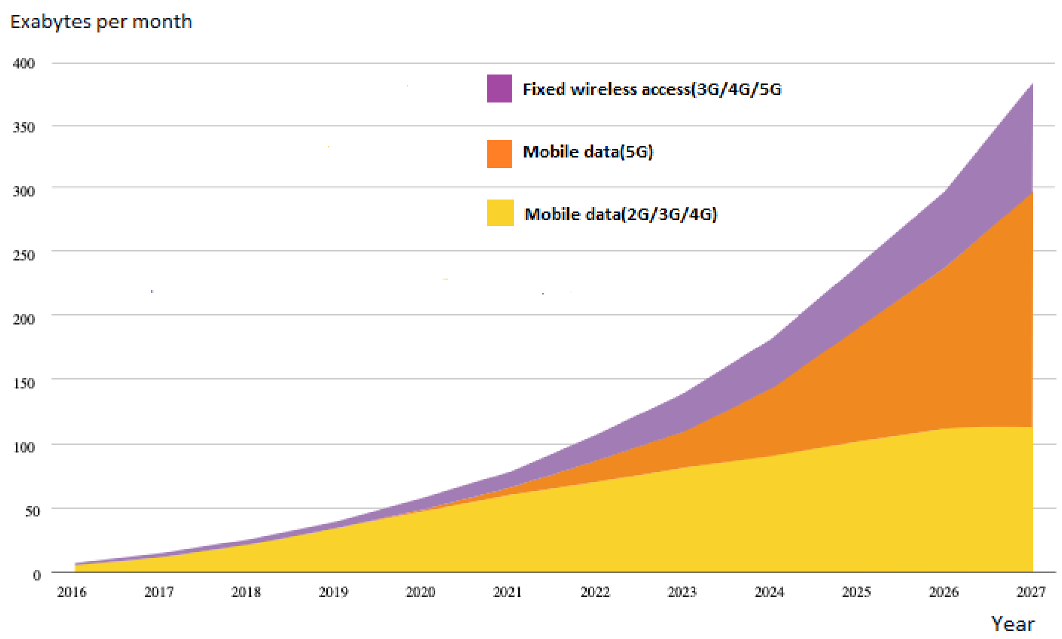

Figure 4-1 Global mobile network data traffic (EB per month) [142].

When the users work on their own mobile devices in movement conditions and with the limitation of networking and computing resources within the device, topology adaptation is required to make sure that nodes are communicated as appropriate. Furthermore, constraints reveal on how much control and administrative information need to be exchanged and for how many times. Using an effective routing algorithm protocol is one of the most significant challenges in mobile cloud networks.

4.3 Routing in (VMACCNs) Challenges

Most mobile multimedia and time-sensitive applications require the establishment of transmission paths that should meet several parameters such as bandwidth and delay, commonly referred to as QoS guarantees. In a virtual mobile Ad Hoc cloud network, due to low signal power, high mobility and limited bandwidth, an active path's link breakdown might occur when, for example, a pair of mobile nodes that are establishing link communication along the data transmission path exceed each other's transmission range. Each node that is forming the network can join or leave the network and is free to move in any direction and at any speed independently. Searching for a new path only occurs when the current path is broken. In general, the cost of detecting the failed link is high and several time-out packets sent before a path is considered as breakdown. Therefore, during a path breakdown period, packets experiencing loss and delays before the broken path is detected and an alternative new path is established. This may cause an increase in packet loss rate and a decrease in throughput. Unfortunately, route failures occur frequently in the virtual Ad Hoc cloud network because of mobile node mobility and limited battery capacity resources. The operations of route discovery also consume additional network resources, reducing network performance and lifetime. Many traditional Ad Hoc reactive routing protocols for mobile wireless networks use a single metric like signal strength or shortest path to building the route for data transmission. Routing metrics are an essential factor to select the best links. However, the single metric method for route selection is insufficient to build a stable route because it most probably causes frequent route failures that motivate routing protocol algorithms used to retransmit further packets to rediscover new paths each time the link breaks. Therefore, combining multiple routing metrics using an efficient, intelligent protocol to select the most reliable nodes and establish the best route starting from the source node to the destination node is essential [143]. Ad Hoc on-demand distance vector routing protocol (AODV) and destination sequenced distance vector, and Dynamic Source Routing (DSR) are examples of shortest path routing protocols [144]. These protocols use minimum hop count to determine the best route paths without considering node resources quality or the path's link stability factor. It means that the path selected for routing may not be reliable, especially for dynamic topology networks with time-varying radio link conditions. Despite that, other protocols have been proposed to overcome some of the weaknesses of shortest path routing algorithms like Sequence Distance Vector Routing [145]. However, overhead and extra processing is so common in these protocols.

To overcome these problems, it has to:

1- Predict link failure and nodes resource limitation values that affect data routing reliability such that an alternative routing path is selected before the established path for a connection breaks.

2- Increase the throughput, which is the amount of data that are passing through the network. Increasing throughput will decrease the waiting time of any data or control packets and will improve network performance.

4.4 Research Gap

Different algorithms and routing protocols have been proposed for VAMCCN and MANET in general. Three classifications of previous works have been overviewed and presented here in this section that is selected based on intelligent routing protocols used and types of metrics chosen to make the routing decisions.

4.4.1 Fuzzy Logic System

Computational intelligence techniques like fuzzy logic have been comprehensively used in different fields of control engineering and engineering research and provide a promising approach in routing algorithms. The author in [146] proposed a mobility management system in the wireless network supported by fuzzy logic techniques. The main target is to keep monitoring a mobile worker's movements and their condition inside the oil refinery. In this work, matrices of End-to-End packet loss and a threshold based on the Received Signal Strength Indicator (RSSI) have been measured and used instantaneously as state inputs of a Fuzzy Logic Mobility Controller (FLMC). Using linguistic rules that explain the behaviour of the environment in different conditions helps decide whether to hand off to another new position. In [147], the author proposed a fuzzy logic routing algorithm. The approach was to find a stable route from source to destination using three parameters: energy, trust value and reliability value. The decision to select the candidate path is related to its reliability value; results show an improvement in path reliability compared to AODV. In [148], a new evolutionary Ad Hoc on-demand fuzzy routing protocol has been proposed. This new algorithm used to determine the best route based on achieving various objective performance in MANETs. Different routing metrics such as signal strength between two neighbouring nodes, remaining node battery lifetime and node queuing length were used in this model to select the best route with less fuzzy cost value. The results have shown a noticeable improvement over conventional MANETs routing protocols.

4.4.2 Neural Network

In this section, a brief overview of the previous works proposed in routing techniques and traffic prediction based on neural networks. In [149], a new sensor routing table realisation protocol was presented using a neural network. In this model, the routing table is replaced by an artificial neural network to improve routing decisions speed. ANN also trained to change the routes in case of network topology changed. In [150] a novel QoS aware routing protocol is proposed to fulfil the requirements of delay-sensitive applications. The protocol successfully improved the PDR and throughput, which improved the performance of the network. In this model, ANNs used to predict end-to-end-delay to incorporate QoS enabled route discovery. The routing overhead is minimised in this model compared to the existing approaches. Some other research proposed different approaches using a neural network to address the designing challenges of location prediction systems. In [151], the authors proposed an STF-RNN model that used a recurrent neural network algorithm to predict the next location of individuals. The proposed model includes space and time interval sequence. This method was used to predict the long term dependencies that provided a significant improvement to that model's efficiency. The related works mentioned above show that the new scientific direction and research approaches are moving towards the developments and find a new architecture of applying neural networks to achieve intelligent adaptive routing algorithms.

4.4.3 Pattern Predication

With mobile node movements in Ad Hoc cloud networks, a rapid change occurs in network topology that causes frequent disconnection. To address the impact of mobile nodes' movements, we need to evaluate nodes' movement in the cloud networks. Mobile prediction is a method of assessing the route of the future position of the mobile nodes.

Studies have been made within this topic that covers different fields in wireless mobile networks such as Ad Hoc and cellular networks [152][153][154][155][156][157]. The mobility prediction techniques support to evolve an image of the future mobile network topology. As a result, it reduces the need for location updating information, leading to minimising communication delay and improving QoS. The better accuracy of the prediction depends on the regularity of mobile nodes movements. However, sometimes regular mobile nodes' movements act unpredictably. Recently, different prediction methods for mobile nodes have been proposed by many researchers. These methods varied in the information they used for mobility prediction, and the results of the parameters they obtained by prediction, and the target for prediction. The author of [158] proposed a sequential learning algorithm for human mobility as a solution for short-term prediction. For the prediction technique, a constant order Markov model has been used. Despite that, the accuracy of prediction is high due to large datasets of sequences used to predict human mobility. However, the mobility prediction process cannot be estimated without the accessibility of mobility history data. In [159], a new technique of Enhanced Localisation Solution (ELS) has been proposed. This solution combined the human mobility model, standard location tracking method and machine learning and is presented as a self-adaptive solution. The results show that this technique worked properly for different node reactions with a low error rate and high-power consumption. A lightweight genetic algorithm also being proposed as a new method for prediction works in [160] to improve routing in MANET. This approach of genetic predictor did not contain all genetic operations. In addition, it was modified to reach its termination condition with a smaller number of iterations. Unlike the probability-based techniques, genetic predictor causes better QoS results. However, when using this technique for prediction, heavy computational power was needed as well as more memory in mobile nodes storage required. A Mobility Prediction of a MANET node based on the Bayesian model has been proposed in [161]. This model improved routing protocol by preventing broadcasting request messages from high mobility nodes by using prediction results. GPS information has not used in this model. The improvements in packet delivery ratio reach 46.32% based on the maximum speed of 30m/s and the density of the nodes of 200 nodes/k. A mobility prediction for MNs' future locations based on neural learning machine was proposed in [162]. Two integrated architectures of Extreme Learning Machine (ELM) and standard Multi-layers Perceptron (MLP) were used as a solution in this method with higher accuracy achieved by an order of magnitude. Improvement in accuracy enhanced the quality of service in MANET. This model reduces data exchange inside the network by predicting routing tables in MANET; it also helped reduce battery power consumption. In [163], a method to predict the future location of the mobile user proposed. Based on considering user's online posts that were already tagged with GPS information (geological coordinates) of the user's smartphone. However, this method's prediction accuracy was low because the amount of information collected was limited.

A routing algorithm for Ad Hoc on-demand distance vector network AODV was proposed in [164] based on mobility prediction. In this algorithm, the measured mobility was used to estimate the lifetime of the link between nodes, and the estimated values (link durations) helped select the best route. Unlike traditional AODV routing maintenance, which used fixed duration values, this algorithm used mobility estimate when dealing with route maintenance. This proposed algorithm effectively decreased the number of control messages used for route discovery and route maintenance. Therefore, significant improvements have been shown on route performance like packet delivery rate and End to End Delay values. The author in [165] proposed a movement prediction model for MNs based on neural network algorithm. Feed-forward Neural Network with three layers was used with a backpropagation algorithm for the learning process. For the training and testing process to the neural network, patterns of Ad Hoc mobility nodes' locations based on the random waypoint mobility model have been applied as the input of a neural network. The outputs consisted of prediction of future position. The system's evaluation performance is applied by adding new input as a node position and calculating the output's accuracy as a future position. Prediction accuracy was acceptable with this model.

Many routing algorithms mentioned focus only on discovering a suitable route for transferring data packets through intermediate nodes. At the same time, little attention was given to improve that discovery by minimising overhead control packets during the discovery process. It is also essential to know and select the best timing for collecting the data required for routing decisions, such as the data used as an input for a fuzzy logic system. Updating all these data during task duration includes assuring the data values' accuracy is an important factor. Lastly, little attention has been given to know the best network/node location to decide the optimal route. Different approaches and algorithms were proposed, but not one of them manages to discuss all these factors combined in one system.

4.5 Proposed System

One of the most critical research areas in vertical Ad Hoc routing protocol is establishing and maintaining network connections through the routing protocol. Although there are so many traditional routing protocols available and proposed, this work used AODV as a traditional Ad Hoc routing protocol for the performance comparisons process with other new adaptive intelligent proposed protocols for its familiarity among all other protocols. To get realistic results when evaluating the performance of the new routing protocols proposed, the network's parameters should be assigned based on realistic conditions such as the initial mobile resources values used and the mobile nodes' movement model that will be simulated (Mobility model). Mobility models severely impact the performance evaluation of any assessment of a new routing protocol or any comparison with other corresponding routing protocol models presented [166]. Thus, it is certainly important to choose a suitable mobility model that correctly expresses a real movement pattern of mobile nodes in the network. As a result, all the proposed routing protocol experiments presented here are based on an Ad Hoc network with Random Waypoint Mobility Model (RWM). This is a well-known model and mostly used by many research as a realistic mobility model [167][168][169][13]. Results obtained from the network experiments were simulated in MATLAB and have been compared and analysed based on important metrics such as end-to-end delay, throughput, and packet delivery ratio. In conventional AODV routing protocol, the best route selection is decided by utilising the minimum value of hop metric [170], but this is not an adequate parameter for establishing the best route to a destination in MANETs in general. It does not consider the other essential factors that may affect the routing reliability and quality of service provided. In this work, the adaptive proposed protocols used important node parameters for route decision calculations; for instance, node energy capacity value and mobility (speed and position) values were considered to be the main parameters to establish a reliable route and to ensure minimising the probability of route failure at the time of data packet transmission. The selected reliable nodes that were used to build a reliable route in these proposed intelligent algorithms were based on nodes that were having higher residual battery energy value, moving with minimum speed, and located at a minimum distance from their neighbouring nodes, the direction of movements was also considered as another important factor that may increase the probability of link failure. In addition to AODV as the first traditional protocol presented in first experiment, three other proposed routing protocols that used a fuzzy logic algorithm to make routing decision were presented as three more experiments. A fuzzy logic technique was used to calculate each link's reliability values by combining residual battery energy value and velocity of each node with the distance between them in this network. The path with the highest reliable value is selected to establish the best route to the destination. Routing protocols, namely Fuzzy logic A, Fuzzy Logic-B, and Intelligent Hybrid Fuzzy-Neural protocol (IHFN), were proposed as three experiments. All experiments have been simulated based on an Ad Hoc network with a random waypoint mobility model. An improvement has been made when using the fuzzy logic concept in routing mechanism with the two experiments (Fuzzy logic-A protocol, and Fuzzy logic-B protocol) compared to AODV. Furthermore, it revealed two approaches to knowing the best time to collect nodes' parameters information from the nodes within the routing path selected and the best location to make a routing decision for better. In the last experiment, a novel intelligent hybrid routing protocol namely IHFN was presented that used fuzzy logic and neural network algorithms combined. The new protocol has used a pattern prediction concept based on a neural network system with feedforward propagation to make the required predictions. The dataset collected form the first experiment was used to train and test the NN system. The system then used in the IHFN routing protocol mechanism to predict the next future steps of nodes' essential parameters to use them later to get the links' reliability values calculated using the fuzzy logic system. Getting links measurements during dynamic topologies helped calculate the highest value used as the best routing path for data transmission from the source node to the destination node. Three different time steps predictions have been applied in three different scenarios within IHFN experiment. All results of all four experiments with scenarios applied were analysed and compared.

4.6 Implementation and Simulation

4.6.1 Core Network with (RWM) Model:

This section explains the simulation environment, including suggested relevant simulation parameter values. This network, namely the core network, will be used as a core for all other experiments by applying all traditional and new proposed routing protocols.

In this work, a virtual Ad Hoc mobile cloud network was simulated with all proposed algorithms using MATLAB R2020a 64-bit. The network consists of 60 Ad Hoc nodes moves according to the Random Waypoint Mobility Model (RWM) and two other nodes as a source and destination. The grid size selected (the arena of the simulation) was set to 600*600. During the simulation, the RWM mobility model creates positioning, speed, and energy patterns per each node. This information was collected and used as datasets for the prediction process. Parameters for the core network design were selected, as shown in table 4-1.

Parameters	
Mobility Model	RWM model
Area of deployment	600*600 m^2
Number of nodes	60
Simulation time	750 sec
Speed (max. and min.)	[0, 8] m/s
Pause time (max. and min.)	[0,1] sec
MNs Max. charging capacity 100%	4 Watts
Transmission Range	100 m
Packet size	CBR
Packet traffic rates	Low rate (200 packet/sec) Medium rate (600 packet/sec) High rate (1200 packet/sec)

Table 4-1 RWM network parameters.

The experiment started by distributing all of the 60 nodes to their randomly distributed initial position (x, y) within the simulation arena as shown in Figure 4-2, each one of the 60 nodes stayed in its initial location for a certain selected randomly period of time (pause time) between [0-1] sec, and as soon as this time comes to an end, each mobile node assigned new random destination within the arena of simulation and also assigned with it a randomly speed range between [0-8] m/sec. Upon arrival, the mobile node takes another pause for a specific random time period before doing the same process again until the simulation time of 750 sec. ended. Using the null matrix, all variables were initialized, and then all new values were ready to be collected and saved inside it. Parameters required to save are node positioning values (x, y), node's speed values, and residual battery energy values. The dataset created from saved information will be formed as Comma Separated Values (CSV formats) file. The simulation time has been selected based on its relationship with the datasets generated from the simulation. If the simulation time increases, then the datasets collected will also increase.

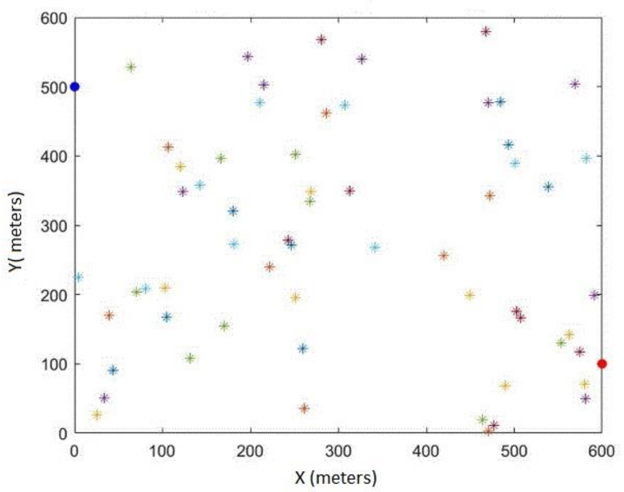

Figure 4-2 RWM simulation with 60 nodes with initial random positioning.

4.6.2 Experiment 1: AODV Routing Protocol

An Ad Hoc On-Demand Distance Vector routing protocol (AODV) is one of the most favourite traditional reactive routing protocols used in the research environment [171]. With the AODV protocol, the source node starts to initiate the route discovery process when there are data packets to be sent, and it is broadcasting a Route Request (RREQ) packet to all the neighbour nodes within the transmission range. Each intermediate node receives the RREQ check if it has a new route to the destination node or if it is the destination. Then, it responds back in a unicast way using a Route Replay (RREP) packet to the source node. AODV is one of the traditional protocols that use the minimum hop count (Short Path-SP) parameter to select the best route to the destination node regardless of the nodes' attributes that will be formed in the route. When the source node receives different RREP Packets, RREP with the shortest hop count is selected. In case a route link failure occurs, the node attached to that faulty link will create a Route Error (RERR) packet that travels back to the source node. Source node starts to rediscovery process for a new route if still needed, or if any other data packets still need to be sent. When the source node is broadcasting the RREQ packet during its discovery process phase, the intermediate node rebroadcasts each received RREQ packet (if this intermediate node is not the destination or does not have an updated route to the destination) following incrementing the hop count parameter value by one. When the intermediate node receives different RREQ packets with the same identification number (Id) and same sequence number with different hop count values, each from its neighbouring nodes, the node checks each RREQ packet separately. If the new hop count value is smaller than for the previously received RREQ packet with the same (Id), the node will then update the value of its hop count for its reverse route with that specific Id and rebroadcasts the RREQ packet. Unless the node already got an equal or lower hop count value, then the RREQ packet will be discarded.

By using the same core network RWM model simulated previously. In this experiment, the maximum charge capacity of each node was assumed to be 4 watts each. Each node is assigned various randomly initial energy values such that mobile nodes numbers from 1 to 20 were assigned randomly with residual mobile battery energy values between 0 and 50% of its maximum charging capacity, while mobile node numbers from 21 to 40 were assigned randomly with residual mobile battery energy values between 25 to 75% of its maximum charging capacity. Finally, node numbers from 41 to 60 were assigned randomly with residual mobile battery energy values between 50 to 100% of its maximum charging capacity. Figure 4-3 illustrates how the battery charging capacity values were assigned randomly as explained for all 60 nodes. The experiment started with RWM movements for all 60 nodes, three scenarios applied by sending data traffic packets of 200, 600 and 1200 packets /second respectively from the source node to the destination node using AODV routing protocol for a duration of 750 seconds each. Metrics such as end-to-end delay, throughput and packet delivery ratio have been measured for all three scenarios. The values of each node's positioning, speed and residual battery energy were saved for the whole simulation time (750 sec) for the three data traffic scenarios. Datasets were saved in Comma Separated Values (CSV formats) files.

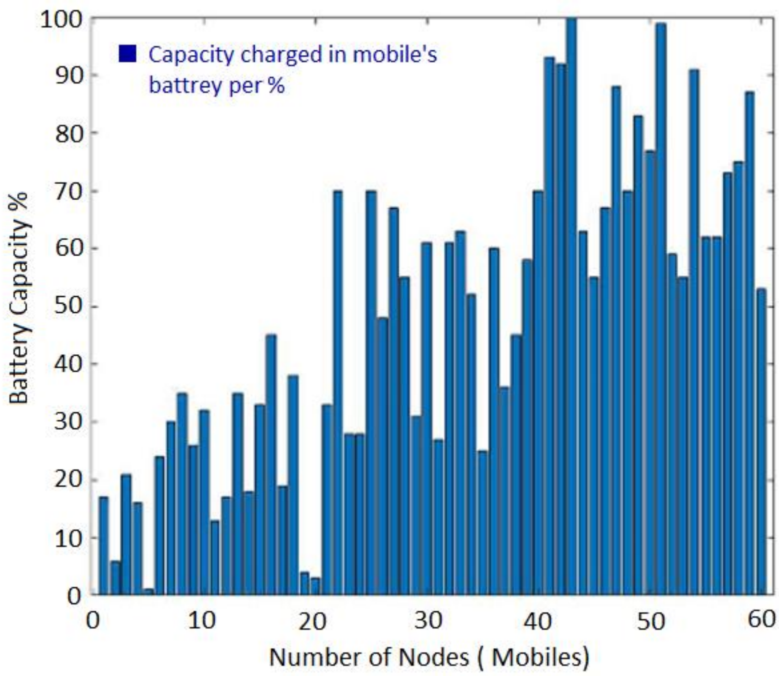

Figure 4-3 Residual battery energy values randomly assigned for 60 nodes.

4.6.3 Experiment 2: Fuzzy logic-A Routing Protocol

In a mobile Ad Hoc network, the communication between one-hop nodes (neighbour) needs the relative information of nodes' movements. In general, the mobile node state comprises the movement speed, position, and movement direction [172]. In this experiment, MN_i (p,v) represents the attribute description of one mobile node, where i denoted the number of one mobile node, p_i represents the position formatted in (x_i, y_i) , v_i represents the velocity of MN_i, velocity is a vector that includes the value (speed) and the direction by assuming we have two adjacent mobile nodes MN_i and MN_j in motion as shown in Figure 4-4,

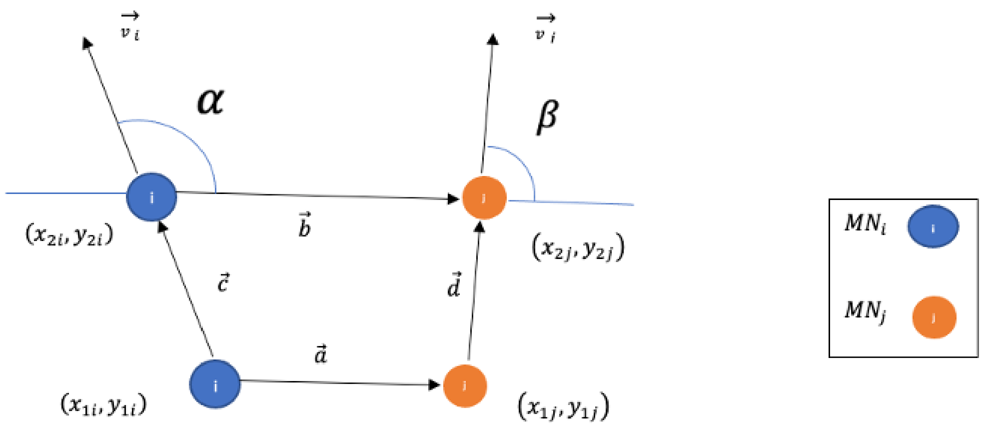

Figure 4-4 Two adjacent (one hop) mobile nodes in movement state.

Here Δd_{ij} is given by the following equation:

$$\Delta d_{ij} = P_i - P_j \tag{4-1}$$

$$\Delta d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{4-2}$$

Here, Δd_{ij} represents the distance between mobile node i (MN_i) and mobile node j (MN_j) .

Also, Δv_{ij} has been calculated as shown

$$\Delta v_{ij} = (v_i \cos\alpha - v_j \cos\beta) - (v_i \sin\alpha - v_j \sin\beta) \tag{4-3}$$

Here, Δv_{ij} represents the velocity differences between mobile node i (MN_i) and mobile node j (MN_j). Also α represents the angle between v_i and the extended line that connected between MN_i and MN_j and it has been calculated as follows:

$$\cos\alpha = \frac{\vec{b}.\vec{c}}{|\vec{b}||\vec{c}|} \tag{4-4}$$

$$\cos\alpha = \frac{(x_{2j}-x_{2i})(x_{2i}-x_{1i})+(y_{2j}-y_{2i})(y_{2i}-y_{1i})}{\sqrt[2]{\left[(x_{2j}-x_{2i})^2+(y_{2j}-y_{2i})^2\right][(x_{2i}-x_{1i})^2+(y_{2i}-y_{1i})^2]}} \tag{4-5}$$

$$\alpha = Arccos \tag{4-6}$$

Same calculation is used to get the value of β as follows:

$$\cos\beta = \frac{\vec{b}.\vec{d}}{|\vec{b}||\vec{d}|} \tag{4-7}$$

$$\cos\beta = \frac{(x_{2j}-x_{2i})(x_{2j}-x_{1j})+(y_{2j}-y_{2i})(y_{2j}-y_{1j})}{\sqrt[2]{\left[(x_{2j}-x_{2i})^2+(y_{2j}-y_{2i})^2\right]\left[(x_{2j}-x_{1j})^2+(y_{2j}-y_{1j})^2\right]}} \tag{4-8}$$

$$\beta = Arccos \tag{4-9}$$

Here, β represents the angle between v_j and the extended line that is connected between MN_j and MN_i

The attribute of Δv_{ij} results in equation (4-3), which denotes that if the value (can be taken from speed) and direction of the velocity vector of two mobile nodes are equal, then the value of Δv_{ij} is zero, while in the case of that v_i and v_j are face to face then the value of Δv_{ij} is positive. Finally, in case that v_i and v_j are in opposite direction, then the value of Δv_{ij} is negative.

- **Fuzzy Logic System Design:**

  Fuzzy logic brings the user's preferences and experiences into the decision by using fuzzy rules and membership functions. In this experiment, the Fuzzy logic controller has designed with three input variables and one output variable, where the three input variables to be fuzzified are residual mobile battery energy value (Energy) for the receiving node, distance (Δd) calculated from equation (4-2), and velocity (Δv) calculated from equation (4-3). A crisp value of Reliability Value (RV) is the output from the system after defuzzification. The Linguistic variables related with the three input variables are (Low), (Medium), and (High) for energy, (Low), (Medium), and (High) for distance Δd and (Negative), (Zero), and (Positive) for velocity Δv. For the output variables of the reliability value (RV), six linguistic variables have been used, (Very low), (Low), (Medium), (Average), (High), and (Very High). All Membership Functions (MFs) used in this work are chosen to be triangular membership functions for simplicity, convenience, efficiency, and speed [173]. Table 4-2 shows the ranges of inputs and outputs variables against each MFs.

Variables Names	Membership Function Name	Range (Lower limit, High limit)
Energy (Residual mobile battery energy value)	Low	0 to 50
	Medium	25 to 75
	High	50 to 100
Distance (Δd)	Low	0 to 50
	Medium	25 to 75
	High	50 to 100
Velocity (Δv)	Negative	-16 to 0
	Zero	-7.5 to 7.5
	Positive	0 to 16
Reliability value (RV)	Very low	0 to 20
	Low	0 to 40
	Medium	20 to 60
	Average	40 to 80
	High	60 to 100
	Very high	80 to 100

Table 4-2 Names and ranges of all MFs for each variable.

In Table 4-3, It shows fuzzy logic rules that have been used in this experiment for the proposed routing algorithm to calculate the output reliability values, where the first rule can be explained as, if Energy is (Low), Distance is (Low) and velocity is (Negative) then the Reliability value will be (Low).

Rules #	Energy	Distance (Δd)	Velocity (Δv)	Reliability value (RV)
1	LOW	LOW	NEG	LOW
2	LOW	LOW	ZERO	Average
3	LOW	LOW	POS	Medium
4	LOW	MEDIUM	NEG	Very Low
5	LOW	MEDIUM	ZERO	Medium
6	LOW	MEDIUM	POS	Low
7	LOW	HIGH	NEG	Very Low
9	LOW	HIGH	ZERO	Low
10	LOW	HIGH	POS	Very Low
11	MEDIUM	LOW	NEG	Medium
12	MEDIUM	LOW	ZERO	High
13	MEDIUM	LOW	POS	Average
14	MEDIUM	MEDIUM	NEG	Low
15	MEDIUM	MEDIUM	ZERO	Average
16	MEDIUM	MEDIUM	POS	Medium
17	MEDIUM	HIGH	NEG	Very Low
18	MEDIUM	HIGH	ZERO	Medium
19	MEDIUM	HIGH	POS	Low
20	HIGH	LOW	NEG	Average
21	HIGH	LOW	ZERO	Very High
22	HIGH	LOW	POS	High
23	HIGH	MEDIUM	NEG	Medium
24	HIGH	MEDIUM	ZERO	High
25	HIGH	MEDIUM	POS	Average
26	HIGH	HIGH	NEG	Low
27	HIGH	HIGH	ZERO	Average

Table 4-3 Fuzzy logic Rules.

In Figure 4-5, the input variables of MFs and the output with ranges selected and simulated using MATLAB platform is illustrated, also Figure 4-6 shows the relationship of Distance and Energy as fuzzy input variables for the proposed routing algorithm that are laid on the horizontal axes of the diagram with respect to the Reliability Value (RV) as an output variable in the vertical axis. Figure 4-7 shows the relationship of Velocity and Distance as fuzzy input variables for the proposed routing algorithm that are laid on the horizontal axes of the diagram with respect to the Reliability Value (RV) as an output variable in the vertical axis. Finally, Figure 4-8 shows the relationship of Energy and Velocity as fuzzy inputs variables for the proposed routing algorithm that are laid on the horizontal axes of the diagram with respect to the Reliability Value (RV) as an output variable in the vertical axis.

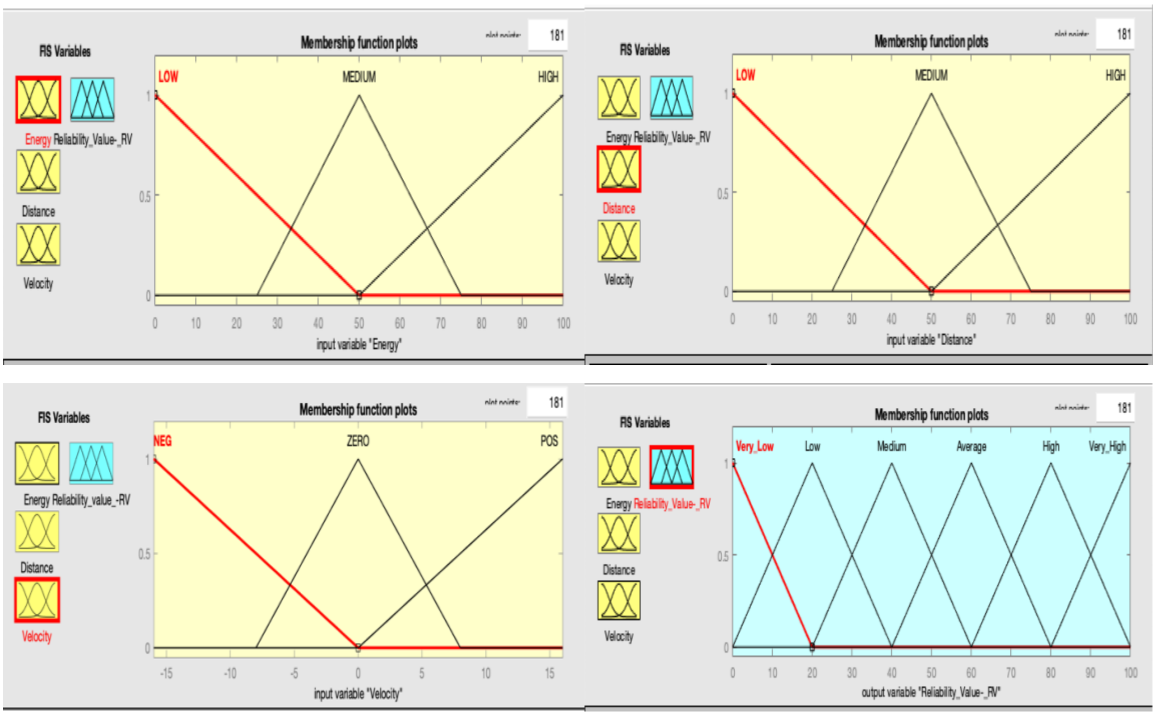

Figure 4-5 Input and output MFs variables ranges simulated in MATLAB.

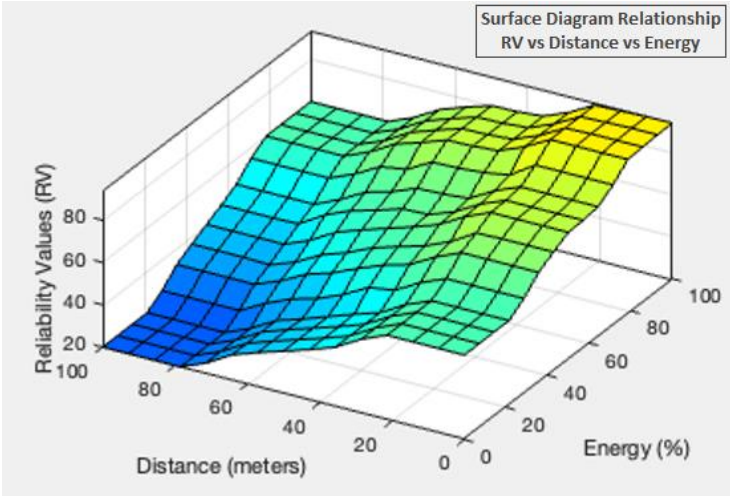

Figure 4-6 Reliability value (RV) with respect to Distance and Energy.

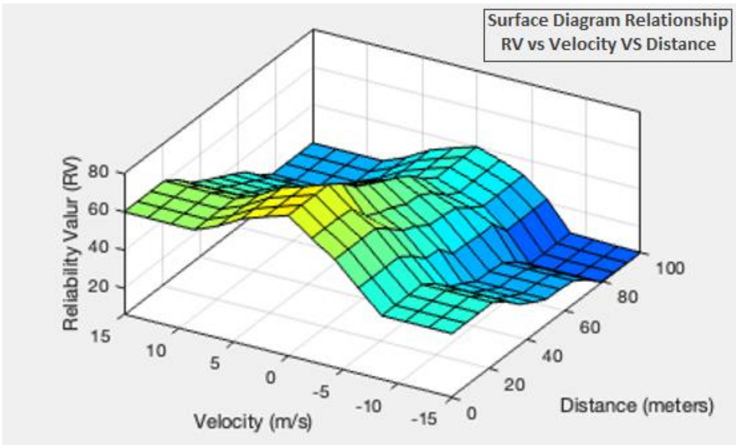

Figure 4-7 Reliability value (RV) with respect to Velocity and Distance.

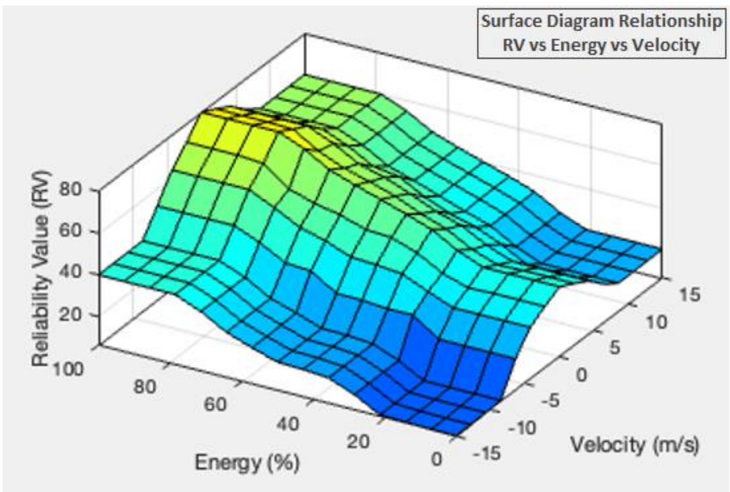

Figure 4-8 Reliability value (RV) with respect to Energy and Velocity.

- **Route Discovery Steps:**

This is based on the core network with RWM model and with battery charging capacity values assigned per each node and by assuming the source node does not have a route to the destination node. Route discovery is started when the source node initiates (broadcast) route Request Packet (RREQ) to all neighbouring nodes. Each node in this network, upon receiving the RREQ packet, rebroadcasts the packet to its neighbours' nodes if the node is not the destination node. At the destination node, as a first step proposed in this experiment, the RREQ packet was designed to carry three new fields of information that has been added to its field structure format. The new three information that are gathered along the complete path that been traversed consists of energy (Residual battery energy value), positioning and speeds values of the nodes.

A waiting time period (time window) is set up to start when the first PREQ packet is received at the destination node. Furthermore, more PREQ packets from the source node passing through different route paths may also receive at the destination node. That waiting time has been set to three RREQ packets received (three successful paths), in other words, the destination node takes only first three RREQ packets for calculation purposes and discards all others that come after. Then, as a second step, by extracting and using the information on each of these three different RREQ packets, Δd, ΔV, energy, and then RV values between each two neighbouring nodes along the whole path from source to destination is calculated using a fuzzy logic controller. When this calculation is completed as a crisp value, the reliability value $RV_{S,d}$ is calculated as a final value from source to destination per each successful path from RREQ packet such that:

$$\mathrm{RV}_{S,d} = (\mathrm{RV}_{s,1} + \mathrm{RV}_{1,2} + \mathrm{RV}_{2,3} + \ldots + \mathrm{RV}_{n,d}) \tag{4-10}$$

In equation (4-10), reliability value $RV_{S,d}$ represents the crisp value of the whole route between source and destination. Now, by comparing the three final values of the three-routes that have been received at the destination, destination node sends a reply packet RREP to the source node along the path with the maximum value of $RV_{S,d}$ calculated. Source node uses this path to send data packets to destination as a source router method. Three scenarios are applied in this experiment by sending data packets traffic rates of 200, 600 and 1200 packets /second respectively from source node to destination node using (Fuzzy Logic – A) routing protocol proposed in this experiment for 750 seconds each. Metrics such as end-to-end delay, throughput and packet delivery ratio has been measured for the three scenarios within this experiment.

4.6.4 Experiment 3: Fuzzy Logic–B Routing Protocol

In this experiment, a different approach has been proposed to select reliable and efficient routing paths for data packets. The new method proposed in this experiment is concentrated on using the same fuzzy logic controller, same configurations, and rules but with changes in the concept of when the three new information parameters of each node will be collected, and where the decision of the best routing path is made and how it is made.

Once again, the experiment is based on (core network with RWM simulation model) with battery charging capacity values assigned per each node as per previous two experiments and by assuming a source node does not have a route to the destination node. Route discovery is started when the source node initiates (broadcast) Route Request (RREQ) packet to all neighbouring nodes until packets arrive at their destination node. Upon receiving the RREQ packet, each node in this network rebroadcasts the packet to its neighbours' nodes if it is not the destination node. Unlike (experiment 2 with Fuzzy Logic-A), when the destination node receives different RREQ packets, the time window is started from the first arrival of the RREQ packet. In this experiment, the destination node sends a Route Replay (RREP) packet per each received RREQ packet immediately without any delay. The time window here is also counted by three RREQ that are successfully received by the destination node. Unlike (experiment 2 with fuzzy logic A), RREP packets are newly formatted by adding three new fields named Energy, positioning, and speed. All intermediates' nodes add their energy capacity values and their positioning and speeds to RREP packets and forward that packet towards the source node in the same direction but with a reverse route all the way across the selected RREQ packet path. Therefore, the source node receives the RREP packet, which includes the whole route topology information. Unlike (experiment 2 with Fuzzy Logic-A), the source node calculates all information received of Δd, ΔV, energy, and then RV values between each neighbouring node along the whole path from source to destination using fuzzy logic controller designed in previous experiment with same rules and configurations. When this calculation is completed as a crisp value, the reliability value $RV_{s,d}$ is calculated using equation (4-10). While receiving the first RREP packet and starting the process to calculate its $RV_{s,d}$, the source node transmits data packet immediately from that discovered path to the destination node. When the next following RREP packet is received, similar calculation is started to get its $RV_{s,d}$. Source node compares their $RV_{S,d}$ with the transmitted packet route $RV_{S,d}$, and if the source node finds the new route with higher $RV_{S,d}$, the source node switches the transmission of data packet path to the new next reliable path. Same process is applied for the third RREP packet received. Three scenarios are applied in this experiment by sending data packets traffic of 200, 600 and 1200 packet /second respectively from source node to destination node using (fuzzy logic – B) routing protocol method for 750 seconds each. Metrics such as end to end delay, throughput and packet delivery ratio has been measured for the three scenarios within this experiment.

4.6.5 Experiment 4: IHFN (Fuzzy Logic-B with Neural network)

An adaptive improvement in routing method is proposed in this experiment by using hybrid routing protocol system based on combining fuzzy logic and neural network algorithms. Three steps have been used to design route discovery procedure with decision making mechanism: -

- **Step 1- Simulate and Train ANN:**

  In general, an Artificial Neural Network (ANN) with its learning and generalisation capability acts as an appropriate tool to predict the mobile node position [174]. In this experiment, a feedforward neural network was selected and used to predict the future node motion and other essential parameters within its dynamic environment. The training and testing datasets used in this system was obtained from the datasets generated from the previous experiment. A total of (750 * 60) * 3 patterns has been used in this work (80% for training, 10% for validation and 10% for testing). The Neural Network (NN) prototype system used for prediction is shown in Figure 4-9. This figure illustrates the nodes' four parameters, which are position (x, y), speed, and energy as inputs, and the next timing step(s) of the nodes' four parameters which are position (x, y), speed, and energy as outputs.

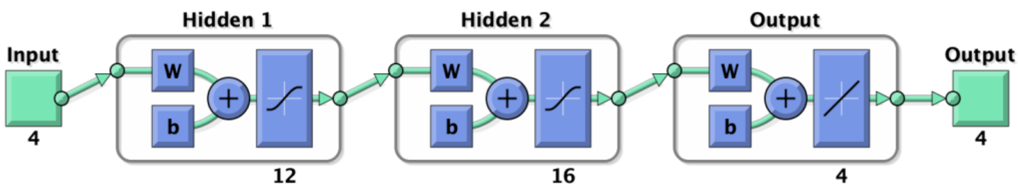

Figure 4-9 Neural Network prototype for pattern prediction.

The numbers of Hidden1 and Hidden2 of NN layers are selected by trial and error from this design. It has also been found that a small variation of numbers of hidden layers can affect the output prediction accuracy. The activation functions selected of a node was sigmoid function. The assumptions used for the inputs is the current values of position x(t), y(t), speed(t) and energy(t) to predict the next step(s) as an output such as x (t+n) y (t+n), speed (t+n), and energy (t+n), where n represents the prediction time step. The training process consists of 96 epochs; an epoch represents one process cycle in the neural network. The batch size for training is 3000*3000. Figure 4-10 presents the degradation of mean squared error during a training phase. The iterations that was required for validation performance to reach a minimum value was 90 epochs.

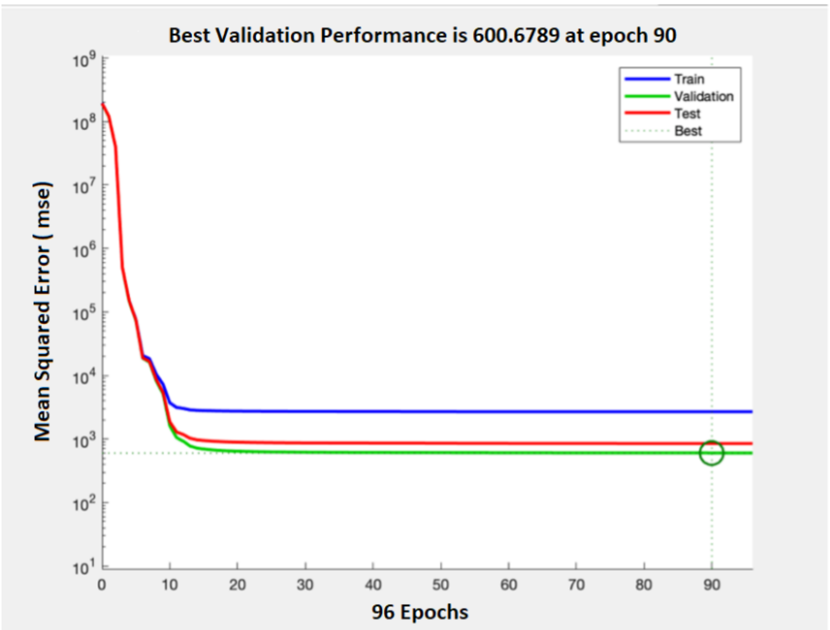

Figure 4-10 The mean squared error during the training.

Figure 4-11 shows network regression values during training the system by the training set, validation set, test set, and the actual prediction values. The R values indicate the correlation between the outputs values and the input values (relationship). The best value of R is when R=1 m. It means then there is an exact linear relationship between the output's values and the target's values. But when R is equal or close to zero, then there is no linear relationship between the output's values and the target's values.

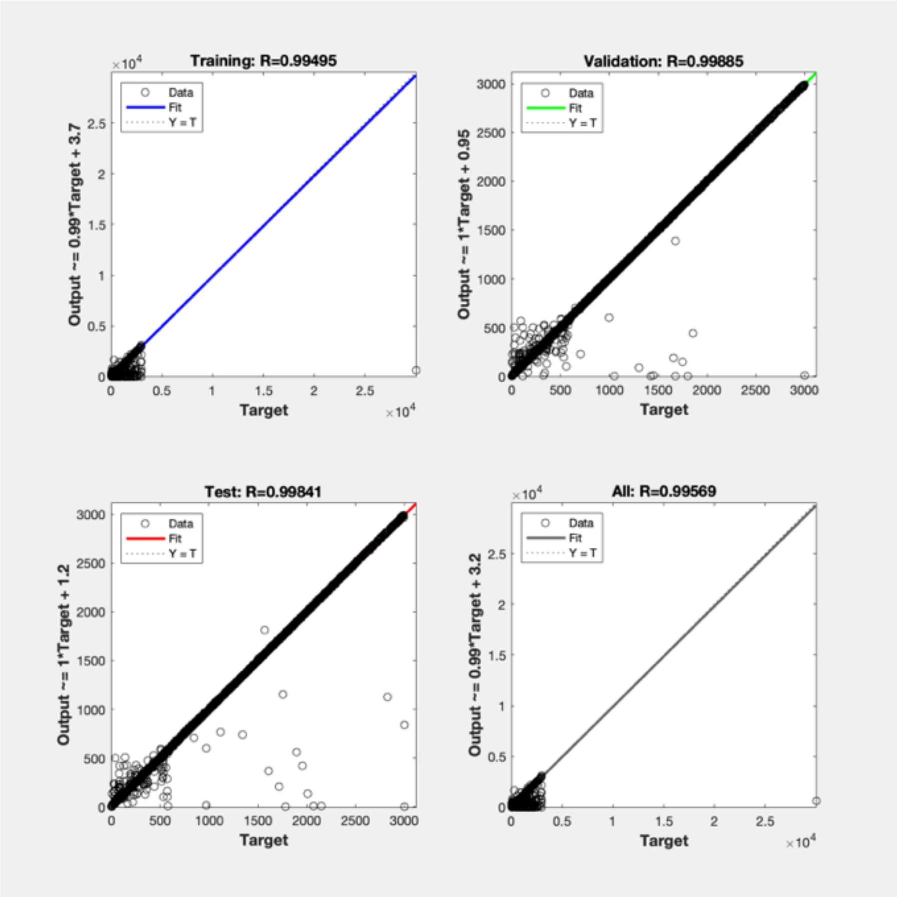

Figure 4- 11 The values of network regression for training set, validation set, testing set and all datasets.

- ### Step 2-Route Discovery Procedure:

The new adaptive routing protocol proposed in this experiment concentrates on adding future mobility and energy awareness to routing decision mechanism by using neural network pattern prediction technique and fuzzy logic algorithms together as one intelligent hybrid routing algorithm. The experiment is based on the use of the core network RWM model simulated with residual mobile battery energy value assigned per each node as per the previous three experiments. By assuming the source node does not have a route to the destination node, the route discovery is started when source node initiates (broadcast) a Route Request (RREQ) packet to all of its neighbouring nodes until packets arrive at their destination node. Each node in this network, upon receiving the RREQ packet, they rebroadcast the packet to its neighbours' nodes if the node is not the destination node. At the destination nodes, when they receive different PREQ packets within a time window that started from the first arrival of the PREQ packet, the destination node sends a route replay packet (RREP) per each RREQ packet received without any delay. The time window here in this work is also counted by three RREP that are successfully received at the destination node. The RREP packets are newly formatted just like previous experiments by adding three new fields named Energy, positioning, and speed. All intermediates' nodes add their residual mobile battery energy values, positioning, and speeds to the RREP packet. Intermediate nodes forward the RREP packet towards the source node using the same route but with a reverse direction all the way the selected RREQ packet passed through. Therefore, the source node receives the RREP packet that includes the whole route topology information. Unlike (experiment 3 with Fuzzy Logic-B), the source node uses a neural network system. The four parameters (Energy, speed, and position (x and y)) data belong to all nodes that have been measured within the route path carried by the first RREP packet received is the input to the neural system. The output values of the neural system that represents the predicted next timing step(s) of the (position, speed, and energy) values, is used to calculate the values between each two neighbouring nodes (links) for the whole path by using fuzzy logic controller designed in previous experiments with same rules and configurations. Next, by using equation (4-10), $RV_{s,d}$ is calculated as a crisp value. While receiving the first RREP packet, the source node transmits data packets immediately from that discovered path to destination node. The same procedure is applied when receiving the next following RREP packet by passing all the four parameters' information to ANN system for prediction process before $RV_{s,d}$ calculated , the source node will compare the new $RV_{s,d}$ with the transmitted packet route $RV_{s,d}$, if the source node find a route with higher $RV_{s,d}$, it will switch the transmission of data packet path to that new reliable path. Three scenarios are applied in this experiment by sending data traffic of 200, 600 and 1200 packet /second respectively from source node to destination node using the proposed intelligent hybrid routing protocol for 750 seconds each.

- **Step 3- (NN1), (NN2) and (NN3) Prediction Values:**

In Step- 2 experiment, the output values from the neural network system for all four parameters (Energy, Speed, and position (x, y)) have been set in advance to predict the first-time step (t+1), i.e., n=1 and it has been used later to calculate $RV_{s,d}$ with higher values to complete the routing process. This step was namely as NN1, which represents the first scenario in this experiment. The second scenario namely NN2 was applied using the second prediction time step (t+2)), i.e., n=2 and it has been used later to calculate $RV_{s,d}$ with higher values to complete routing process. Finally, the third scenario applied namely NN3, it was applied using the third prediction time step (t+3)), i.e., n=3 and it has been used later to calculate $RV_{S,d}$ with higher values to complete routing process.

The three scenarios within this experiment were implemented using data traffic rates of 200, 600, and 1200 packet /second respectively transmitted from the source node to the destination. Metrics such as end-to-end delay, throughput, and packet delivery ratio have been measured for the nine scenarios within this experiment.

4.7 Simulation, Results and Discussion

Performances of different routing protocols proposed under the same realistic mobility environments with different data packet rates have been evaluated using various quantitative metrics. In this work, popular performance evaluated metrics have been used for wireless Ad Hoc virtual cloud network routing protocols such as (throughput, end-to-end delay, and packet delivery ratio. The three measurements pause time (300, 500 and, 700 seconds) with respect to the simulation total time of 750 seconds have been selected to present and compare mostly the average results between all algorithms proposed.

- **Average Throughput:** was calculated as the percentage of the quantity of data being sent / received by the unit of time. The effects of each of the four experiments and scenarios included (AODV, Fuzzy logic-A, Fuzzy logic-B, IHFN with (NN1, NN2, and NN3 scenarios) on network performance using the average throughput metric with three various data packet rates of 200, 600, and 1200 packet/sec applied to the networks are shown in Figure 4-12, Figure 4-13, and Figure 4-14 respectively. The routing protocol of IHFN with proposed scenarios NN3 and NN2 have the highest throughput values. The new protocol with motion and energy predictions mechanism added further enhancements to calculate the best reliable path for data routing by increasing the accuracy of predicted networks future state in dynamic topologies. The more accurate calculations for reliable paths used have led to more packets going through the network, which shows better performance. The results also show the effectiveness of the other proposed routing protocols. The throughput of the AODV protocol have the lowest values because it has been affected by movements of nodes that changed their positions and speeds randomly during the whole simulation time, which frequently caused link failures. In addition, the limitation of battery energy values of each node and battery draining condition during the simulation time have also played a prominent role in the throughputs values results especially in AODV protocol that its routing selection mechanism does not consider this essential parameter, which was related directly to all nodes responsible for forming the best routing path. Throughput results show also some differences and better improvements when using the parameters' prediction values with higher timing steps. The throughput value of the fuzzy logic-B protocol shows an average of 5% higher than the throughput value of the Fuzzy Logic-A protocol. These throughput improvements were caused by increasing the accuracy of measurements by collecting the updated values of node parameters during collection by switching from RREQ packets to PREP packets. These improvements are also caused by changing the location of routing decision analysis and technique used from the destination node to the source node. In conclusion, the new IHFN routing protocol with its NN3 scenario has improved network throughput with an average of 20% compared to the traditional AODV routing protocol throughput value.

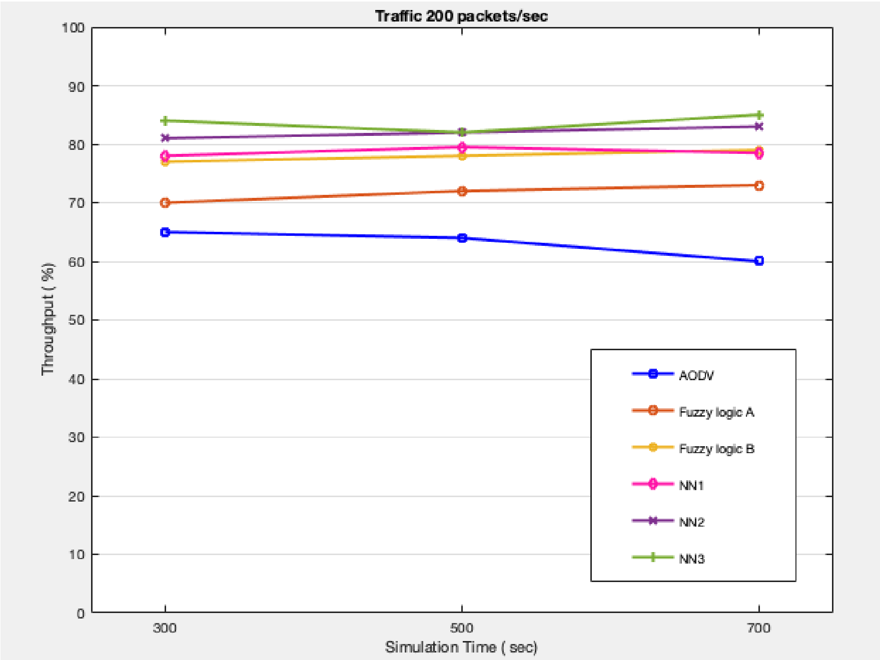

Figure 4-12 Throughput at different protocols with 200 packets/sec data traffic.

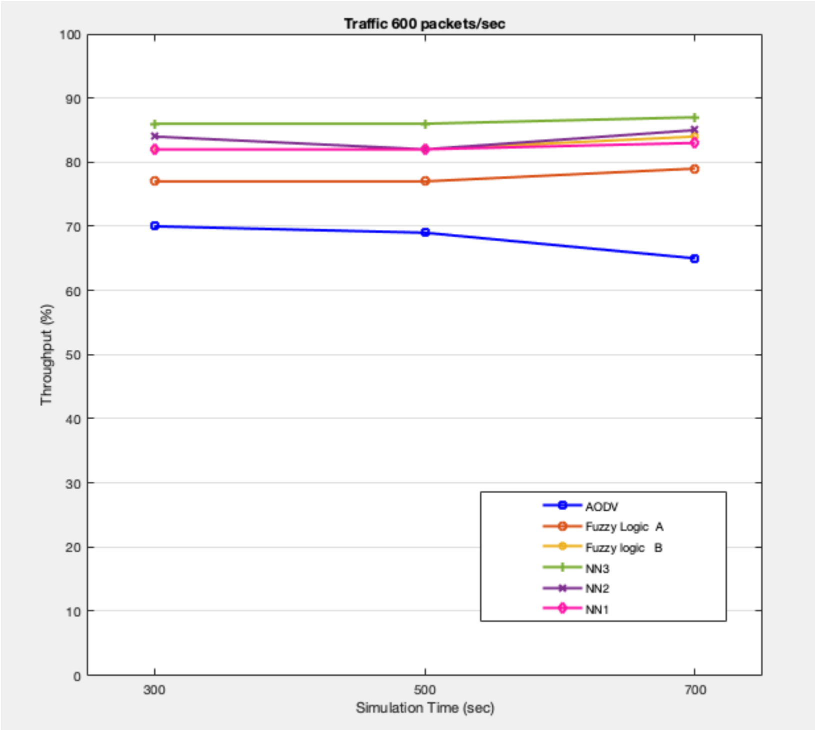

Figure 4-13 Throughput at different protocols with 600 packets/sec data traffic.

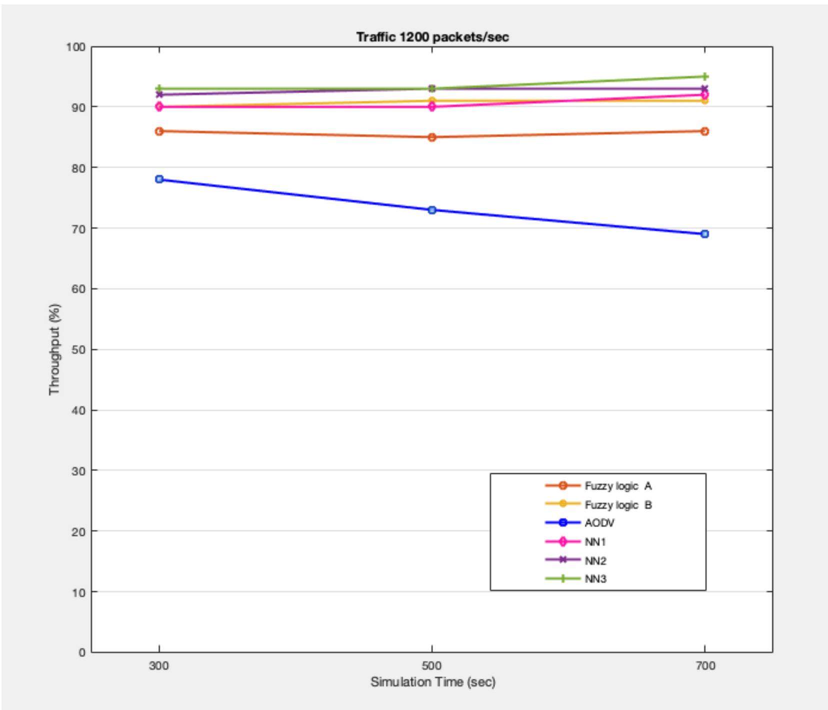

Figure 4-14 Throughput at different protocols with 1200 packets/sec data traffic.

- **End-to-End Delay:** It is defined as the average time it takes for the packets to propagate from the source node to the destination node across the network. The calculation also considers all possible delays caused by latency, buffering during discovery procedure, retransmission delays, queuing, transfer, and propagation times. A higher value of End-to-End delay indicates that the network is congested, and there is an issue with the routing protocol used. End to End delay can be measured as follows:

  End to End delay = ∑ (arrival time – send time) / No. of Delivered packets ……….(4-11)

  Simulation results shown in Figures 4-15, 4-16, and 4-17 represent all proposed protocols for experiments that have been measured with three traffic data rates of (200, 600, and 1200) packet/sec, respectively.

  It is clear from the results that the AODV routing protocol has less end-to-end delay than the other cognitive protocols proposed. The higher delay for cognitive protocols presented in other experiments, such as fuzzy logic-based routing protocols, was mainly because of the time wasted to discover the reliable routes. The packets during the discovering time stays in the node buffer until valid route results found. This process takes some time and increases the average delay, while in the AODV routing protocol, the mechanism is based on choosing the shortest path as a reliable path. More delays occurred with the new IHFN routing protocol using their scenarios NN1, NN2, and NN3 due to the sequential mode of using the fuzzy logic discovering mechanism mentioned before in addition to the pattern prediction process used within its routing decision mechanism that added further delay to the system. Limited node buffer is filled much quicker during high mobility speed. Again, things are different with AODV that does not require that much processing to select the routing path. However, with high data traffic rates, unlike other protocols using a fuzzy logic concept, AODV shows more delay caused by high network links failure due to node mobility and the drain of some of the nodes’ battery charge. In this condition, more route maintenance is required that caused more delay in the network performance.

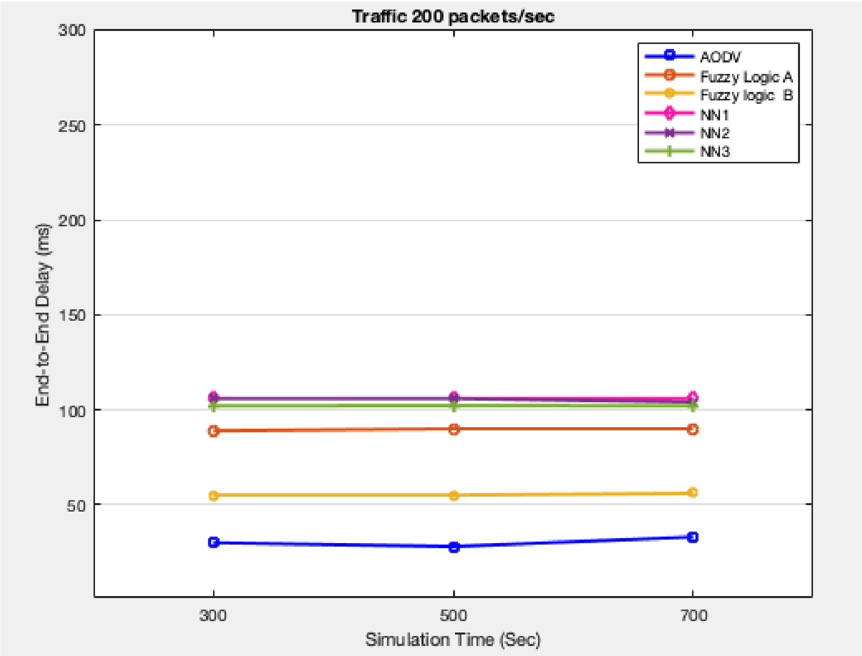

Figure 4-15 End-to-End Delay at different protocols with 200 packets/sec data traffic.

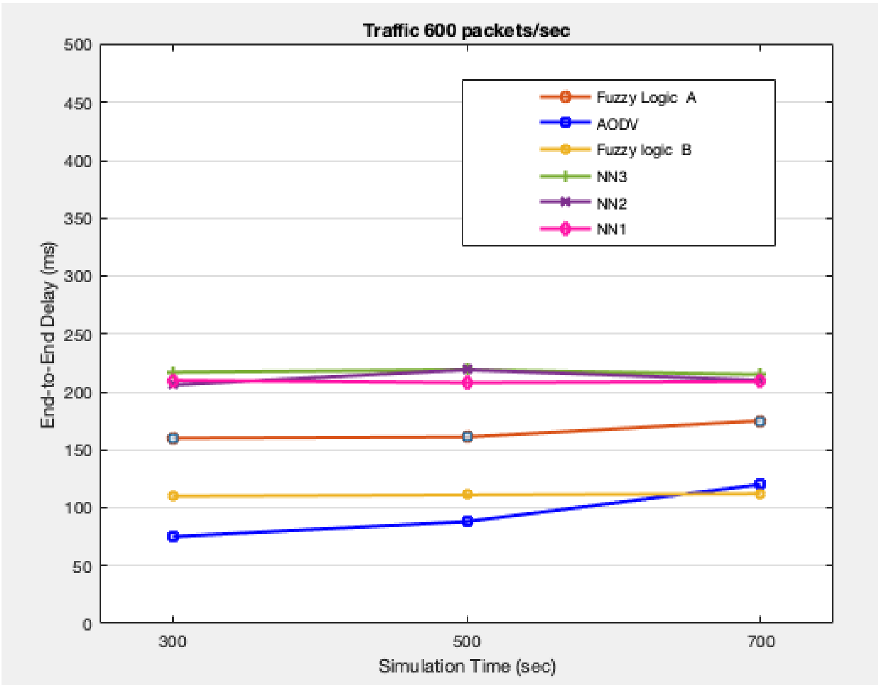

Figure 4-16 End-to-End Delay at different protocols with 600 packets/sec data traffic.

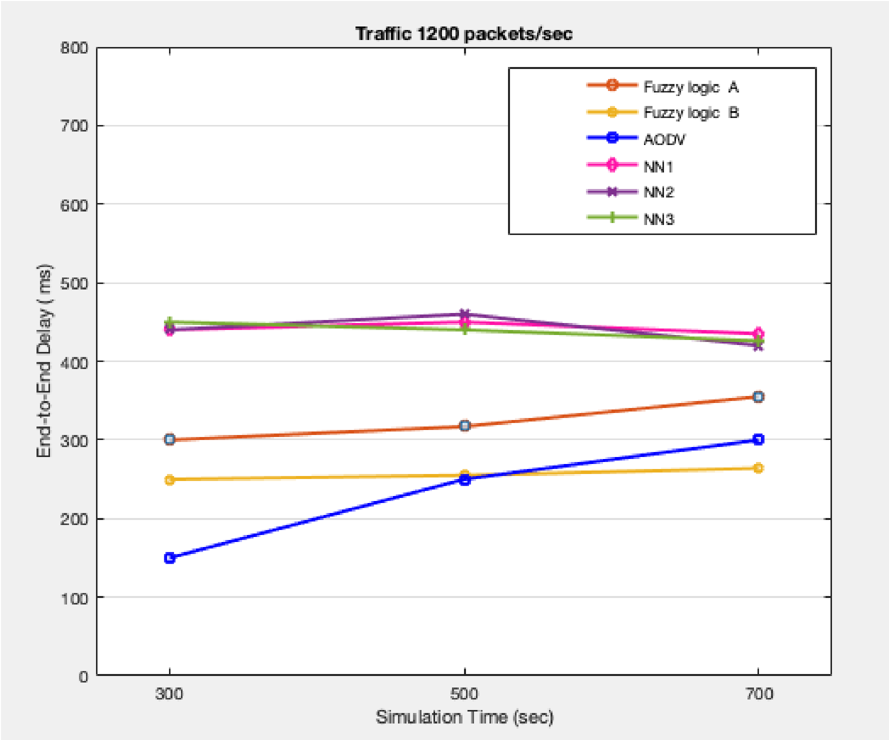

Figure 4-17 End-to-End Delay at different protocols with 1200 packets/sec data traffic.

- **PDR packets delivery ratio:** This is the ratio between the number of packets received by the destination node to the number of packets sent by the source node.

PDR = [(Packets Received /Packet sent)] *100 ……………………………………….(4-13)

Simulation results shown in figures (4-18), (4-19) and (4-20) represent all proposed protocols as experiments and scenarios that were measured with three different traffic data rates (200, 600, and 1200) packet/sec, respectively.

Results in scenarios NN2 and NN3 routing protocols provide a higher PDR ratio than other presented protocols. Using the fuzzy logic concept in routing path selection mechanism such as in Fuzzy Logic-A, Fuzzy Logic-B and IHFN routing protocols have improved the network performance. It gives a high PDR ratio by choosing only a reliable path different from the traditional AODV routing protocol that used the shortest path mechanism. However, a further enhancement to the network performance was applied by adding a prediction mechanism of essential network parameters using the new IHFN routing protocol. The prediction mechanism increased the accuracy of results to select reliable data routing paths in dynamic topologies networks condition.

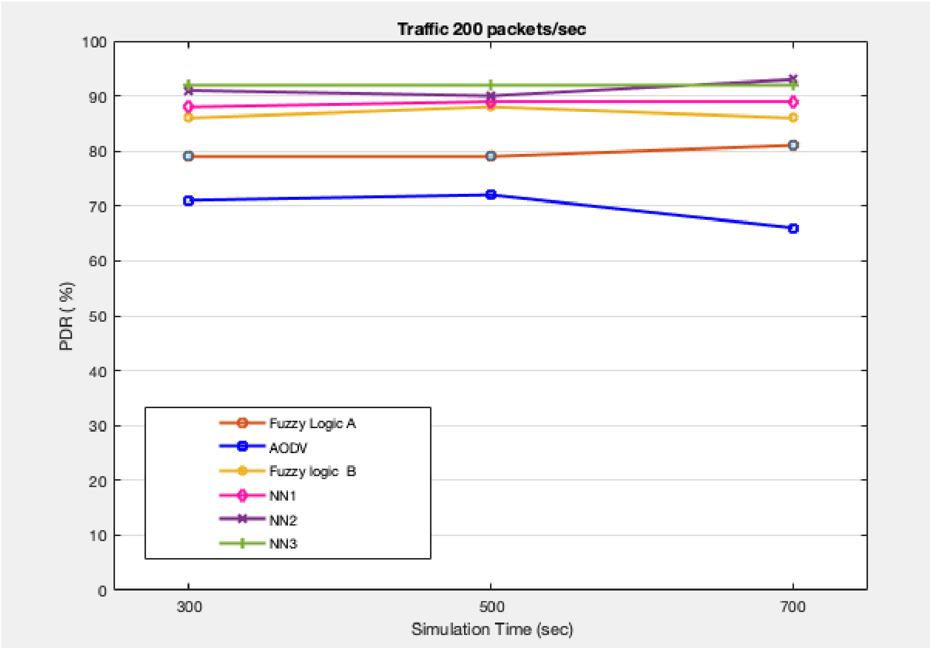

Figure 4-18 PDR at different protocols with 200 packets/sec data traffic.

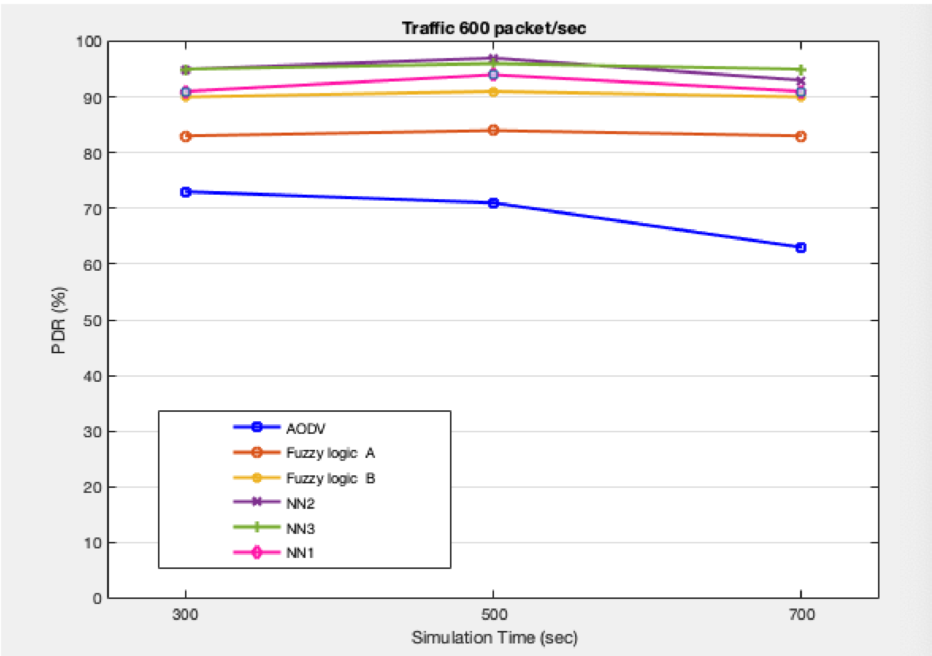

Figure 4-19 PDR at different protocols with 600 packets/sec data traffic.

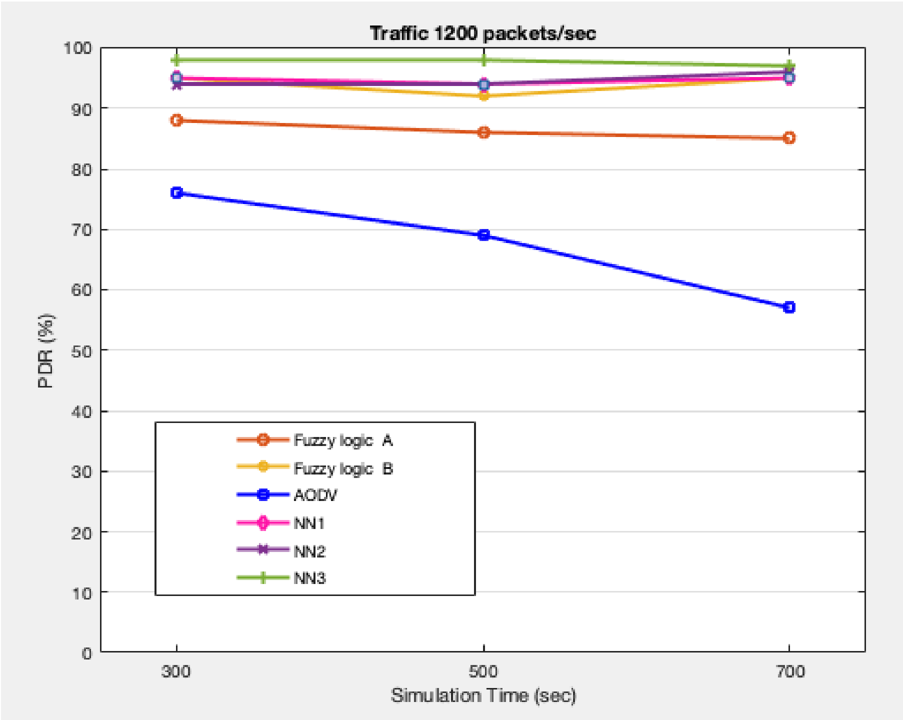

Figure 4-20 PDR at different protocols with 1200 packets/sec data traffic.

Results proved that reactive traditional protocols do not always provide better network performance because its more sensitive to the mobility, traffic rates flow, and link connectivity changes rates compared to intelligent techniques that are more adaptable to network changes.

4.8 Summary

Virtual Ad Hoc Mobile Cloud Networks (AVMCCNS) can be considered as a technique full of uncertainties because of their dynamic topology's features, different application contexts, and dynamic traffic. In AVMCCNS, due to the low signal power, limited bandwidth, and high mobility, the wireless links are frequently broken, requiring frequent new connections to be established. The dynamic network topology and the emergence of new intensive and time-sensitive mobile cloud applications can be considered significant challenges for network routing algorithms. Therefore, traditional routing protocols may not be the right choice for cloud networks because they cannot be efficiently handled to provide QoS required for a particular cloud application nor do they have the capability to adapt to any network topology changes.

This chapter introduced intelligence routing algorithms and enhancement techniques that improved network performance in AVMCCNS, where the work aims were achieved by the proposed adaptive Intelligent Hybrid Fuzzy-Neural routing algorithm (IHFN) which was simulated based on a core network that consists of realistic random mobility RWM model in a virtual Ad Hoc mobile cloud network of 60 nodes with three data traffic rates applied (Low with 200 packets/sec AODV as traditional protocol, Fuzzy Logic-A, Fuzzy Logic-B and IHFN that were presented with three scenarios). The new IHFN routing protocol was designed based on combining a fuzzy logic system that offered a natural way of reasoning and representing the problems and a neural network system that added an intelligent awareness capability of movement and resource values patterns that are considered as essential parts of network's state information. Results of the new IHFN routing protocol with NN3 scenario shows improvements in network performance with an average of 20% throughput and an average of 25% PDR higher than traditional AODV routing protocol. In addition to that, the routing protocol throughput value of the Fuzzy Logic-B shows an average of 5% higher than the routing protocol throughput value of the Fuzzy Logic-A. These throughput value improvements were caused by increasing the accuracy of measurements by collecting the updated values of node parameters when switching from RREQ packets to PREP packets. These improvements are also caused by changing the location of routing decision analysis and technique used from the destination node to the source node.

As a result, using an intelligent hybrid routing algorithm by adding pattern prediction features to the fuzzy logic algorithm, it was shown to have noticeable advantages, for instance:

- Improves the routing decision by adding the ability to switch over to another alternative route before the current working link is disconnected or disrupted either by draining node resources below the required or operational threshold or by its movements.
- Improve the routing decision by adding the ability to predict the future or the available resources and locations of nodes intelligently.
- Improvements in an accurate real-time view of almost the whole dynamic network topologies.

Chapter 5

Evaluation Performance of Different DL Algorithms in Mobility and Energy Pattern Prediction System

5.1 Introduction

With the on-going developments for cloud applications, there is always a massive increase in cloud users' demands with mobile communication technology. It is essential in this case to provide uninterrupted network access and high-quality services, especially when working in mobility with limited resources environments and when the applications required are time sensitive. Mobile networks create a huge amount of information, either as data or control information. This information can be useful to improve the quality of services provided to the mobile user and the network performance in general. Thus, there is always an increase in the need to use cognitive solutions in cloud networks by using network information intelligently to overcome all challenges related to routing and data management currently used by traditional networks. Many researchers have focused on using mobile users' movement patterns to improve the conditions and performance of communications and services [175]. With the various existing mobility models, large volumes of data like mobile positioning, energy and speeds can be sensed and collected form the network, which can be highly useful to improve network performance. In VAMCCNs, the advantages like minimising node energy consumption and improving routing decision can be achieved using mobility prediction [176]. Mobility prediction in cognitive networks is a useful technique that can be utilised by mobile devices and wireless networks, focusing on effective and efficient resource management of networks, and predicting mobile users' future locations [177].

An increase in orientation towards using machine learning algorithms as a powerful approach for pattern prediction in a mobility environment causes a significant variation in the accuracy of each algorithm performance used, which became a serious challenge in improving cognitive network data management routing protocols concepts design.

This work aims to discuss six selected Deep Learning (DL) algorithm performances to predict the future steps of position, speed, and energy of nodes based on the dataset collected from the mobile Ad Hoc network with a realistic RWM model with different data traffic rate scenarios. This work also goes one step beyond, by presenting two different techniques for using datasets to train and test the same algorithms to predict the next 300-time steps. Datasets of five randomly distributed nodes have been used in this experiment to get a trustable and more accurate analysis of the results extracted from the experiment.

5.2 Research Gap

Mobile location can easily be identified with a large degree of localization accuracy using technologies like GPS used to provide positioning with an excellent sample rate. Some previous research used the location-based concept by using location history shared by social networking applications. Research approaches to the prediction of the location have been improved with higher precision and greater feasibility. In [97], the author presented a mobile location identification method using mobility patterns predictions taken from the history of mobile movements' path by collecting the data of time, location, and mobile user's current state. Markov Model [178] is one of the traditional presented approaches. This model uses one order and multiple order state transition matrix to form people's mobility pattern. In [179], the author proposed a mobility prediction model based on the n-Mobility Markov Chain algorithm to predict the next location of the mobile user using the previous (n) visited locations. There were several other types of studies and research conducted by many researchers to overcome location prediction problems. The rising interest in artificial intelligence and machine learning in the pattern prediction system is well illustrated by the sharp increase of developments and research interest in this domain. Mobility prediction in Ad Hoc neural networks is also an important method to improve routing decisions in wireless mobile Ad Hoc networks. In [180], the author proposed a recycling time series production using a recurrent neural network with a scheme that predicts mobile movement in Ad Hoc networks. This scheme is based on a multi-layer and recurrent neural network using backpropagation through a time algorithm for training. The prediction of the node's future destination is performed by stable path estimation, which leads to an optimal routing process. Another approach to comparing the performance of different techniques using mobility prediction, is presented in [181] where author evaluated the performance of using time-series forecasting techniques for vehicular mobility prediction. The work started by extracting vehicles' movements featured into times series sequence of observations and used them as an input to different machine learning predictors. The performance of each algorithm used was evaluated and compared with others for accuracy based on Route Mean Square Error (RMSE).

5.3 Proposed System

A dataset was created and collected from a virtual Ad Hoc mobile cloud network with RWM mobility model and with three data traffic rate scenarios. This proposed work started by creating three datasets from the original datasets collected.

The first dataset is called DATA-Comp, consisting of all original datasets collected, combined, and separated into two sets of data, training and testing datasets. This dataset was applied in the first experiment named (Comparison Experiment). In this experiment, six selected machine learning models (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM) were used, where each of these models has been trained with the (DATA-Comp) training dataset and tested with the DATA-Comp testing dataset for x, y, speed, and energy parameters. Results were compared between all algorithms to find a suitable model for pattern prediction that has less error value, where the metric used for comparison is the Root Mean Square Error (RMSE). The second experiment proposed (Single-Multiple Experiment) applied as two scenarios representing two techniques of using the datasets for training and testing the same algorithms for the pattern prediction model. Comparison of results gives a good indication of the influence of dataset technique used that improves the process to give better prediction results. The second experiment used five randomly selected nodes (7, 15, 24, 33 and 48) for its analysis. The (DATA-Single) is the dataset arranged from the original dataset collected of each node individually that was used for the first scenario. Each node dataset combined its (three data traffic rates) then splits into training and testing. Five single datasets were formed and applied separately to all the six deep learning models (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM). Results of predicting positioning (x, y), speed and energy parameters were calculated using the next 300-time steps. RMSE was the metrics used to evaluate each machine learning model's performance per each node per each parameter.

The second scenario used (DATA-Multiple datasets). These datasets were obtained from taking the original total dataset collected of 60 nodes to extract the training dataset from it, while the testing datasets were taken from the same five selected nodes used in this experiment's first scenario. The (DATA-Multiple) datasets were applied to the same six selected machine learning algorithms. The results of predicting the positioning (x, y), speed, and energy parameters were calculated using the next 300-time steps. RMSE value is a metric used to evaluate each algorithm model's performance per each node per each parameter.

Finally, both scenarios' results were compared and analysed to find the best technique for arranging training and testing datasets to be applied to ML algorithms to get better accurate result for the pattern prediction models.

5.3.1 Datasets (Creation and Collection)

Datasets have been created and collected from a Virtual Ad Hoc mobile cloud network with RWM mobility model with three data traffic scenarios. All scenarios were simulated using MATLAB R2020a 64-bit platform. The network includes 60 Ad Hoc mobile nodes moved according to the RWM mobility model and two other nodes as a source and destination. The grid size selected for this mobility model (the arena of the simulation) was set to 600X600. Parameters for this network has been selected as shown in table 5-1.

Parameters	
Mobility Model	RWM model
Area of deployment	600x600 m^2
Number of nodes	60
Simulation time	750
speed (max. and min.)	[0, 8] m/s
Pause time (max. and min.)	[0,1] sec
MNs Max. charging capacity 100%	3000 joules = 4 Watt
Transmission Range	100m
Packet size	CBR
Packet rates	Low rate (200 packet/sec) Medium rate (600 packet/sec) High rate (1200 packet/sec)

Table 5-1 RWM Parameters.

The dataset collection experiment started by distributing all of the 60 nodes into randomly distributed initial positions (x, y) within the simulation arena; each of the 60 nodes stayed in its initial location for a certain selected randomly distributed period (pause time) between [0-1] sec. As soon as this time comes to an end, each mobile node is assigned a new randomly distributed destination within the simulation arena and given a uniformly distributed random speed range between [0-8] m/sec. Upon arrival, the mobile node takes another pause for a specific random period of time before doing the same process until the simulation time of 750 ended. By using a null matrix, all variables are initialized and then all new values are being collected and saved inside it. In this experiment also, a full charge capacity of each node has been assigned as an estimation to be 4 watts which converted to 3000 Joules for 750 seconds periods that represents the simulation time by using the following equations:

$$E_{(J)} = P_{(W)} * t_{(S)} \tag{5-1}$$

Where the energy E in joules (J) is equal to the power P in watts(W), times period in seconds(s). Mobile node numbers from 1 to 20 are assigned randomly with residual mobile battery energy values between 0 and 50% of its maximum charging capacity. Mobile node numbers from 21 to 40 are assigned randomly with residual mobile battery energy values between 25 to 75% of its maximum charging capacity. Finally, node numbers from 41 to 60 are assigned randomly with residual mobile battery energy values between 50 to 100% of its maximum charging capacity. The experiment started with RWM movements for all 60 nodes, where three scenarios applied by sending data traffic packets of 200, 600 and 1200 packets /second respectively from the source node to the destination node using an AODV routing protocol for 750 seconds simulation time each. Values of each node's positioning, speed and battery charging capacity were measured and saved for the whole simulation time (750 sec) for the three scenarios. Datasets were saved in Comma Separated Values (CSV formats).

5.3.2 **Training and Testing Datasets Forms for The Two Experiments.**

In this work, two experiments have been applied, a Comparison Experiment and a Single-Multiple dataset experiment. Datasets for training and testing were formulated according to these experiments' main requirements and the work's aims. For that reason, three datasets were obtained from the total datasets collected to perform the two proposed experiments.

The original total datasets have been created and collected from the simulation outputs of 60 nodes Ad Hoc mobile network. All nodes in mobile condition is based on RWM mobility model with three data traffic scenarios applied (low, medium, and high). The simulation time selected was 750 seconds time slots that represented the whole simulation period per each scenario, which can also be considered the size of rows of dataset per node per scenario. Four parameters (columns) of data measured represent (x, y) for positioning, speed, and energy of the nodes. It means that the total dataset consists of the following information:

Total dataset per parameter (column) = (750 *60*3) = 135000 data rows, here 750 represents the time steps readings saved, 60 represents the number of nodes, and 3 represents the number of data traffic rate scenarios used. The three datasets were formulated as shown below:

- **For the First Experiment (Comparison Experiment)**

  DATA-Comp.

  The original data collected were divided into the training dataset and testing dataset to train and test the six selected models' performance representing the first experiment.

  The total dataset per parameter (column) is consisting of (750 *60*3) = 135000 data then by splitting the last 100 time slots per each node per each data traffic rate for testing. the remaining were used for training, the two datasets was formulated as shown below:

  Training dataset = 650*60*3 = 117000 rows * 4 columns

  Testing Datasets = 100*60*3 = 18000 rows * 4 columns

  Figure 5-1 illustrates the training and testing dataset rows and columns used in this experiment as the main dataset.

```
Main Dataset that is Used:
==========================


Training Dataset:


         pos_x    pos_y     speed      energy
0       307.922  503.508  0.000000  121.188655
1       307.871  503.345  1.974358  120.165500
2       306.494  504.296  1.974358  119.142345
3       305.116  505.246  1.974358  118.119190
4       303.738  506.197  1.974358  117.096035
...         ...      ...       ...         ...
116995  223.377  451.449  1.591080  294.398640
116996  223.885  450.748  1.591080  290.438050
116997  224.393  450.046  1.591080  286.477460
116998  224.901  449.344  1.591080  282.516870
116999  225.409  448.643  1.591080  278.556280

[117000 rows x 4 columns]


Testing Dataset:


        pos_x    pos_y  speed  energy
0      204.428  583.988    0.0     0.0
1      204.428  583.988    0.0     0.0
2      204.428  583.988    0.0     0.0
3      204.428  583.988    0.0     0.0
4      204.428  583.988    0.0     0.0
...        ...      ...    ...     ...
17995  210.673  468.992    0.0     0.0
17996  210.673  468.992    0.0     0.0
17997  210.673  468.992    0.0     0.0
17998  210.673  468.992    0.0     0.0
17999  210.673  468.992    0.0     0.0

[18000 rows x 4 columns]
```

Figure 5-1 Training and testing datasets used in the first Experiment- DATA-Comp.

- **For the Second Experiment (Single-Multiple Scenarios)**
  - **(DATA-Single) Dataset**

    DATA-Single, represents the datasets that was used for the second experiment for the single node datasets scenario. Dataset collected per each node was used separately for training and testing each algorithm. The total dataset per each node was split as shown below:

Total dataset = (750 *60*3) = 135000 data * 4 columns

Training dataset = 650*3 = 1950 rows * 4 columns per each node

Testing Datasets = 100*3 = 300 rows * 4 columns per each node

Five nodes were randomly selected for this experiment, these nodes are (7, 15, 24, 33 and 48).

Figure 5-2 illustrates an example of node 24 datasets with its training and testing dataset rows and columns to form a single dataset (DATA-Single).

```
Main Datasets of Node 24 that is Used:
====================================

Training Data:

        pos_x    pos_y     speed       energy
0     368.653  91.6318  0.000000  2137.551711
1     368.548  90.8244  0.370118  2137.110000
2     368.392  89.6361  0.370118  2136.668289
3     368.237  88.4477  0.370118  2136.226578
4     368.082  87.2593  0.370118  2135.784867
...       ...      ...       ...          ...
1945  381.943  69.8461  2.405508  1787.561232
1946  382.406  70.5222  2.405508  1786.899060
1947  382.869  71.1982  2.405508  1786.236888
1948  383.332  71.8743  2.405508  1785.574716
1949  383.794  72.5503  2.405508  1784.912544

[1950 rows x 4 columns]

Testing Data:

       pos_x    pos_y     speed       energy
0    384.720  73.9024  2.405508   775.181610
1    385.183  74.5785  2.405508   773.047275
2    385.646  75.2545  2.405508   770.912940
3    386.109  75.9306  2.405508   768.778605
4    386.572  76.6066  2.405508   766.644270
..       ...      ...       ...          ...
295  382.406  70.5222  7.435760  1467.316100
296  382.869  71.1982  7.435760  1463.320168
297  383.332  71.8743  7.435760  1459.324236
298  383.794  72.5503  7.435760  1455.328304
299  384.257  73.2264  7.435760  1451.332372

[300 rows x 4 columns]
```

Figure 5-2 Training and testing datasets for node 24- DATA-Single.

- **(DATA-Multiple) Dataset**

  This dataset was used for the second experiment in the multiple node's datasets scenario. The training and the testing datasets obtained from this scenario was applied to each of the six selected algorithms.

  DATA-Multiple datasets were formulated from the total original dataset as follows:

  Training dataset = 650*60*3 = 117000 rows * 4 columns

  Testing Datasets = 100*3 = 300 rows * 4 columns

  As shown above, the training dataset includes the whole data of the 60 nodes out of 100 times slots per each node per each data traffic rate, the 100-time slots left of any specific node multiplied by three which reparents the data traffic scenarios formed the training dataset. The same five randomly selected nodes before (7, 15, 24, 33 and 48) are used for testing the dataset to make the required comparison.

  Figure 5-3 illustrate an example of training and testing dataset rows and columns for node 24 that was used as multiple datasets.

```
Main Datasets of Node 24 that is Used:
=======================================

Training Dataset:

          pos_x    pos_y     speed       energy
0        307.922  503.508  0.000000  121.188655
1        307.871  503.345  1.974358  120.165500
2        306.494  504.296  1.974358  119.142345
3        305.116  505.246  1.974358  118.119190
4        303.738  506.197  1.974358  117.096035
...          ...      ...       ...         ...
116995   223.377  451.449  1.591080  294.398640
116996   223.885  450.748  1.591080  290.438050
116997   224.393  450.046  1.591080  286.477460
116998   224.901  449.344  1.591080  282.516870
116999   225.409  448.643  1.591080  278.556280

[117000 rows x 4 columns]

Testing Data:

       pos_x    pos_y     speed        energy
0    384.720  73.9024  2.405508    775.181610
1    385.183  74.5785  2.405508    773.047275
2    385.646  75.2545  2.405508    770.912940
3    386.109  75.9306  2.405508    768.778605
4    386.572  76.6066  2.405508    766.644270
..       ...      ...       ...           ...
295  382.406  70.5222  7.435760   1467.316100
296  382.869  71.1982  7.435760   1463.320168
297  383.332  71.8743  7.435760   1459.324236
298  383.794  72.5503  7.435760   1455.328304
299  384.257  73.2264  7.435760   1451.332372

[300 rows x 4 columns]
```

Figure 5-3 Training and testing data for node 24- DATA-Multiple.

In Figure 5-4, the plotted graph illustrates an example of the user mobility speed data for the five selected nodes, the X-axis is representing the simulation time slots and Y-axis is the speed in (m/s). This data was used for the second experiment (Single-Multiple) training dataset.

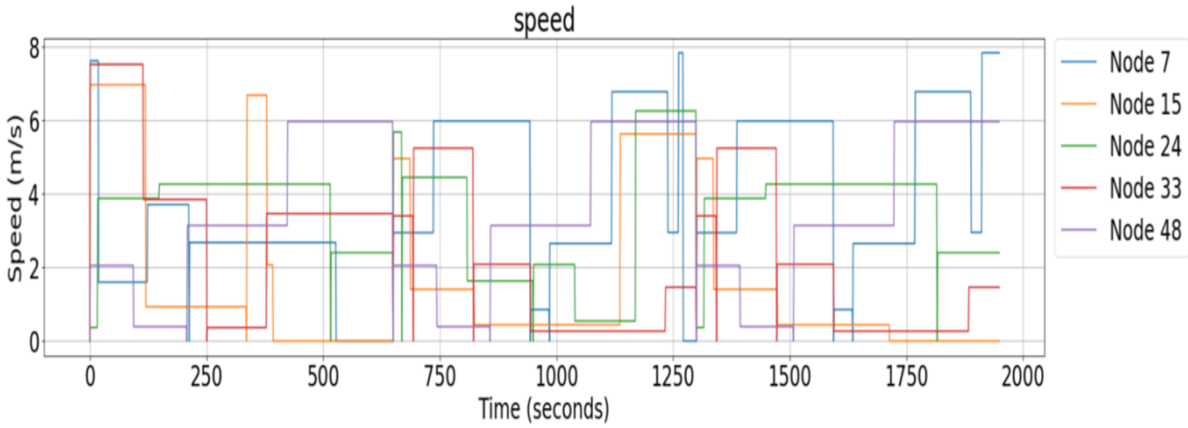

Figure 5-4 Example of five nodes speed dataset for (Single-Multiple) training dataset.

5.4 Simulation and Results

Python version 3.7.4 has been selected for this experiment because it is considered one of the most preferred programming languages for simulating machine learning as the syntaxes used are easy and simple to learn [129]. Jupyter Notebook was used in this work as an open-source web application; it includes all opensource required in providing data analysis and for machine learning algorithms.

The selected models used for comparison for both experiments are:

- Recurrent Neural Network (RNN).
- Long short-term memory (LSTM)
- Gated Recurrent Unit (GRU).
- Conventional neural network-LASTM (CNN-LSTM).
- Bi-directional LSTM.
- Stacked LSTM.

5.4.1 **Experiment 1 (Comparison Experiment)**

In this experiment, DATA-Comp training and testing datasets were applied for all six algorithms. The prediction results of these models (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM) with respect to the position x, position y, speed and energy are shown in the Figures 5-, 5-6, 5-7 and 5-8 respectively.

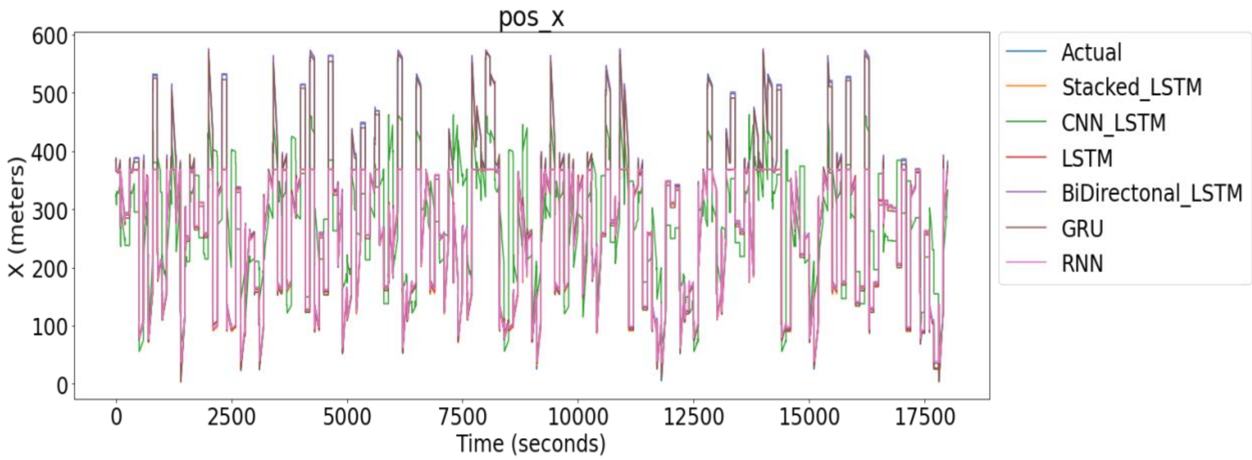

Figure 5-5 Predicted values of position x for all algorithms.

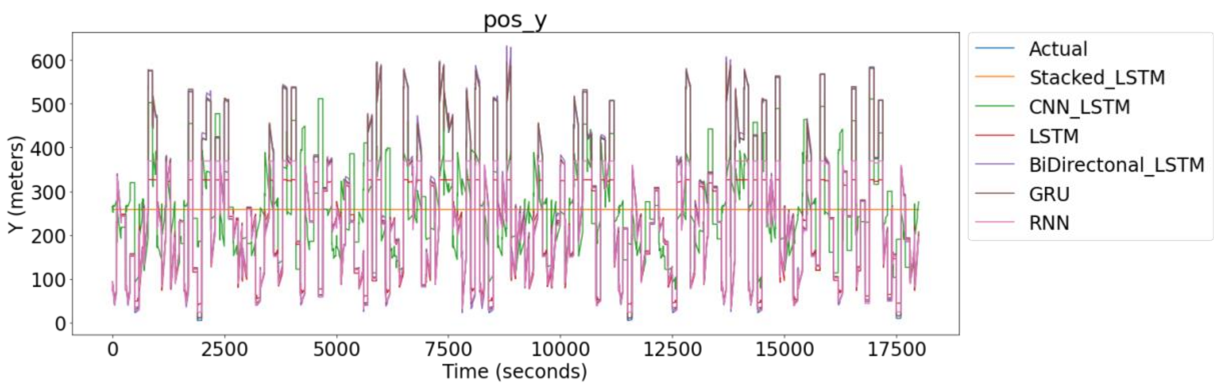

Figure 5-6 Predicted values of position y for all algorithms.

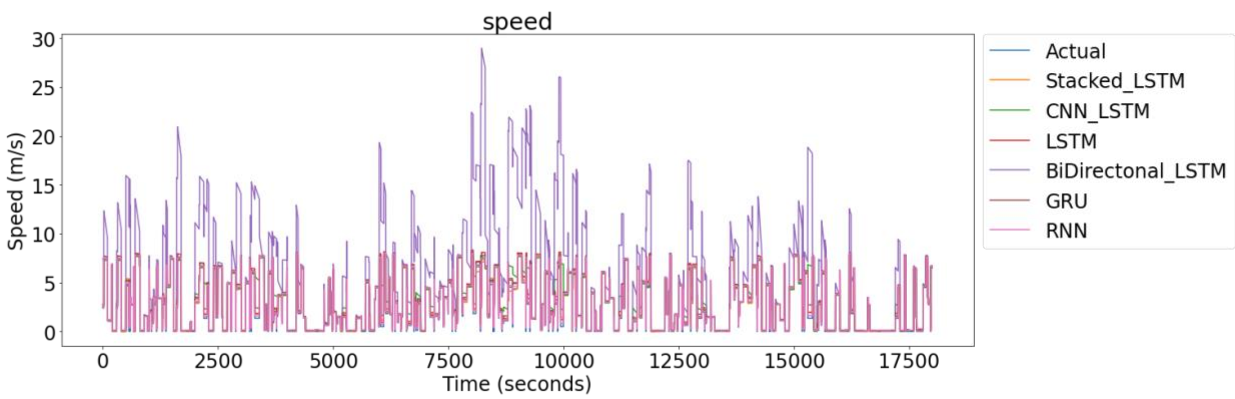

Figure 5-7 Predicted values of speed for all algorithms.

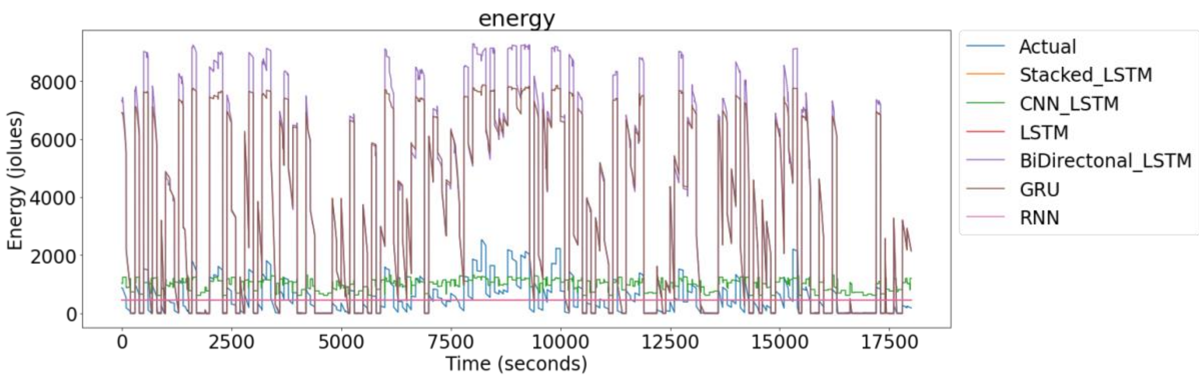

Figure 5-8 Predicted values of energy for all algorithms.

Table 5-2 illustrate the final results of the prediction in the route mean square error (RMSE) values. The results have also been plotted as a bar chart as shown in figure 5-9.

Algorithm	Pos X RMSE	Pos Y RMSE	Speed RMSE	Energy RMSE
Stacked_LSTM	61.57	141.6	308.98	222.47
CNN_LSTM	102.38	111.98	308.68	748.43
LSTM	61.29	163.6	308.94	224.23
BiDirectonal_LSTM	20.1	215.57	306.97	5831.72
GRU	20.72	208.72	308.83	4421.74
RNN	62.8	194.91	309.03	222.87

Table 5-2 RMSE values of the total algorithms comparison results.

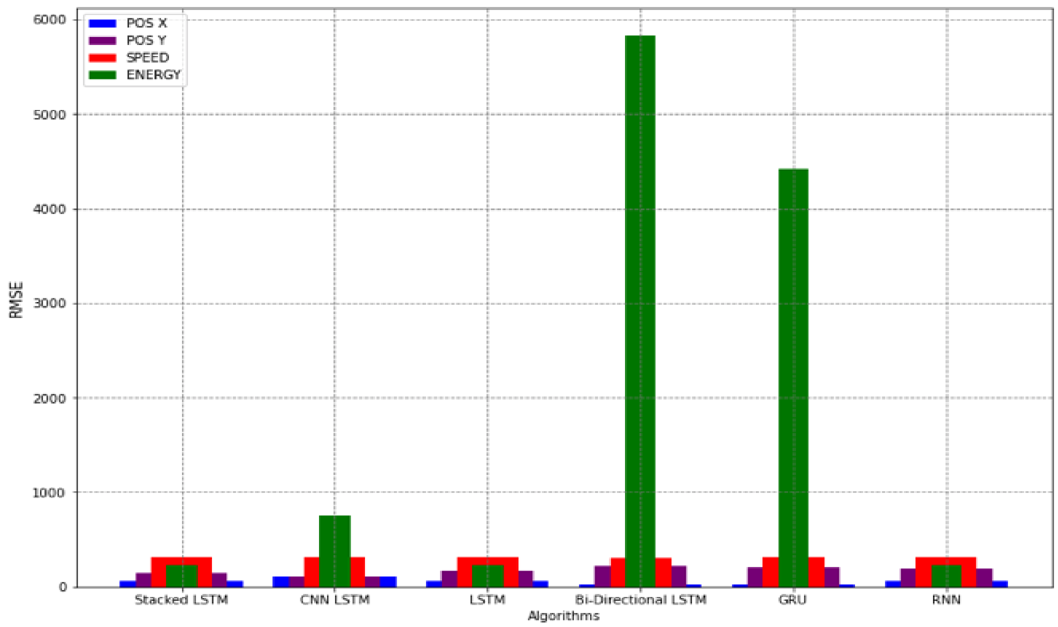

Figure 5-9 RMSE values of all algorithms.

The bar chart above shows the results in RMSE when applied (DATA-comp) on all the six algorithms (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM) to predict the next steps for nodes parameters (x, y, speed, and energy). Stacked- LSTM has the lowest errors with the prediction results for all four parameters used in mobility and energy prediction. RNN and LSTM models also show very good results, while for the other three algorithms models (CNN-LSTM, BI- Directional LSTM and GRU), the results were good with respect to the prediction of the position (x, y) and speed parameters, but they gave a large error value when energy values were predicted. It can be concluded that the shape of the data pattern evaluated, affecting algorithms' performance sometimes makes the algorithms give inconsistent and inaccurate results. However, for other algorithms like Stack LSTM, RNN and LSTM, there were more reliable in this experiment when dealing with complex datasets.

5.4.2 Experiment 2 (Single- Multiple node datasets experiment)

In this experiment, two scenarios (techniques) were implemented:

- **Single node dataset:** In this technique, each node dataset with its three scenarios were combined as a single dataset (DATA-single), and it was split into training and testing datasets and was applied on all six deep learning algorithms selected (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM) to predict the next 300-time steps. Five single datasets related to nodes (7, 15, 24, 33 and 48) have been selected for this experiment separately for the four parameters (x, y, speed, and energy).
- **Multiple node dataset:** In this technique, the total original dataset has been extracted to get the training dataset, while for the testing dataset, the same datasets have been used in single node scenario for the same five selected nodes. The final datasets (DATA-Multiple) were applied into all six deep learning algorithms selected (RNN, GRU, Bi-Directional LSTM, LSTM, CNN, LSTM and Stacked LSTM) to predict the next 300-time steps. The same five single datasets related to nodes (7, 15, 24, 33 and 48) have been selected for this experiment separately for the four parameters (x, y, speed, and energy).

Both scenarios' results of the single-node model and multiple nodes model were calculated and compared based on using the datasets of the five nodes selected and by applying it to all six deep learning algorithms. Figures 5-10, 5-11 , 5-12, and 5-13 illustrate an example of the results comparison for node 48 dataset to predict the next 300-time steps for position x, y, speed and energy parameters respectively using both techniques single and multiple models by applying them to all six selected algorithms.

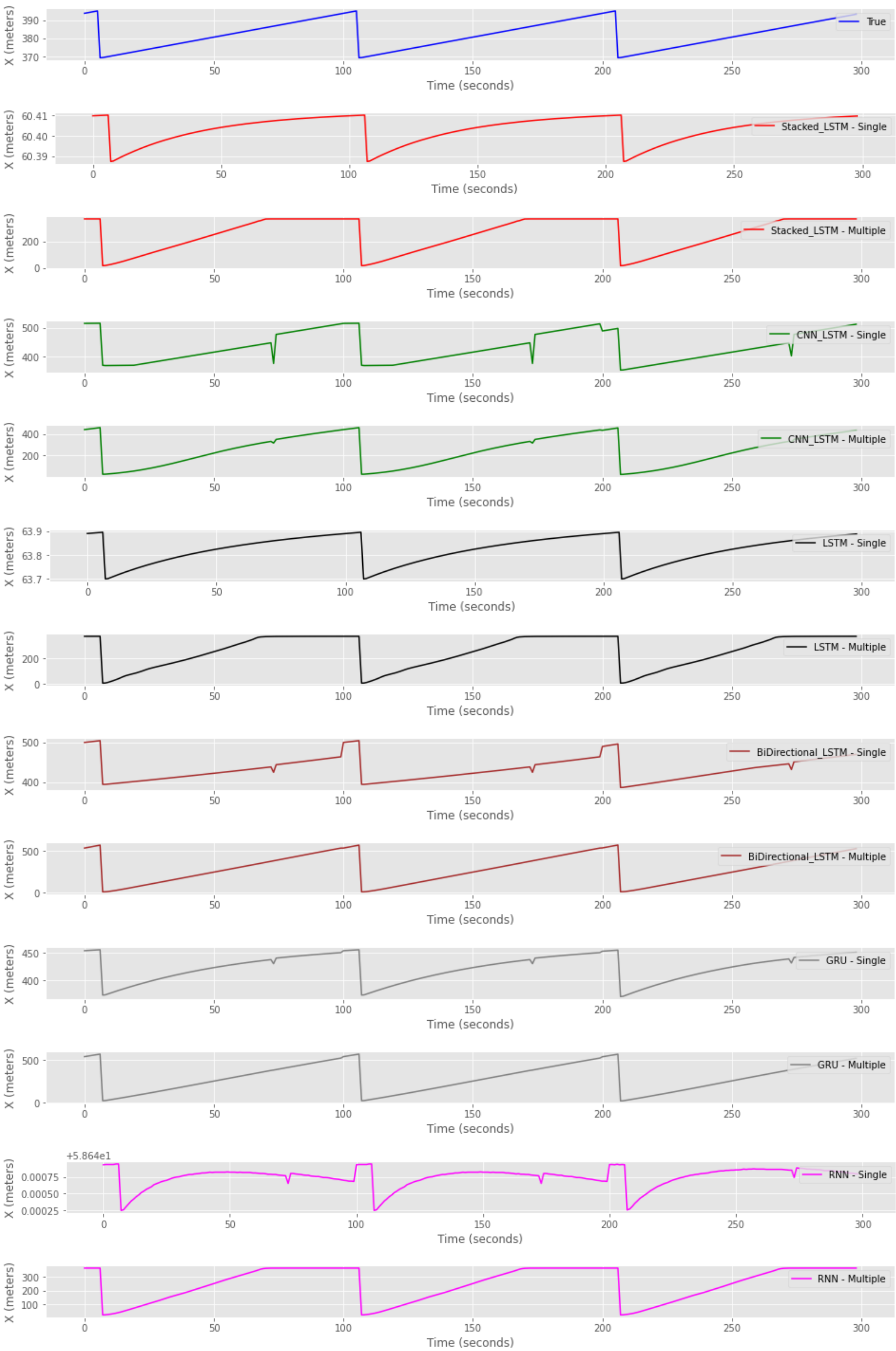

Figure 5-10 Single- Multiple techniques comparison for node 48-X prediction parameter.

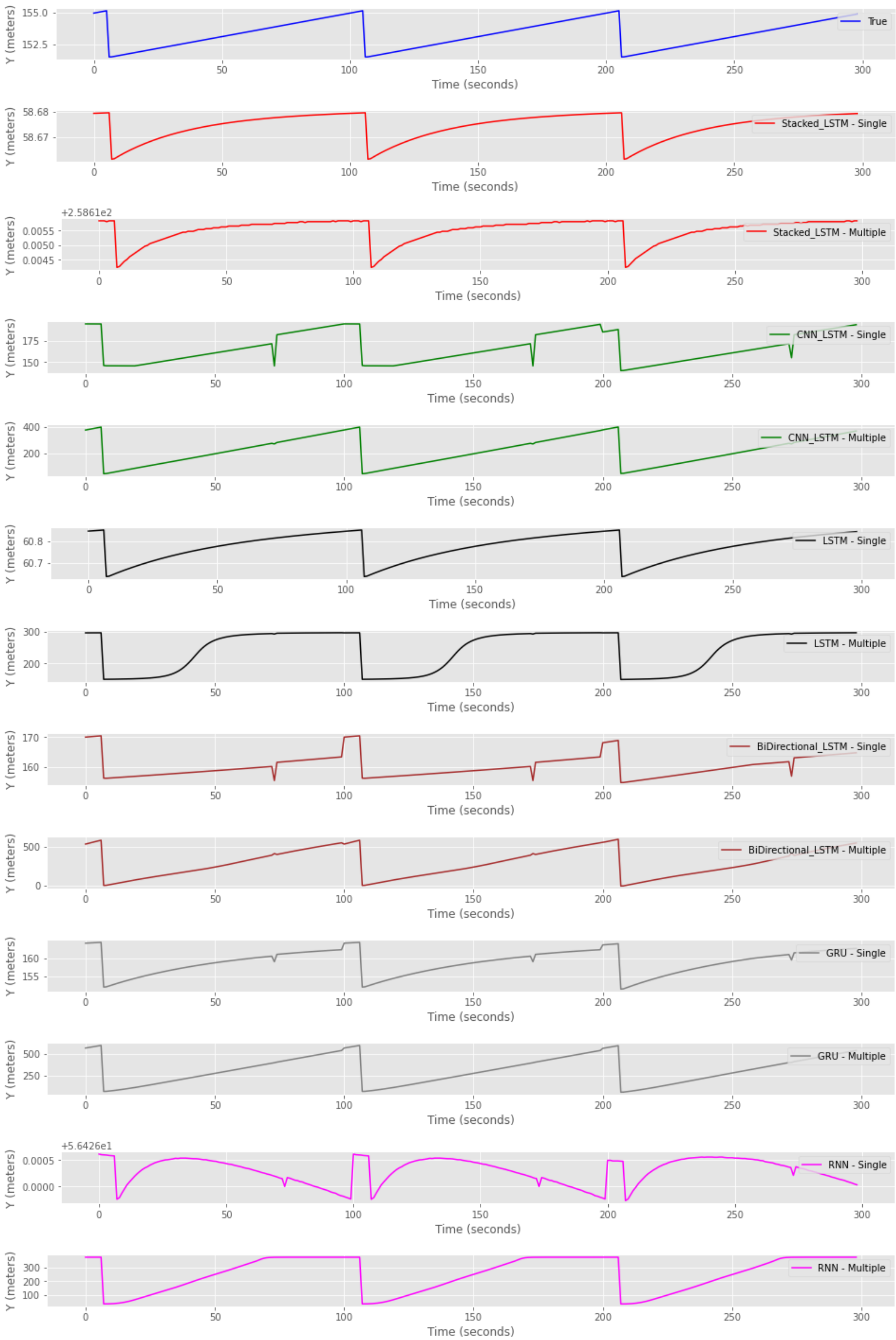

Figure 5-11 Single- Multiple techniques comparison for node 48-Y prediction parameter.

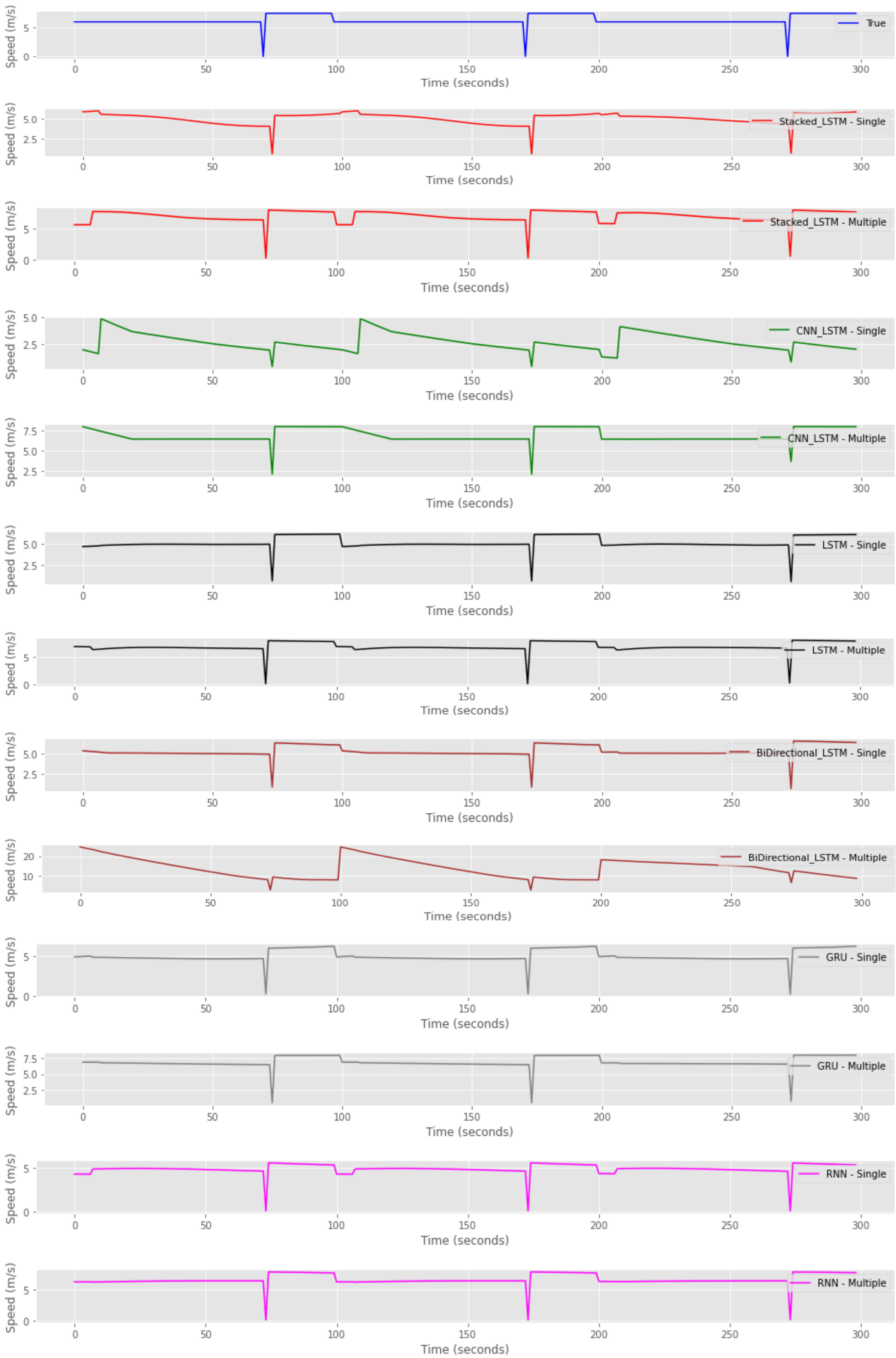

Figure 5-12 Single- Multiple techniques comparison for node 48-speed prediction parameter.

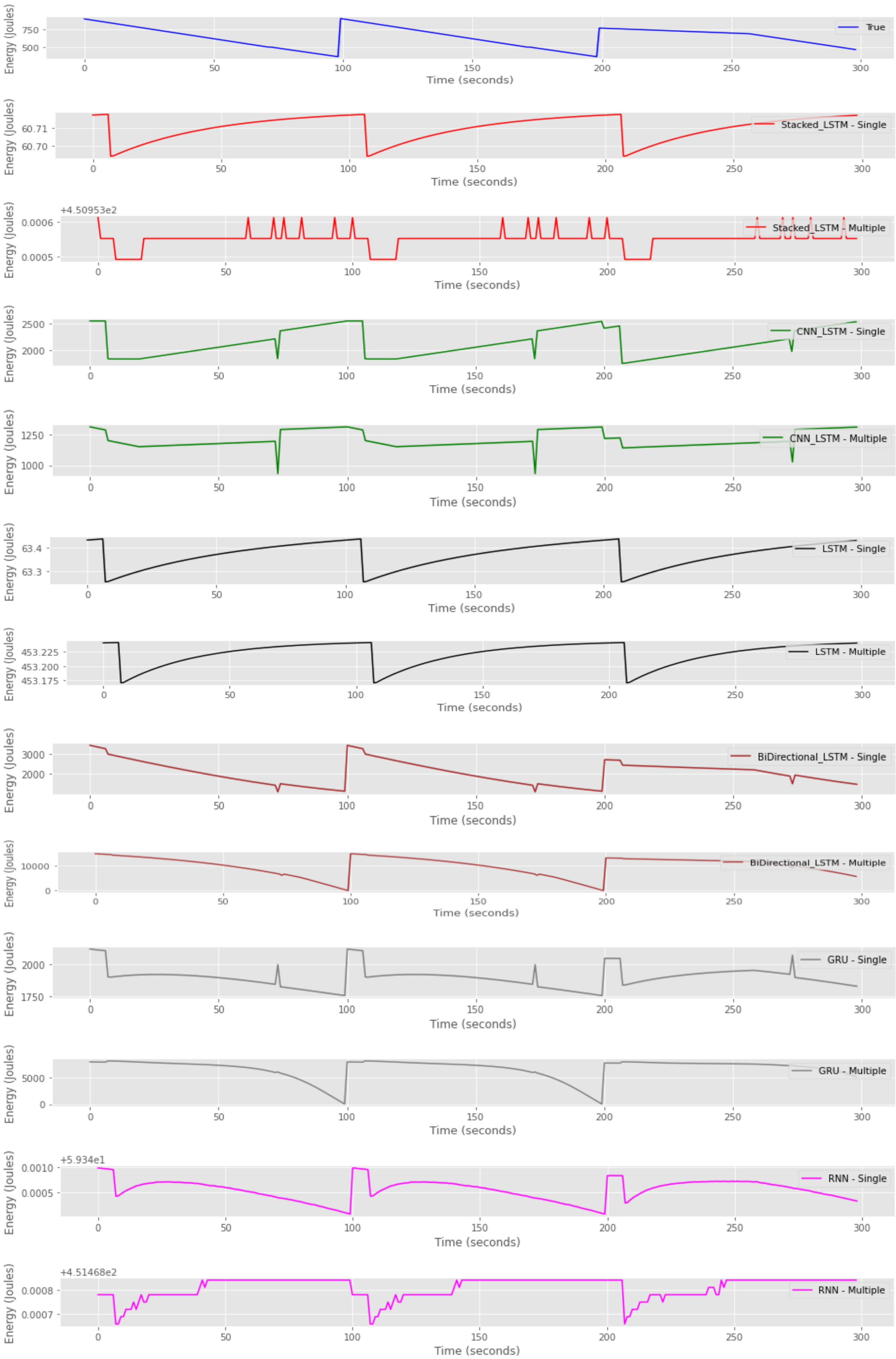

Figure 5-13 Single- Multiple techniques comparison for node 48-energy prediction parameter.

The final comparison results in RMSE values for nodes (7, 15, 24, 33 and 48) datasets have been illustrated in Appendices (A, B, C, D, and E) respectively. The results show each node's four parameters to predict the next 300-time steps using single and multiple node datasets techniques applied to all six parameters.

Figure 5-14 illustrates the bar chart for all single node model final results (five selected nodes with all six algorithms) of Position x, y, and Energy while for the speed's result, it is illustrating separately in Figure 5-15.

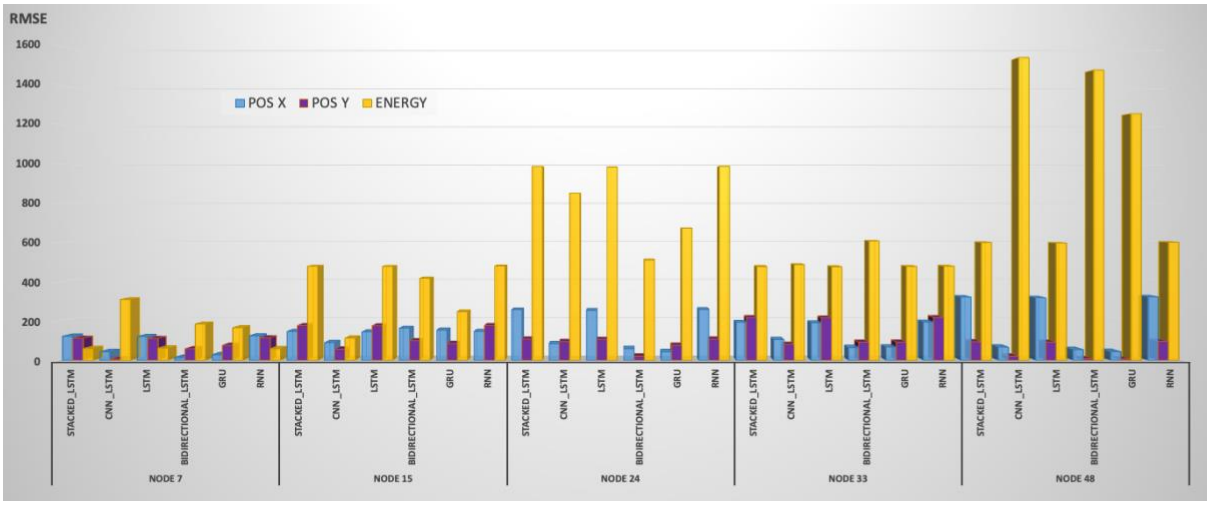

Figure 5-14 Single node model scenarios with Position x, y, and Energy final results.

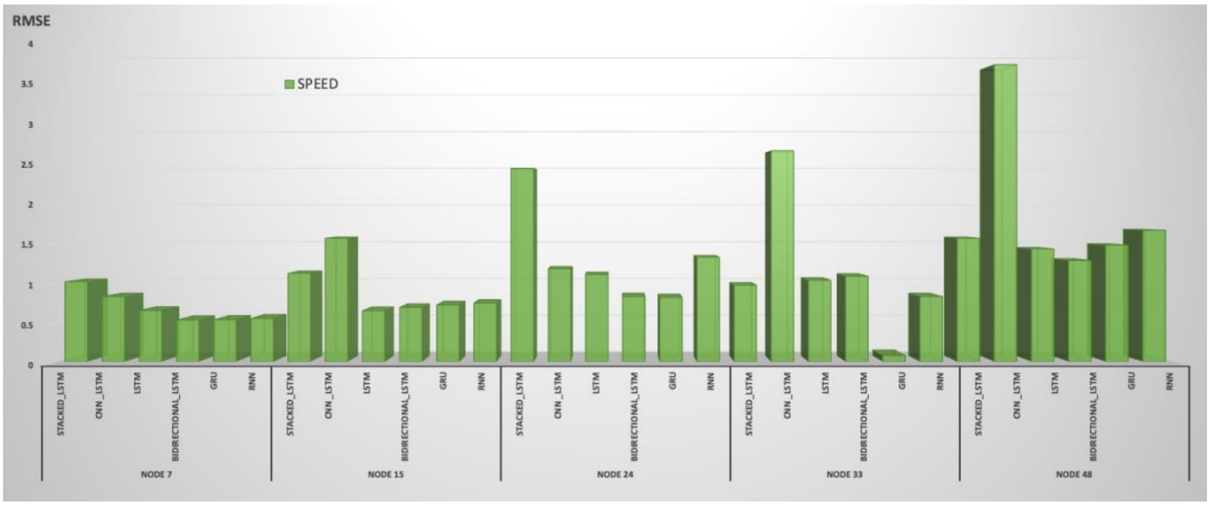

Figure 5-15 Single node model scenarios with the Speed final results.

Also, Figure 5-16 illustrates the bar chart for all Multiple nodes model final results (five selected nodes with all six algorithms) of Position x, y, and Energy while for the speed's result, it is illustrating separately in Figure 5-16.

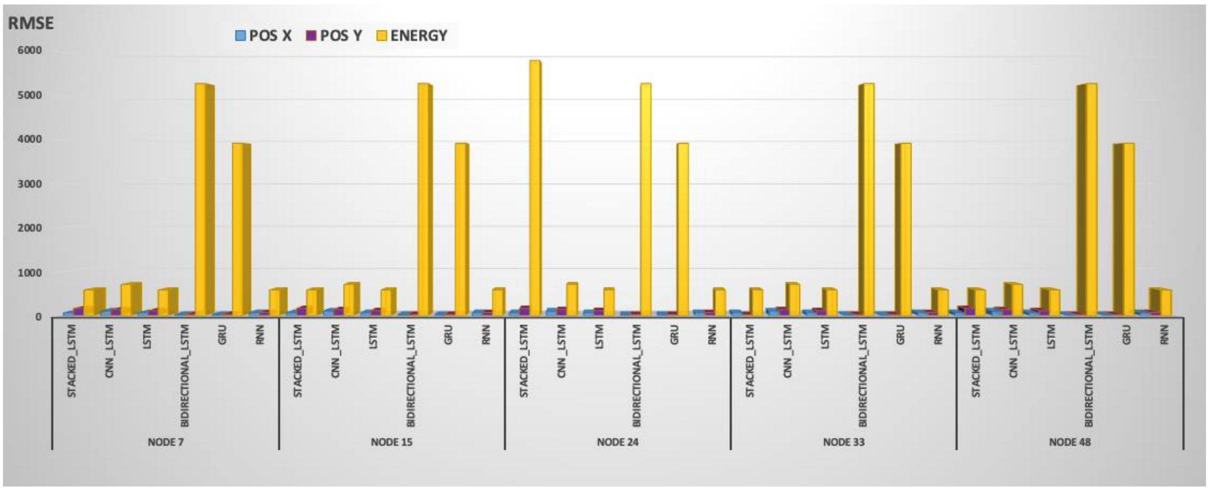

Figure 5-16 Multiple nodes model scenarios with Position x, y, and Energy final results.

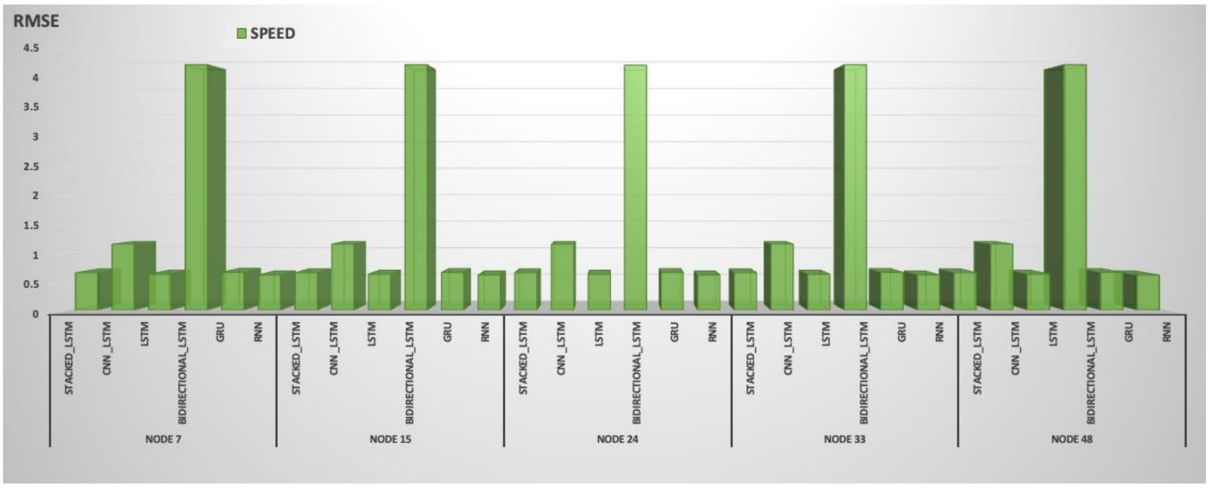

Figure 5-17 Multiple nodes model scenarios with the Speed final results.

Based on the above final results for the second experiments:

- Stacked LSTM, RNN and LSTM algorithms, gave lower RMSE values with the next 300-time steps prediction for x, y, speed, and energy using the multiple nodes dataset compared to single node dataset.
- With Bi-Directional-LSTM algorithm, the RMSE values were lower when using a single node dataset compared to multiple node dataset.

- In GRU and CNN-LSTM algorithms, RMSE values were lower in multiple node dataset for x, y and energy parameters but with speed parameters situation is different as its more accurate to predict the next 300-time steps for the speed using the single node.

It can be concluded that:

1- Datasets contents used for training and testing were considered a critical factor in getting better performance. From the second experiment, different algorithms showed different performance related to the training and testing dataset techniques used.
2- GRU control the flow of data like the LSTM but without having to use memory and that was the reason of the GRU results. GRU algorithm trains faster and performs better than LSTM on less training data and that was the main reason of inconsistent results.
3- Bi-Directional-LSTM is not appropriate for all sequence prediction problems.
4- CNN-LSTM is used for predictions that are interrelated and have longer dependencies and because the datasets used did not have longer dependencies, the results were inconsistent and inaccurate.
5- In general, according to the results shown, multiple nodes dataset provides better accuracy for motion prediction than single node dataset.

5.5 Summary

This chapter presents some of the deep learning algorithms (RNN, GRU, Bi-Directional LSTM, LSTM, CNN-LSTM and Stacked LSTM) that can be used for mobility pattern prediction. Based on the original datasets that were created and collected form virtual Ad Hoc mobile cloud network with a realistic RWM mobility model with three data traffic rates scenarios. In this work, three new datasets have been formed namely DATA-Comp, DATA-Single, and DATA-Multiple form the original datasets collected. Two experiments were presented. In the first experiment, the six selected algorithms were trained and tested with DATA-Comp datasets to predict the positions of (x, y), speed and energy parameters. Evaluating the results shows that Stacked-LSTM gives highly accurate results (Low RMSE). RNN and LSTM algorithms also provide very good accurate prediction values. However, the other three CNN-LSTM, Bi-Directional LSTM and GRU, algorithms show good prediction accuracy with respect to the positioning x, y and speed but not for the energy.

The second explement consists of two scenarios, Single node dataset with its datasets DATA-Single and Multiple nodes dataset with its datasets DATA-Multiple. Five datasets related to nodes (7, 15, 24, 33, 48) were selected randomly from the total of 60 nodes. The selected algorithms were trained and tested using these two models to predict the next 300-time steps for the four parameters x, y, speed, and energy per each node form the five selected. Results evaluated and compared to find suitable techniques that might show more accurate prediction results. Stacked LSTM, RNN and LSTM show Lower RMSE values for the next 300-time steps prediction for x, y, speed, and energy using DATA-Multiple dataset compared to the DATA-Single dataset. While in Bi-Directional-LSTM, RMSE values were lower when using DATA-Single dataset compared to DATA-Multiple dataset. In GRU and CNN algorithms, RMSE values were lower using DATA-Multiple dataset for x, y, and energy parameters. However, the speed parameters results were different as it was more accurate to predict the next 300-time steps for the speed using DATA-Single dataset.

Prediction accuracy results in this work influence the complexity of datasets used. The impact on prediction results for a moving node is different from the same prediction model's impact when the node position is the same per time. That also includes the speed and energy values' influences on prediction accuracy due to their inconsistent patterns. Furthermore, the various characteristics of the used algorithms have played a significant role in final accuracy results. In conclusion for experiment two, using all nodes datasets for training and testing algorithms gives higher accurate results regarding node movements and energy predication than using only individual node dataset to train and test the same algorithms to get the movements and energy prediction values of the same node.

Chapter 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

Cloud Networking, or sometime called as cloud-based networking, refers to the network resources and network management functionality that must be available to enable cloud computing. A good example of cloud networking is the provisioning of high reliability and high performance networking between the cloud provider and the user; this will include the traffic that passes between them. Cloud networking presents imposing challenges for an efficient and effective flow of data traffic through networks. There are many vital research projects concerning the management of cloud-based networks. The background research on the related areas of cloud computing networks illustrates the drawbacks and benefits of cloud networks. Traditional cloud-based networks composed of different interworking technologies (wired and wireless) reach a point where existing network traffic management, maintenance methods, and routing protocols will be no longer capable of carrying on with the raised level of traffic by the emerging new cloud applications. The use of adaptive cognitive concepts is an encouraging method to overcome these challenges. Hence, applying cognition to the cloud network will create a network that will respond to demanding applications' requirements more efficiently, effectively, and adaptively.

This thesis focused mostly on improving end to end cloud network performance by addressing the challenges shown in AMCCNs architectures as a working case. Link breakage, routing, and network information data analysis are the main areas selected in this thesis because of their adverse effects on data communications' overall performance in cloud networks and the services provided to the user. Throughout this thesis, adaptive approaches have been presented to help alleviate these issues by applying the concept of cognitive networks to the virtual mobile Ad Hoc cloud networks.

Cognitive Routing and data management in the Heterogeneous Mobile Cloud Computing Networks model (HMCCNs) have been presented. This new model optimises the utilisation of heterogeneous computing and network resources in cloud network environments by integrating different cloud network architectures into one workflow. In this work, HMCCNs has been proposed by created AMCCN that was integrated with MCCN to overcome routing path congestion and link breakage issues. Two approaches for traffic management, namely: optimal cloud model selection and routing decisions, were proposed. The first approach suggested using FAH model, and the second suggestion is using a cognitive SDN controller. The experiments' results show that using HMCCNs model concept leads to improvements in the network's end-to-end performance with respect to throughputs, network latency, and reducing power consumption of nodes. A further experiment has been implemented that shows the improvements in network latency when using an SDN network compared to a traditional network.

Moreover, this thesis's work highlighted traditional routing challenges in AMCCNs and proposed an adaptive IHFN routing protocol. The new proposed protocol and the other protocols AODV, Fuzzy Logic-A and Fuzzy Logic-B, have been evaluated and compared. From the simulation results, it can be seen that IHFN protocol provides better results than the other three protocols AODV, Fuzzy logic-A, and Fuzzy Logic-B in terms of network throughput and packet delivery ratio. There were improvements in network performance firstly because the new hybrid protocol was designed based on fuzzy logic concept that offered a better way to select the reliable routing path using essential parameters like node movements and battery energy values, and secondly, by integrated that fuzzy logic concept with a neural network system, which added an intelligent awareness capability of nodes future movement and resources usage patterns also networks future states information to the routing process. The dynamic topology characteristic is responsible for the unsatisfactory performance of several routing algorithms in the cloud network, especially in the mobile cloud network. In contrast to other routing algorithms, the hybrid fuzzy-neural protocol is based on customising matrices related to mobility and nodes energy values and pattern predictions that gives extra features when making routing decisions. Finally, in this thesis, while there are many benefits to implementing machine learning in different areas, including pattern prediction systems, it is also important to know the potential limitations to its implementations and the impact of datasets complexity in prediction results accuracy. For those reasons, a new adaptive work was presented based on a dataset collected from the simulation of mobile Ad Hoc network with a realistic RWM mobility model with three data traffic rates scenarios. Three datasets have been formed from the original datasets that were used in two experiments. The performance behaviours of six selected DL algorithms have been evaluated and compared to predict the next steps of position, speed, and residual battery energy values of mobile nodes. Results were analysed concerning pattern prediction accuracy and have shown differences in algorithms performance. A second experiment highlighted the effect of using either single node dataset or multiple nodes dataset techniques on prediction accuracy results when applied on the same selected DL algorithms.

6.2 Direction to Future Work

All experiments and scenarios in this thesis were designed and presented to show the important changes that cognitive algorithms bring to cloud networks in order to improve the network's performance. The outcomes of introduced cognitive algorithms applied to HMCCNs in general and to VAMCCNs in particular have been presented.

Results were satisfactory with cognition network when compared with a network without cognitively. However, there are further works that can implement in the future.

We identify the following areas and topics of future work.

- Create scenarios using different traffic with multiple source nodes and multiple destination nodes in order to show the general suitability and applicability of our proposed integrated fuzzy logic and neural network routing algorithms.
- Using different mobility models with the new hybrid protocol presented and to compare its effect on network performance.
- Using optimisation techniques like Particle Swarm Optimisation (PSO), genetic algorithms (GA) or Ant Colony Optimisation (ACO) to make further improvements to the prediction system used.
- Currently, each cloud provider makes its own infrastructure to serve efficient cloud services to the users anywhere and anytime. Therefore, all the management is performed in a centralised way. It will be an excellent approach if some providers integrate to enable each one of these providers to access and use the infrastructure of the others; this will help to reduce the cost of deployments and will achieve better efficient utilisation of the whole available resources. In addition, it will allow the user to migrate and get services from more than one cloud provider.

References

[1] A. Yousefpour, G. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, “Alone needs to know about fog computing and related edge computing paradigms: A complete survey,” Journal of system architecture, vol. 98, pp. 289-330, 2019.

[2] B. Zhang, N. Mor, J. Kolb, D.S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E.A. Lee, and J. Kubiatowicz, “The cloud is not enough: saving IoT from the cloud,” HotStorage, 2015.

[3] P. Mell, and T. Grance. “The NIST Definition of Cloud Computing,” Technical Report, National Institute of Standards and Technology, Gaithersburg, MD, USA, September 2011.

[4] X. Yang, T. Pan, and J. Shen, "On 3g mobile e-commerce platform based on cloud computing," Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference, pp. 198-201, 2010.

[5] R. Ferzli and I. Khalife, "Mobile cloud computing educational tool for image/video processing algorithms," Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), IEEE, pp. 529-533, 2011.

[6] W. Zhao, Y. Sun, and L. Dai, "Improving computer basis teaching through mobile communication and cloud computing technology," Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 1. IEEE, pp. 452, 2010.

[7] G. A’Angelo, and S. Ferretti, “ Cloud for Gaming,” Encyclopedia of Computer Graphics and Games book, pp.1-6, may 2015.

[8] N. Kaur, A. and Verma, “Efficient path Establishment Algorithm in MANETs using Bio-Inspired Techniques,” International Journal of the Advance Research in Computer Science and Software Engineering, vol. 7, Aug. 2017.

[9] C. Doukas, T. Pliakas, and I. Maglogiannis, “Mobile healthcare information management utilizing cloud computing and android os,” Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pp. 1037-1040, 2010.

[10] W. T. Tang, C. M. Hu, and C. Y. Hsu, “A mobile phone based homecare management system on the cloud,” Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, vol. 6. IEEE, pp. 2442-2445, 2010.

[11] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” In proceeding of Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587-1611, 2013.

[12] C. Schindelhauer, “Mobility in Wireless Networks,” In proceeding of International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), pp. 100-116, 2006.

[13] T. Alam and B. K. Sharma, “Examining the movements of mobile nodes in the real world to produce accurate mobility models,” In proceeding of the International Journal of Engineering Science and Technology, vol. 2, Sep. 2010.

[14] W. Stallings, “Foundation of Modern Networking SDN, NFV, QoE, IoT and Cloud,” Book, Pearson Education, Inc. 2016.

[15] L. Zhong, B. Wang, and H. Wei, “Cloud Computing Applied in the Mobile Internet,” In proceedings of the 7th International Conference on Computer Science and Education (IEEE/ICCSE), pp. 218-221, 2012.

[16] D. Huang and H. Wu, "Chapter 1 - Mobile Cloud Computing Taxonomy,” Mobile Cloud Computing, Eds. Morgan Kaufmann, pp. 5–29, 2018.

[17] D. De, “Mobile Cloud Computing, Architectures, Algorithms and Applications,” Book, CRC Press, 2016.

[18] V. B. Souza, X. M. Bruin, E. M. Tordera, W. Ramirez, and S. Sanchez, “Towards Distributed Service Allocation in Fog-to-Cloud (F2C) Scenarios,” IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2016.

[19] S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog computing: Platform and Applications,” In proceeding of the 3rd IEEE workshop on Hot Topics in Web Systems and Technologies, pp. 73-78, Washington, 2015.

[20] G. klas, “Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium, ETSI MEC and Cloudlets,” Nov. 22, 2015.

[21] A. Ateya, A. Muthanna, I. Gudkova, A. Vybornova, and A. Koucheryavy, “Intelligent core network for Tactile Internet System,” ACM, p.15, Cambridge, Jul. 2017.

[22] S. Singh, Y. Chiu, Y. Tsai, and J. Yang, "Mobile Edge Fog Computing in 5G Era: Architecture and Implementation," IEEE International Computer Symposium (ICS), pp. 731-735, Dec. 2016.

[23] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A survey of the Emerging 5G Network Edge Architecture & Orchestration," IEEE Communications Surveys and Tutorials, 2017.

[24] A. Ahmed and E. Ahmed," A survey on mobile edge computing," In proceeding of the 10th International Conference in Intelligent System Control, pp.1-8, 2016.

[25] Y. Jaraweh A. Al-Doulat, and O. A. Alqudah, "The future of mobile cloud computing: Integrating cloudlet and mobile edge computing," In proceeding of the 23rd International Conference in Telecommunications, pp.1-5, 2016.

[26] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5G era," IEEE Communications Magazine, 55(2), pp. 192-200, Feb. 2017.

[27] Y.C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing: A Key technology towards 5G," ETSI White Paper, first edition, No. 11, Sep. 2015.

[28] M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: Distributed Edge Cloud for Data- Intensive Computing," IEEE International Conference on Cloud Engineering (IC2E), vol. 28, pp. 3229-3242, Jun. 2017.

[29] S. Wang, G. Tu, R. K. Ganti, and T. He, "Mobile Micro-Cloud: Application classification, Mapping and Deployment," In processing of the Annual Fall meeting of ITA (AMITA), New York, Oct. 2013.

[30] R. Szabo, K. Farkas, and B. Windt, "Measurements of real time transit feed architecture for mobile participatory sensing," IEEE in Wireless Days, IFIP, pp.1-4, 2013.

[31] W. S. Ansari, A. M. Alamri, M. M. Hassan, and M. Shoaib, "A survey on sensor-cloud: Architecture, applications and approaches," International Journal of Distributed Sensor Networks, pp. 1-18, 2013.

[32] W. Kurchl and W. Beer, "Combining cloud computing and wireless sensor networks," In the Proceeding of the 11th International conference Information Integration and Web-Based Applications and Services, pp. 512-518, 2009.

[33] T. Ahmed, “Modelling and simulation of routing protocol for ad hoc networks combining queuing network analysis and ANT colony algorithms,” University of Duisburg-Essen, Germany, 2005, pp.1-154.

[34] A.T. Zahary, and A. Ayesh, “A Comparative Study for Reactive and Proactive Routing Protocols in Mobile Ad -hoc Networks,” In proceeding of Journal of Intelligent Computing, vol. 1, pp. 20-29, 2010.

[35] J. P. Broach, R. Young, M. Manuell and C. Nichols,” Use of facial recognition software to identify disaster victims with facial injuries,” Disaster Medicine and Public Health Preparedness, 11(05), 1-5, April 2017.

[36] C. Ramming, “Cognitive network,” in DARPATech, 2004.

[37] M. Ashraf “A New Approach for Constructing Missing Features Values,” International Journal of Intelligent Information Processing, vol.3(1), 2012.

[38] S. Ezzati, H. R. Naji, A. Chegini, and P. Habibimehr, “Intelligent Firewall on Reconfigurable Hardware,” European Journal of Scientific Research. vol. 47, no.4 pp.509-516, 2010.

[39] J. Mitola, “Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio,” PhD thesis, Royal In statute of Technology (KTH) 2000.

[40] N. Abbas, Y. Nasser, and K. Ahmed, “Recent Advances on Artificial Intelligence and Learning Techniques in Cognitive Radio Networks,” EUURASIP Journal on Wireless Communications and Networking, Article number 174, 2015.

[41] X. Gao, G. Wu and T. Miki “End-to-End QoS Provisioning in mobile heterogeneous network,” IEEE Wireless Communications, vol. 11, pp. 24-34, June 2004.

[42] Al-Dulaimi and L. Al-Saeed, "An Intelligent Scheme for First Run Cognitive Radios,” NGMAST, pp.158-161, 2010.

[43] S. Al-Rubaye, A. Al-Dulaimi, L. Al-Saeed, H.S. Al-Raweshidy, and E. Kadhum, W. Ismail, “Development of Heterogeneous Cognitive Radio and Wireless Access Network,” 24th Meeting of Wireless World Research Forum, Penang, Malaysia, April 2010.

[44] W. Xia and L. Shen, "Joint resource allocation using evolutionary algorithms in heterogeneous mobile cloud computing networks," China Communications, vol. 15, no. 8, pp. 189-204, Aug. 2018.

[45] F. López-Pires and B. Baran, "Machine Learning Opportunities in Cloud Computing Data Centre Management for 5G Services," ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), pp. 1-6, 2018.

[46] N. Fatma and J. Qaddour, “Integration of SDN and 5G and Its Related Challenges and Security Issues,” International Journal of Networks and Communications, vol 9, no. 2, pp. 82-88, 2019.

[47] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, “OpenFlow: enabling innovation in campus networks,”, In proceeding of SIGCOMM, 2008.

[48] D. Shetty, S. Navi, and M. R. Ahmed, “Diving Deep into Deep Learning: History, Evolution, Types and Applications,” The International Journal on Media Management, Jan. 2020.

[49] A. Azzouni, R. Boutaba, and G. Pujolle, “NeuRoute: Predictive Dynamic Routing for Software-Defined Networks,” In proceeding of the 13th International Conference on Network and Service Management (CNSM), Nov. 2017.

[50] J. Wang, Y. Mio, P. Zhou, M.S. Hossain, and S. Mizanur, “A software defined network routing in wireless multi-hop network,” Journal of Network and Computer Applications. vol. 85, pp. 76-83, 2016.

[51] R. Kanagavelu and K. M. M. Aung, “SDN Controller Local Re-Routing to Reduce Congestion in Cloud Data Centres,” In proceeding of International Conference on Cloud Computing Research and Innovation, Oct. 2015.

[52] K. Kaur, S. Garg, G. Aujla, N. Kumar, J. Rodeigues, and M. Guizani “Edge Computing in the Industrial Internet of Things Environment: Software-Define Networks-Based Edge-Cloud Interplay,” IEEE Communications Magazine, vol. 56, Issue 2, Feb. 2018.

[53] A. Almohaimeed and A. Asaduzzaman, “Introducing Edge Controlling to Software Define Networking to Reduce Processing Time,” IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Jan 2019.

[54] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, “Balance flow: Control Load Balancing for Open Flow Network,” In proceeding of IEEE 2nd International Conference on Cloud Computing and Intelligence System, Nov. 2012.

[55] N. Handigol, S. Seetharaman, N. McKeown, and R. Johari, “Plug-n-Serve: Load-Balancing Web Traffic Using OpenFlow,” In proceeding of the Conference ACM SIGCOMM, 2009.

[56] R. Wang, D. Butnariu, and J. Rexford.” OpenFlow-Based Server load Balancing Gone Wild,” In proceeding of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, Jan. 2011.

[57] P. Kaur, J. Chahal, and A. Bhandari, “Load Balancing in software Define Networking: A Review,” Journal of Computer Science and Technology, vol. 7, pp. 1-5, 2018.

[58] S. Ejaz, Z. Iqbal, P. Saha, B. Bukhari, A. Ali, and F. Aadil, “Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function,” IEEE Access, vol 7, April 2019.

[59] D. Das, J. Bapat, and D. Das, “SDN assisted self-organizing network architecture for multi-RAT networks and mobility prediction,” Journal wireless Networks, Issue 1, 2020.

[60] L. A. Zadeh, "Fuzzy logic," Computer, vol. 21, no. 4, pp. 83-93, April 1988.

[61] J. M. Keller, D. Liu, and D. B. Fogel, “Fundamentals of Computational Intelligence, Neural Networks, Fuzzy Systems and Evolutionary Computation” IEEE press Serious on Computational Intelligence, Book – Wiley, 2016.

[62] M. Voskoglou,” Fuzzy sets, Fuzzy Logic, and Their Applications,” MDPI Book, Mathmatics, 2020.

[63] L. Barolli, A. Koyama, S. Motegi, and S. Yokoyama, “An Intelligent Policing-Routing Mechanism Based on Fuzzy Logic and Genetic Algorithms” In proceeding of International Conference on Parallel and Distributed Systems, vol. 1, p. 390, Dec. 1998.

[64] B.P. Chaythanya and M. M. Ramya, “Fuzzy Logic Based Approach for Dynamic Routing in Manet,” In processing of International Journal of Engineering Research and Technology (IJERT), vol. 1, Issue 6, June 2014.

[65] E. Arzhmand and H. Rashid, “Expansions of Vehicular Cloud Service on Crossroads using Fuzzy Logic and Genetic Algorithms,” In proceeding of the 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), July 2015.

[66] T. Su, S. Wang, H. Vu, D. Ku, and J. Huang, “An Application of Fuzzy Theory to the Power Monitoring System in Cloud Environments,” International Symposium on Computer, Consumer and Control (IS3C), Xi'an, pp. 350-354, doi: 10.1109/IS3C.2016.97, 2016.

[67] E. Qafzezi, K. Bylykbashi, M. Ikeda, K. Matsuo, and L. Barolli, “Coordination and Management of Cloud, Fog, and Edge resources in SDN-VANETs using fuzzy logic: A comparison study for two fuzzy-based systems,” Internet of Things 11 Journal, 2020.

[68] J. Zhang, A. Liu, P. Hu, and J. Long, “A fuzzy-rule-based packet reproduction routing for sensor networks” International Conference of Distributed Sensor Networks, vol. 14(4), 2018.

[69] J. Kumar, A. K. Singh, and R. Buyya. "Self-directed learning-based workload forecasting model for cloud resource management," In proceeding of Information Sciences Journal, vol. 543, pp. 345-366, 2020.

[70] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, “An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment.” Procedia Computer Science, vol. 151, p.p. 519-526, 2019.

[71] U. Paul, M. Buddhikot, and S. R. Das, “Opportunistic traffic scheduling in cellular networks,” In proceeding of IEEE International Symposium on Dynamic Spectrum Access Networks, pp. 339-348, October 2012.

[72] R. Li, Z. Zhao, X. Zhou, and H. Zhang, “Energy saving scheme in radio access networks via compressive sensing-based traffic load prediction,” Transactions on Emerging Telecommunications Technologies, vol. 25, no. 4, pp. 468-478, 2012.

[73] Z. Li, J. Bi, and S. Chen, “Traffic prediction-based fast routing algorithms for wireless multimedia sensors networks,” International Journal of Distributed Sensor Networks, vol. 9, no. 5, 2013.

[74] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks, In mobile computing,” Imelinsky T. Korth H (eds). Kluwer Academic Publisher, MA, pp/ 153-181. 1996.

[75] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, “Multi-hop wireless ad hoc network routing protocols,” In proceeding of the ACM/IEEE international Conference on Mobile Computing and Networking (MOBICOM), PP 85-97, 1998.

[76] C. Chiang and M. Gerla, “On-Demand multicast in mobile wireless networks,” In proceeding of the IEEE International Conference on Network Protocol (ICIP), 1998.

[77] J. Garica and M. Spohn, “Source- tree routing in wireless networks,” In proceeding of the 7th International Conference on Network Protocols (ICNP), 1999.

[78] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Routing Protocols for the mobile ad hoc networks a comparative performance analysis,” In proceeding of the CAN/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp. 66-75, 1998.

[79] Y. Ko and NH. Vaidya, “Location -aided routing (LAR) in mobile as hoc networks,” In proceeding of the ACM/IEEE International Conference in Mobile Computing and Networking (MOBICOM, pp 66-75, 1998.

[80] T. Camp, J. Boleng and V. Davies,” A survey of mobility models for ad hoc network research,” In proceeding of the Wireless Communications and Mobile computing, pp. 483-502, 2002.

[81] M. Sanchez and P. Manzoni,” A java-based ad hoc networks simulator,” In proceeding of the SCS Western Multiconference Web-based Simulation Track, January 1999.

[82] A. Bar-Noy and M. Sidi,” Mobile users to update or not to update?,” In proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 570-576, 1994.

[83] J. Garcia-Luna-Aceves and EL/ Madrge, “A multicast routing protocol for ad-hoc networks,” In proceeding of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 784-792, 1999.

[84] I. Rubin and C. Choi, “Impact of the location area structure on the performance of signalling channels in wireless cellular networks,” In proceeding of the IEEE Communications Magazine, 35(2), 108-115, 1997.

[85] M. Zonoozi and P. Dassanayake, “User mobility modelling and characterisation of mobility pattern,” In proceeding of IEEE Journal on Selected Areas in Communications, 15(7), 1239-1252, 1997.

[86] B. Liang and Z. Hass, “Predictive distance-based mobility managements for PCS networks,” In proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 1999.

[87] X. Hong, M. Gerla, G. pie, and C. Chiang, “A group mobility model for ad hoc wireless networks,” In proceedings of the CAN International Workshop on Modelling and Simulation of Wireless and Mobile Systems (WSWIM), August 1999.

[88] E. Royer, PM. Melliar-Smith, and L. Moder, “An analysis of the optimum node density for ad hoc mobile networks,” In proceedings of the IEEE International Conference on Communications (ICC), 2001.

[89] Z. Hass, “A new routing protocol for reconfigurable wireless networks,” In proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC), pp. 562-565, Oct. 1999.

[90] C. Chiang, "Wireless Network Multicasting," PhD thesis University of California, LA, 1998.

[91] Md. Hossain and M. Atiquzzman, "Stochastic Properties and Applications of City section Mobility Model," In proceeding of the IEEE Global Telecommunications Conference (GLOBECOM), Honolulu, USA, 2009.

[92] P. Singh, T. Alam, and P. Kumar, "Searching Mobile Nodes Using Modified Column Mobility Model," In proceeding of the International Journal of Computer Science and Mobile Computing, vol 3, Issue.1, pp. 513-518, Jan. 2014.

[93] S. Dalu, M. Naskar, and C. Sakar, "Implementation of a Topology Control Algorithm for MANETs Using Nomadic Community Mobility Model," In proceeding of the IEEE 3rd International Conference of Industrial and Information System, India, 2008.

[94] J. Yoon, M. Liu, and B. Noble, "A general Framework to Construct Stationary Mobility Models for the simulation of Mobile Networks," In proceeding of IEEE Transactions on Mobile Computing, 5(7), pp. 860-871, Aug. 2006.

[95] J. Geetha and G. Ganapathy, "Reference Point Group Mobility and Random Waypoint Models Performance Evaluation of Manet Routing Protocols," In proceeding of the Journal of Computer System Networks and Communications, Jan. 2009.

[96] G. Singal, V. Laxmi, and V. Rao, "Moralism, Mobility prediction with link stability-based multicast routing protocol in MANETs," Wireless Networks, vol. 23, pp. 663-679, 2017.

[97] C. Xu and C. Xu, "Predicting Personal Transitional Location Based on Modified -SVM," In proceeding of the International Conference on Computer Science and Computational Intelligence (CSCI), pp. 340-244, Dec. 2017.

[98] M. De Para, URL: https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc, 2020.

[99] J. Chung, C. Gulechre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modelling," arXiv: 1412.555, 2014.

[100] J. Brownlee, "Machine Learning Mastery," https://machinelearningmastery.com/, 2017.

[101] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM network," In proceeding of IEEE International Joint Conference on Neural Networks, vol. 4, pp. 2047-2052, July 2005.

[102] A. B. Said, A. Erradi, H. A. Aly, and A. Mohammed, "Predicting COVID-19 Cases Using Bidirectional LSTM on Multivariate Time Series," Environmental Science and Pollution Research, vol 28, no 12, 2020.

[103] J. Wang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," In proceeding of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 2, January 2016.

[104] W. Creixell and K. Sezaki, "Mobility Prediction Algorithm for Mobile Ad Hoc Network Using Pedestrian Trajectory Data," In proceeding of the IEEE region 10th conference TENCON, vol. B, No. 1, pp. 668-671, 2004.

[105] H. Abu-Ghazaleh and A. Alfa, "Application of Mobility Prediction in Wireless Networks Using Markov Renewal Theory," In proceeding of the IEEE Trans. Veh. Technol., Vol. 59, No. 2, pp. 788-802, 2010.

[106] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," In proceedings of the IEEE International Conference on Computer Vision, pp. 4346-4354, 2015.

[107] D. Pavllo, D. Grangier, and M. Auli, "QuaterNet: A Quaternion-based Recurrent model for human Motion," British Machine Vision Conference (BMVC), 2018.

[108] J. Martinez, M. J Black, and J. Romero, "On human motion prediction using recurrent neural networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[109] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5226–5234, 2018.

[110] A. Dastjerdi, K. Bakar, and S. Tabatabaei, "Distributed Intrusion Detection in Clouds Using Mobile Agents," In proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Science, IEEE/ADVCOMP, pp. 175-180, 2009.

[111] J. Carolan and S. Gaede, "Introduction to Cloud Computing Architecture," SUN Microsystems Inc., pp.–1-40, 2009.

[112] Gartner, "Gartner Forecast Global Device Shipments Will Grow 0.9% in 2020," https://www.gartner.com/en/newsroom/press-releases/2019-01-21-gartner-forecasts-global-device-shipments-will-grow-0, Egham, UK, January 2020.

[113] W. Lee and M. Choi, "Web Application Framework for Customizable Mobile Cloud Services," In proceedings of the ICT Conference (IEEE/ICTC), pp. 510-515, 2012.

[114] V. Koufi, F. Malamateniou, G. Vassilascopoulos, and A. Prentza, "An Android-Enabled Mobile Framework for Ubiquitous Access to Cloud Emergency Medical Services," In proceedings of the 2nd Symposium on Network Could Computing and Applications (IEEE/NCCA), pp. 95-101, 2012.

[115] R. Klauck, J. Gaebler, M. Kirsche, and S. Schoepke, "Mobile XMPP and Cloud Service Collaboration an Alliance of Flexible Disaster Management," In proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Work-sharing (CollaborateCom), pp. 201-210, 2011.

[116] B. Girod, V. Chandrasekhar, R. Grzeszczuk, and Y.A. Reznik, "Mobile Visual Search Architectures, Technologies, and the Emerging MPEG standard," IEEE Multimedia, vol. 18, no. 3, pp. 86–94, 2011.

[117] G. Cai-dong, L. Kaa, W. Jaian-ping, F. Ying-li, L. Jin-Xiang, X. Chang-shui, S. Mao-xin, and L. Zhao-bin, "The Investigation of Cloud-Computing –based Image Mining Mechanism in Mobile Communication WEB on Android," In proceeding of the 9th International Conference on Grid and Cooperative Computing (IEEE/GCC), pp. 408-411, 2010.

[118] T. Verblen, P. Simoens, F. Turck, and B. Dhoedt, "Cloudlet: Bringing the Cloud to the Mobile User," In proceedings of the 3rd ACM Workshop on Mobile Cloud Computing and Services (MCS), pp. 29–36, 2012.

[119] G. Huerta-Canepa, and D. Lee, "A virtual cloud computing provider for mobile devices," In proceeding of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. ACM, Article. 6, 2010.

[120] R.Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using Bandwidth Data to Make Computation Offloading Decisions," In proceedings of the International Symposium on Parallel and Distributed Processing (IEEE/IPDPS), pp. 1-8 ,2008.

[121] M. Shiraz and A. Gani, "Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines," In proceedings of the International Conference of Information and Computer Networks (ICICN), pp. 11 ,2012.

[122] M. Armbrust, A. Fox, R. Griffth, A. Joseph, R. Katz, A. Konwiniski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia "A View of Cloud Computing," In

proceedings of the International Conference communications (CACM), pp. 50-58 ,2010.

[123] K. Sripanidkulchai, S. Sahu, Y. Ruan, A. Shaikh, and C. Dorai, "Are clouds ready for large distributed applications?," SIGOPS Oper. Syst. Rev., vol. 44, pp. 18–23, 2010.

[124] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," In proceedings of the 9th International Conference on Internet measurement, IMC (SIGCOMM), pp. 280–293, 2009.

[125] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, "Opportunistic Mobility with Multipath TCP," In proceedings of the 6th international workshop on MobiArch, (ACM MobiArch), pp. 7-12, 2011.

[126] H. Harada, H. Murakami, K. Ishizu, S. Filin, Y. Saito, H. Tran, G. Miyamoto, M. Hasegawa, Y. Murata, and S. Kato, "A Software Defined Cognitive Radio System," Proceedings of the 7th International Conference on Global Telecommunications, (GLOBECOM), pp. 26–30, 2007.

[127] Y. T. Larosa, J. Chen,D. Deng, and H. Chao "Mobile Cloud Computing Service Based on Heterogeneous Wireless and Mobile P2P Networks," In proceedings of the 7th International Conference on Wireless Communications and Mobile Computing (IWCMC), pp. 661–665, 2011.

[128] S. C. Shah, "Mobile Edge Cloud: Opportunities and Challenges," In proceeding of International Conference on Computational Science and Computational Intelligence", vol. 70, PP. 59-63, Dec. 2017.

[129] J. C. Guevara, L. F. Bittencourt, and N.L.S. da Fonseca, "Class of Service in Fog Computing," In proceeding of IEEE 9th Latin-American Conference on Communications (LATINCOM), Nov. 2017.

[130] T. L. Saaty, "The analytic hierarchy process," McGraw-Hill book, New York, 1980.

[131] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using Bandwidth Data to Make Computation Offloading Decisions," In proceeding of the International Parallel and Distributed Processing Symposium (IPDPS), High -Performance Grid Computing Workshop, April 2008.

[132] H. Liang, D. Huang, and D. Peng, “On Economic Mobile Cloud Computing Model,” In proceeding of the International Workshop on Mobile Computing and Clouds (MobiCloud In conjunction with MObiCASE), 2010.

[133] K. Kumar and Y. Lu, “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?,” IEEE Computer, vol. 43, pp. 51-56, April 2010.

[134] S. Ou, K. Yang, A. Liotta, and L. Hu, “Performance Analysis of Offloading System in Mobile Wireless Environments,” In proceeding of IEEE International Conference on Communications (ICC07) Glasgow, UK, pp. 24-27, June 2007.

[135] C. Singla and S. Kaushal “Cloud path selection using Fuzzy Analytic Hierarchy Process for offloading in Mobile Cloud Computing,”, (RAECS UIET), Dec. 2015.

[136] S. Haykin, M. Fatemi, P Setoodeh, and Y. Xue, “Cognitive Control,” In proceeding of the IEEE, vol. 100(12), PP. 3156-3169, 2012.

[137] T. A. Assegie, “Software defined network emulation with OpenFlow protocol,” In proceeding of International Journal of Advance in Applied Sciences, vol. 9, Issue 1, March 2020.

[138] V. Hariharasudhan, E.K. Lee, I. Rodero, and D. Pompili, “Uncertainty-aware autonomic resource provisioning for mobile cloud computing,” In proceeding of IEEE Trans. Parallel Distribution System, vol. 26, pp. 2363-2372, 2015.

[139] S. C. Shah, “Recent advance in mobile grid and cloud computing,” In proceeding of Intelligent Automation Soft Computing,”, Vol. 24(2), pp. 1-13, Feb. 2017.

[140] S. Ghasemi-Falavarjani, M. Nematbakhsh, and B.S. Ghafarokhi, “Context-aware multi-objective resource allocation in mobile cloud,” In proceeding of Computer and Electrical Engineering, vol. 44, pp. 2018-240, 2015.

[141] S.C. Shah and M.S. park, “An energy-efficient resources allocation scheme for mobile ad hoc computational grids,” In proceeding of Journal of Grid Computing, vol. 9, pp. 303-323, 2011.

[142] Ericsson Mobility Report Data and Forecasts,https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast, Nov. 2021.

[143] S. Verma, P. Nayak and R. Agarw, “Energy efficient routing in mobile ad hoc network based on AODV protocol,” In proceeding of International Journal for Computer and Science, Issues 9(6), 2011.

[144] M. Abolhassan, T. Wyscocki, and E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks,” Ad Hoc Networks Journal, vol. 2, Issue 1, pp, 1-22, 2004.

[145] S. Garg, P. Kansal, and V. K. Sharma, "A fuzzy Logic Based Network Dependent Routing Algorithm for Ad Hoc Wireless Network," International Journal of Engineering Research and Applications (IJERA), vol. 1, Issue 1, PP. 001-010, January 2011.

[146] Z. Zinonos, C. Chrysostomou, and V. Vassilliou, "Mobility Management in WSNs Using Fuzzy Logic: An Industrial Application Scenario," In proceeding of IEEE 8th International Conference on Distributed Computing in Sensor System (DCOSS) , China, May 2012.

[147] A. Dana, G. Ghalavand, A. Ghalavand, and F. Farokhi, "A reliable routing algorithm for Mobile ad-hoc Networks based on fuzzy logic," International Journal of computer Science Issues (IJCSI), vol. 8, Issue 3, No 1, May 2011.

[148] S. Marwaha, D. Srinivasan, C. Tham, and A. Vasilakos, "Evolutionary fuzzy multi objective routing for wireless mobile ad hoc networks," In proceeding of Evolutionary Computation (CEC2004), Congress 2, pp 1964-1971, 2004.

[149] M. Turcanik "Network Routing by Artificial Neural Network," In proceeding of the Communications and Information System Conference (MCC), 2012.

[150] P. Srivastava and R. Kumar, "A New QoS-Aware Routing Protocol for NANET Using Artificial Neural Network," Journal of Computing and Information Technology, vol. 24, No. 3, September 2016.

[151] A. Al-Molegi, M; Jabreel, and B. Ghaleb, "STF-RNN: Space-Time Features-Based Recurrent Neural Network for Prediction People's Next Location," IEEE Symposium Serious Computational Intelligence (SSCI), pp. 6-9, Dec. 2016.

[152] A. Azari, J. S. Harsini, and F. Lahouti, "Performance Analysis of ad-hoc Routing in Heterogeneous Clustered Multi-Hop Wireless networks," In proceeding of the Computer Networks, vol. 80, pp. 143-154, Apr. 2015.

[153] M. K. Denko, "Mobility Prediction Schemes in Wireless Ad Hoc Networks," In proceeding of Advanced Wired and Wireless Networks, Academic Publisher, pp. 171-186, Boston, 2001.

[154] M. Yu and V. Leung," Mobility-Based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks," In proceedings of the Computer Networks, vol. 38, pp. 557-589, 2002.

[155] J. Ananthi and V. Ranganathan, "A Review: On Mobility Prediction for Wireless Networks," In proceeding of the Int. Journal of Emerg. Technol. Adv. Eng., vol. 3, No. 4, pp. 891-902, 2013.

[156] M. Makhlouf, L. Nagy, and P. Vajsar, "Prediction of Movements of Wireless Nodes in Mobile ad -hoc Networks," In proceeding of the 4th IEEE Int. Conference on Cognitive Info communications, pp. 115-120, 2013.

[157] N. Makhlouf, "Enhancing On-Demand Multicast Routing protocols using mobility prediction in mobile ad -hoc Network," In proceeding of the International Journal of Advance Telecommunication Electrotechnology Signals System, vol. 3, no. 2, pp. 40-43, Aug. 2014.

[158] B. Hawelka, I. Sitko, P. Kazakopulos, and E. Beinat, "Collective Prediction of Individual Mobility Traces for Users with Short Data History," In proceeding of the PLoS One, vol. 12, no. 1, p. e0170907, Jan. 2017.

[159] M. Papandrea and S. Giordano, "Location Prediction and Mobility Modelling for Enhanced Localization Solution," Journal Ambient Intell. Humanis. Comput. vol. 5, no. 3, pp. 279-295, Jan 2014.

[160] S. Suraj, R. Tapaswi, S. Yousef, K. K. Pattanaik, and M. Cole, "Mobility Prediction on Mobile Ad Hoc Networks Using A Lightweight Genetic Algorithm," In proceeding of the wireless Networks, vol. 22, no. 6, pp. 1797-1806, 2016.

[161] T.T. Son, H. Le Minh, G. Sexton, N. Aslam, and Z. Ghassemlooy, "Bayesian Model for Mobility prediction to support Routing in Mobile Ad-Hoc Networks," In proceeding of the IEEE 24th Annual International Symposium on Personal Indoor, and Mobile Radio Communications (PIMRC), pp. 3186-3190, 2013.

[162] L. Ghouti "Mobility Prediction in Mobile Ad Hoc Network Using Neural Learning Machine," Simul. Model. Pract. Theory, vol. 66, pp. 104-121, Aug. 2016.

[163] C. Comito, D. Falcone, and D. Talia, "Mining Human Mobility Patterns from Social Geo-tagged Data," In proceeding of the Pervasive Mobile Computing, vol. 33, pp. 91-107, 2016.

[164] T. K. Vu and S. Kwon," On-Demand Routing Algorithm with Mobility Prediction in The Mobile Ad-Hoc Networks," In proceeding of the Netw. Internet Archit., p. 10, Sept. 2016.

[165] N. Makhlouf, "Exploiting Neural Network for Mobility Prediction in Mobile Ad Hoc networks," In proceedings of the Elecktrorevue Journal, vol. 18, no. 3, pp. 63-67, June 2016.

[166] S. Islam, A. A. Hashim, M. H. Habaebi, and M. K. Hasan, "A simulation analysis of mobility models for network mobility environments," In proceeding of the APRN Journal of Engineering and Applied Science, vol. 10, no. 21, Nov. 2015.

[167] C. Bettstetter, H. Hartenstein, and X. Perez-Costa, “Stochastic properties of the random waypoint mobility model,” In proceeding of the Wireless network, vol. 10, no. 5, pp. 555-567, 2004.

[168] M. S. Hossain and M. Atiquzzaman, “Cost analysis of mobility protocols,” In proceedings of the Telecommunication System, vol. 52, no. 4, pp. 2271-2285, 2013.

[169] A. S. Reza, P. K. Chowdhury, M. Atiquzzaman, and W. Ivancic, “Signalling cost analysis of SIEMO: seamless end-to-end network mobility,” In proceeding of the first ACM/IEEE International workshop on Mobility in the evolving internet architecture, 2006.

[170] R. L. Moila and M. Velempini, “Enhanced AODV protocol using QoS constraints to improve routing in Cognitive Radio Ad Hoc Network,” In processing of IEEE International Conference on Artificial Intelligence, Big Data Computing and Data Communication Systems (ICAABCD), Aug. 2020.

[171] N. Abbas, M. Ikan, and E. Ozen, “Fuzzy approach to improving route stability of the AODV routing protocol,” (EURASIP) Journal on Wireless Communications and Networking, pp. 235, 2015.

[172] A. Dana and M. H. Babaei, “A fuzzy Based Stable Routing Algorithms for MANET,” International Journal of Computer Science Issues (IJCSI), vol. 8, Issue 1 Jan. 2011.

[173] A. Sadollah, “Fuzzy logic based in Optimization Methods and control System and Its Applications,” IntechOpen books, chapter 1, Oct. 2018.

[174] Y. Yayeh, H. Lin, G. Berie, A. B. Adege, L. Yen, and S. Jeng, “Mobility predication in Mobile Ad-Hoc Network Using Deep learning,” In processing of IEEE international Conference on Applied System Innovation (ICASI), Japan, 2018.

[175] H. Si, Y. Wang, J. Yuan, and X. Shan, “Mobility Prediction in Cellular Network Using Hidden Markov Model,” In proceeding of 7th IEEE Consumer Communications and Networking Conference (CCNC), pp. 1130-1134, Feb. 2010.

[176] Y. Qiao, Z. Si, and Y. Zhang, “A Hybrid Markov-based Model for human Mobility Prediction,” In proceeding of Neurocomputing, vol. 278, pp. 99-1-0, 2018.

[177] D. Stynes, K. Brown, and C. Sreenan, “A probabilistic approach to user mobility prediction for wireless services,” International Wireless Communications and Mobile Computing Conference (IWCMC), PP. 120-125, Sept. 2016.

[178] A. Asahara, K. Maruyama, A. Sato, and K. Seto, “Pedestrian- Movement Prediction Based on Mixed Markov- Chain Model,” In proceedings of the 19th ACM

SIGSPATIAL International Conference on Advances in Geographic Information System, pp. 25-33, Nov. 2011.

[179] S. Gambs, M. Killijian, and M. N. Cortez, “Next Place Prediction Using Markov Chains,” In proceedings of the 1st Workshop on Measurement Privacy and Mobility, 2012.

[180] H. Kaaniche, and F. Kamoun. "Mobility prediction in wireless ad hoc networks using neural networks." Journal of Telecommunications, Vol. 2, Issue 1, April 2010.

[181] N. Aljeri and A. Boukerche, “A performance Evaluation of Time-Series Mobility Prediction for Connected Vehicular Networks,” In proceeding of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 127-131, 2020.

[182] “5 Most Popular AI Programming Languages in 2021”, https://lvivity.com/ai-programming-languages .

Appendix A

RMSE of POS X for Node 7:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	61.573222263641156	123.05958576129149
CNN_LSTM	102.38391004837169	43.680930890927165
LSTM	61.29386084509197	121.19202780387786
BiDirectonal_LSTM	20.09891101158634	14.39416744173342
GRU	20.72135914716499	28.115337727983484
RNN	62.79970610404073	124.73436529393986

RMSE of POS Y for Node 7:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	159.58409213220474	112.44895866353313
CNN_LSTM	133.5151398503312	55.27212978572837
LSTM	110.5818274520263	111.07358226558765
BiDirectonal_LSTM	25.020331438683645	58.76953640939727
GRU	24.499603953670736	76.92776470754842
RNN	66.96652484100159	114.4957716704598

RMSE of Speed for Node 7:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	0.6541958912251449	1.0268848149834653
CNN_LSTM	1.1540222154722959	0.8333292139463294
LSTM	0.6303590610354146	0.6533756398375985
BiDirectonal_LSTM	4.287745319186575	0.5303980048617654
GRU	0.6618241009181162	0.5339567818428066
RNN	0.6145220797746913	0.5535005742626309

RMSE of Energy for Node 7:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	581.0397307458476	60.06379720653752
CNN_LSTM	706.6520316284772	310.3891916569228
LSTM	580.8618119442879	62.00322534689446
BiDirectonal_LSTM	5284.45272634067	185.52940873921997
GRU	3925.1928444891723	166.69175205205582
RNN	580.9987205580067	59.352369291437384

RMSE – Single-Multiple datasets comparison for Node 7.

Appendix B

RMSE of POS X for Node 15:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	61.573222263641156	147.43747071124255
CNN_LSTM	102.38391004837169	90.41380299041376
LSTM	61.29386084509197	145.02895512368397
BiDirectonal_LSTM	20.09891101158634	163.36525419849164
GRU	20.72135914716499	154.410018259992
RNN	62.79970610404073	147.97765027339327

RMSE of POS Y for Node 15:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	159.58409213220474	177.6874476825636
CNN_LSTM	133.5151398503312	57.03869246192235
LSTM	110.5818274520263	176.13299415130209
BiDirectonal_LSTM	25.020331438683645	99.87220240844042
GRU	24.499603953670736	86.12715615297384
RNN	66.96652484100159	177.3211126421705

RMSE of Speed for Node 15:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	0.6541958912251449	1.1278949191787995
CNN_LSTM	1.1540222154722959	1.5766812419985001
LSTM	0.6303590610354146	0.6487531115894236
BiDirectonal_LSTM	4.287745319186575	0.691609107750066
GRU	0.6618241009181162	0.742468684615478
RNN	0.6145220797746913	0.7463269325836593

RMSE of Energy for Node 15:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	581.0397307458476	479.09675248030686
CNN_LSTM	706.6520316284772	113.50577142691449
LSTM	580.8618119442879	477.91572689510866
BiDirectonal_LSTM	5284.45272634067	417.6449722649848
GRU	3925.1928444891723	247.4159422401167
RNN	580.9987205580067	480.1756658362799

RMSE – Single-Multiple datasets comparison for Node 1.

Appendix C

RMSE of POS X for Node 24:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	61.573222263641156	256.9820604512129
CNN_LSTM	102.38391004837169	84.6837525768609
LSTM	61.29386084509197	254.69284070096347
BiDirectonal_LSTM	20.09891101158634	59.56913362602231
GRU	20.72135914716499	43.43431219797655
RNN	62.79970610404073	258.9419030396612

RMSE of POS Y for Node 24:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	159.58409213220474	108.42258836351884
CNN_LSTM	133.5151398503312	96.10702572600522
LSTM	110.5818274520263	106.49661707105462
BiDirectonal_LSTM	25.020331438683645	20.583328918825075
GRU	24.499603953670736	76.90056023394429
RNN	66.96652484100159	107.97802342927213

RMSE of Speed for Node 24:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	0.6541958912251449	2.469713315725961
CNN_LSTM	1.1540222154722959	1.857852168493383
LSTM	0.6303590610354146	1.1127101050166064
BiDirectonal_LSTM	4.287745319186575	0.830033255487566
GRU	0.6618241009181162	0.8206279589873368
RNN	0.6145220797746913	1.3328281422075363

RMSE of Energy for Node 24:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	581.0397307458476	987.2102492363715
CNN_LSTM	706.6520316284772	851.3483584139731
LSTM	580.8618119442879	984.7883505621235
BiDirectonal_LSTM	5284.45272634067	510.00748679456626
GRU	3925.1928444891723	671.9898259079368
RNN	580.9987205580067	987.8088258184881

RMSE – Single-Multiple datasets comparison for Node 24.

Appendix D

RMSE of POS X for Node 33:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	61.573222263641156	193.67341798856376
CNN_LSTM	102.38391004837169	105.95222352728548
LSTM	61.29386084509197	191.1184123890409
BiDirectonal_LSTM	20.09891101158634	64.8120493242895
GRU	20.72135914716499	67.74018362385752
RNN	62.79970610404073	195.16416111057606

RMSE of POS Y for Node 33:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	159.58409213220474	218.68933800304248
CNN_LSTM	133.5151398503312	81.22974177614887
LSTM	110.5818274520263	216.42541336567507
BiDirectonal_LSTM	25.020331438683645	92.39650091292256
GRU	24.499603953670736	92.94813986092797
RNN	66.96652484100159	218.97301322950645

RMSE of Speed for Node 33:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	0.6541958912251449	0.9740930376646503
CNN_LSTM	1.1540222154722959	2.6932398426703514
LSTM	0.6303590610354146	1.0387092979016215
BiDirectonal_LSTM	4.287745319186575	1.0894530962946702
GRU	0.6618241009181162	0.8789905330740273
RNN	0.6145220797746913	0.8305385868192453

RMSE of Energy for Node 33:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	581.0397307458476	477.4955265662695
CNN_LSTM	706.6520316284772	487.07252224705024
LSTM	580.8618119442879	475.967697109381
BiDirectonal_LSTM	5284.45272634067	607.504467641732
GRU	3925.1928444891723	478.0291356886364
RNN	580.9987205580067	479.1608525772863

RMSE – Single-Multiple datasets comparison for Node 33.

Appendix E

RMSE of POS X for Node 48:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	61.573222263641156	321.76561370280916
CNN_LSTM	102.38391004837169	67.47253859302006
LSTM	61.29386084509197	318.3458609519366
BiDirectonal_LSTM	20.09891101158634	54.707142572200944
GRU	20.72135914716499	44.31980466840187
RNN	62.79970610404073	323.5277168268766

RMSE of POS Y for Node 48:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	159.58409213220474	94.63089169313274
CNN_LSTM	133.5151398503312	21.184180156371706
LSTM	110.5818274520263	92.52998741510957
BiDirectonal_LSTM	25.020331438683645	7.521945636877024
GRU	24.499603953670736	6.01852593491646
RNN	66.96652484100159	96.8788162993617

RMSE of Speed for Node 48:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	0.6541958912251449	1.5762097274150835
CNN_LSTM	1.1540222154722959	3.7928538141786654
LSTM	0.6303590610354146	1.4392880143416922
BiDirectonal_LSTM	4.287745319186575	1.2910646915607835
GRU	0.6618241009181162	1.478184171109786
RNN	0.6145220797746913	1.6784535691618139

RMSE of Energy for Node 48:

Algorithm	Multiple Model	Single Model
Stacked_LSTM	581.0397307458476	599.8623182136664
CNN_LSTM	706.6520316284772	1541.0387866019682
LSTM	580.8618119442879	597.2800637900656
BiDirectonal_LSTM	5284.45272634067	1477.5277026242852
GRU	3925.1928444891723	1255.0653572794781
RNN	580.9987205580067	601.1948940890286

RMSE – Single-Multiple datasets comparison for Node 48.