Big Data Analytics for Prediction of Mobile Users Movement using Neural Networks

By: **Selamawit Engliz**

Advisor: **Dr. –Eng. Yihenew Wondie**

A thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment for the Degree of Masters of Science in Communication Engineering

(Communication and Networking Track)

Addis Ababa University

Addis Ababa Institute of Technology

Addis Ababa, Ethiopia

June, 2019

Certification

This is to certify that the thesis prepared by Selamawit Engliz, entitled: *Big Data Analytics for Prediction of Mobile Users Movement using Neural Networks* and submitted in partial fulfillment of the requirements for the degree of Masters of Science in Communication Engineering complies with the regulations of the university and meets the accepted standards with respect to originality and quality.

Approved by board of examiners.

Dr. -Eng. Yihenew Wondie		
Thesis Advisor	Signature	Date

Dr. -Ing. Dereje H. Mariam		
Examiner	Signature	Date

Dr. Beneyam Berehanu		
Examiner	Signature	Date

School Chair Person

Dr. Yalemzewd Negash

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with proper citation of sources. I, the undersigned, declare that this thesis has not been presented for a degree in any other university.

Selamawit Engliz

Acknowledgments

I would like to express my deepest gratitude to:

- Almighty God for helping me in this research and throughout my life.
- My research advisor Dr. –Eng. Yihenew Wondie, for his immense knowledge, time, prompt response, constructive feedback and follow up in this research. It has helped me to enrich my knowledge and critical thinking skills.
- Addis Ababa University (AAU) library director Mr. Mesfin Gezahegn, for his patience and dedication to obtain the Mobile Data Challenge (MDC) data set.
- Prof. Cristian-Liviu, for the supportive comments he gave me to this work.
- My family for their support and guidance throughout my life.
- Last but not least all who helped me in one way or another during this research.

Abstract

The rise of data warehouses and the rise of multi-media, social media and the Internet of Things (IoT) generate an increasing Volume of structured, semi-structured and unstructured data. Towards the investigation of these large Volumes of data, Big Data and data analytics have become emerging research fields attracting the attention of academia, industry and governments. Researchers, entrepreneurs, decision makers and problem solvers view 'Big Data' as an important tool used to revolutionize various industries and sectors, such as business, health-care, retail, research, education and public administration.

In this context, a general view on analysis of Big Data, especially in telecommunications industry is proposed. In order to allocate scarce resources efficiently the location of mobile users should be predicted. So, this work focuses on analysis of data for mobile users movement prediction in telecommunications network. The objective of this work is to process and analyze obtained samples of OpenCelliD data by means of Neural Network and provide as accurate mobile users movement prediction as possible. More specifically, Modified Levenberg-Marquardt (LM) algorithm is presented as an effective algorithm for movement prediction. Obtained result from prediction is optimized by iteration method designed for finding the best possible combination of Neural Network parameters. Efficiency of mobile users movement prediction is verified by simulation in Matrix Laboratory (**MATLAB**). Simulation results show sufficient accuracy for wide use of prediction for mobile networks optimization or services exploiting prediction of mobile users movement. Measured results fully reflect real solution for telecommunications industry and can help to plan activities connected with mobile users movement in a given area.

Keywords

Big Data, Data Analytics, Location-Based Services (LBS), Telecommunications, Prediction, Neural Network, Non-Linear Autoregressive with External Input Neural Network (NARXNN), Dataset, Training, Optimization

This page is left intentionally blank.

Contents

List of Figures

List of Tables

List of Abbreviations

1G	First Generation
2G	Second Generation
2.5G	2.5 Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AC	Authentication Center
AMPS	Advanced Mobile Phone System
AMTS	Advanced Mobile Telephone System
ANN	Artificial Neural Network
BR	Bayesian Regularization
BS	Base Station
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CN	Core Network
CRM	Customer Relationships Management
CSI	Channel State Information
CSP	Communication Service Provider
DRNC	Drifting Radio Network Controller
DVB	Digital Video Broadcasting
EIR	Equipment Identity Register
ETSI	European Telecommunication Service Institute
FCC	Federal Communications Commission
FDMA	Frequency Division Multiple Access
FFNN	Feed-Forward Neural Network

FIR	Finite Impulse Response
GGSN	Gateway GPRS Support Node
GIS	Geographical Information System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HDTV	High Definition Television
HPY	Hierarchical Pitman-Yor
HLR	Home Location Register
HSPDA	High Speed Packet Data Access
IDIAP	Instituto de Investigación Agropecuaria de Panamá
IEEE 802.16	Institute of Electrical and Electronics Engineers 802.16 Standard
IIR	Infinite Impulse Response
IMT-2000	International Mobile Telecommunication for the Year 2000
IMTS	Improved Mobile Telephone Service
IoT	Internet of Things
IP	Internet Protocol
IS	International Standard
IT	Information Technology
ITU	International Telecommunication Union
JDC	Japanese Digital Cellular
LBS	Location-Based Services
LCS	Location Collection Service
LM	Levenberg-Marquardt
LTE	Long Term Evolution
MATLAB	Matrix Laboratory
MCM	Markov Chain Model
MDC	Mobile Data Challenge
MLP	Multi-Layer Perceptron
MIMO	Multiple Input Multiple Output
MIT	Massachusetts Institute of Technology
MMS	Muti-Media Messaging Service
MN	Mobile Node
MSC	Mobile Switching Centers
MSW	Minimum Squared Weight
MTS	Mobile Telephone System

MU	Mobile User
NARXNN	Non-Linear Autoregressive with External Input Neural Network
NoSQL	Not only SQL
NSS	Network Switching Subsystem
OpenLS	Open Location Services
P	Parallel
PCS	Personal Communications Service
PDC	Personal Digital Cellular
PS	Packet Switching
PSTN	Packet Switched Telephone Networks
PTT	Push to Talk
QoS	Quality of Service
RNC	Radio Network Controller
RNS	Radio Network Subsystems
SGSN	Serving GPRS Support Node
SMS	Short Message Service
SOM	Self-Organizing Map
SP	Series-Parallel
SQL	Structured Query Language
SRNC	Serving Radio Network Controller
TDMA	Time Division Multiple Access
TM	Transition Matrix
UI	User Interface
UMTS	Universal Mobile Telecommunications System
US	United States
USA	United States of America
USD	United States Dollar
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register
WAP	Wireless Access Personal
W-CDMA	Wideband Code Division Multiple access
WEKA	Waikato Environment for Knowledge Analysis
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WWWW	Wireless World Wide Web

Chapter 1

Introduction

This Chapter discusses about basic definition of Big Data for prediction of mobile users movement, problem statement, objective, related works, methodology, scope, and contributions of this research.

1.1 Background

In today's world data is everywhere. The idea of storing data into a special place called data warehouse in order to be able to analyze it and to obtain the most valuable information has already been thought and realized by many companies [1]. They copied the right data into their warehouses and selected the right technique to analyze and display the data [2]. Every piece of the data is organized and categorized in data warehouses [1]. Big Data can be described by ten main characteristics, denoted as 10Vs: Volume, Velocity, Variety, Variability, Veracity, Validity, Vulnerability, Volatility, Visualization, and Value [2]. See Figure 1.1.

The Volume means how much data is coming to Information Technology (IT) systems [2]. Velocity is the speed, at which data is being generated, produced, created, or refreshed [1]. The Variety is actually the reason why the inputs of data are so massive. In general, data can be structured, unstructured or semi-structured [1]. Variability is understood as multiple disparate data types and sources that cause a multitude of data dimensions [1]. Variability also refers to the inconsistent speed at which Big Data is loaded into the database [2]. When any or all of the above properties increase, the Veracity or in other words, confidence or trust in the data dips. This is similar to but is not the same as, Validity or Volatility [2]. Quite akin to Veracity, Validity refers to how accurate and correct the data is for its intended use [2]. Vulnerability refers to the trustworthiness of the data [2].

The Volatility is required to establish rules for data currency and availability as well as ensure rapid retrieval of information whenever it is required [1]. Yet another characteristic of Big Data is how challenging it is to Visualize. Due to limitations of in-memory technology and poor scalability, functionality, and response time, current Big Data visualization tools face technical challenges [2]. Last, and positively the most important of all, is Value. The other characteristics of Big Data are meaningless if one does not derive business Value from the data [2]. So, Big Data can be used for Location-Based Services (LBS) that are beneficiary for the users day-to-day activities.

The extensive availability of mobile communication devices together with the expansion of LBS and applications has led to an increased research attention towards well-organized methods for prediction of movements where a mobile user will travel [3]. These methods aim to improve both end-user applications, such as route planning, carpooling, meeting planners or location-based advertisements, and also be combined to solve network management issues [3, 4] such as resource allocation, traffic planning, Quality of Service (QoS) improvement, and availability.

Figure 1.1: Big Data Summary [2]

Movement prediction can be used as an efficient proactive network management measure. For example, in order to ensure a smooth handover between neighboring cells during an audio call, a common proactive measure is to allocate resources in all neighboring cells. The use of mobility prediction offers more precise resource allocation and reduces the level of network signaling by allocating resources only inside the most probable future cell. In order to achieve movement prediction we must assume that the mobile device user follows specific patterns that can be modeled. Movement prediction is also influenced by variations in the schedule of a person during a week, caused by different habits on working days and in weekends.

1.2 Statement of the Problem

Due to its importance in many industrial applications, Big Data is one of the major research areas used for prediction purposes. Many researches have been carried out on movement prediction using Neural Networks. Unfortunately, most of these researches have their flaws and they were not able to accurately predict the users movement.

One of the main challenges in movement prediction is the users mobility is random and this randomness results in inaccurate prediction of the users movement. The randomness in users mobility can be alleviated if the schedule of the users during the working day and weekends are known. Predicting the mobile users movement has many challenges due to the unrestricted mobility of the users. Researches in movement prediction [9][10][11][21], tried to deal with the challenges in different approaches. These approaches have problems and one of the problem is the related significance of the users location is not known. The other problem is telecommunication industry have huge amount of data and this data is not easily available which can be used to extract valuable information to increase revenue income and reduce overall costs. So, to solve the problems the related significance of the users location should be known and the right data should be gathered to accurately predict mobile users movement.

This work endeavors to answer the following questions:

RQ1. Is Neural Network the best method used for predicting mobile users movement?

RQ2. Among the three algorithms which type of training algorithm outperforms?

RQ3. Is it possible to improve telecommunication industry QoS, if it is likely to predict mobile users movement?

1.3 Motivation

Imagine a world without data storage; a place where every detail about a person or organization, every transaction performed, or every aspect which can be documented is lost directly after use

[3]. Organizations would thus lose the ability to extract valuable information and knowledge, perform detailed analyses, as well as provide new opportunities and advantages [3]. Anything ranging from customer names and addresses to products available, to purchases made, to employees hired, etc. So, data has become essential for day-to-day continuity [1]. Data is the building block upon which any organization thrives [1].

Now think of the extent of details and the surge of data and information provided nowadays through the advancements in technologies and the Internet. With the increase in storage capabilities and methods of data collection, huge amounts of data have become available. Every second, more and more data is being created and needs to be stored and analyzed in order to extract value. Furthermore, data has become cheaper to store, so organizations need to get as much value as possible from the huge amounts of stored data.

Imagine also the situation when you hop in your car to head to the gym after work. You don't turn on your phone's navigation because you know the way, but there is an accident on your route and a lot of traffic. If your phone knows where you are going without you telling it, it could warn you before you start driving and help you reroute. Of course, all this movement prediction sounds a bit creepy, but if it is all happening on a device, and never storing your location history in the cloud, maybe people will feel a little more secure. This work aim is to predict users location using their current location and the current time, learning from the users location history.

So, from Big Data found in telecommunication industry we can predict users movement in mobile networks to allocate resources efficiently, traffic planning, QoS improvement and availability [4].

1.4 Objective

1.4.1 General Objective

- To process and analyze Big Data for prediction of the users movement in mobile networks.

1.4.2 Specific Objectives

- To use Neural Network method to predict mobile users movement;
- To use the Gradient Descent with Adaptive Learning algorithm, the Bayesian Regularization (BR) algorithm, and the Modified Levenberg-Marquardt (LM) algorithm, for predict-

ing the future location of the user and show that the Modified LM algorithm outperforms the other algorithms;

- To obtain prediction results for significant improvement of telecommunication industry in terms of marketing activities, customer QoS and infrastructure planning.

1.5 Scope

The aim of this research work is to predict the users movement in mobile networks. The scope of our study is limited to computer simulations using OpenCelliD measured data. The simulation results of the Gradient Descent with Adaptive Learning algorithm, BR algorithm, and Modified LM algorithm are evaluated based on their performance.

1.6 Contributions

The main contributions of this work consists of:

- Well-arranged definition of LBS in cellular communications network, Big Data concept for telecommunications and provision of a deeper insight to Big Data use cases and customer's behavior in the network and QoS.

- Analysis of three types of algorithms for prediction of users movement exploiting processing of Big Data.

- Prediction of mobile users movement using OpenCelliD data set and Neural Network for prediction of future movement based on the samples of position of the users in the past.

- Optimization of created Neural Network by adaptation of the Neural Network parameters to increase prediction accuracy.

1.7 Literature Review

Former work on movement prediction use approaches such as statistics, data mining, handover analysis [5], Markov models [6-8], decision trees, Artificial Neural Networks (ANNs) [9]. Most related work on future movement prediction does not try to find the related significance of the location which could be used to enhance the results of the prediction methods. Many of these works rely on extracting spatial trajectories and do not take into consideration the user schedule and the calendar data regarding the desired day for prediction.

The work proposed in [10] is a machine learning prediction system aimed at predicting the next cellID where mobile devices will handover to. Their system uses location data gathered from the cellular network consisting of Channel State Information (CSI) and handover history [10]. Their proposed method treats location prediction as a classification problem solved by applying Support Vector Machines (SVM) [21]. Another proposed method aimed at predicting only the next location is presented in [11], which uses sequence mining by utilizing multiple support thresholds for different levels of pattern generation.

Xiaofeng et al. [12] proposes a location prediction algorithm that make use of directional antennas. Their work is mainly oriented at predicting and tracking location of moving vehicles. Parija et al. [9] employ the use of Neural Networks as a solution to the location management problem in cellular networks. Their work proposes a prediction based location management scheme using the Multi-Layer Perceptron (MLP) as basis.

In [22], Gao et al. uses both spatial and temporal data to predict users locations. In order to use temporal information in prediction phase, they assume that a person's daily visit frequencies follow Gaussian distribution. They also break down the temporal information as hour and day. They managed to predict user locations with an accuracy rate of 50% with their proposed method which is named as Hierarchical Pitman-Yor (HPY)-Prior Hour Day Model [22].

In [23], Rajagopal et al. proposes a location prediction algorithm which first calculates cell-to-cell transition probabilities of a mobile user using the previous movements of the user, then saves this information to a matrix. Based on this matrix, next location is predicted using k, which is a user defined parameter, most probable cells which are the neighbors of the current cell. This algorithm is called Mobility Prediction based on Transition Matrix (TM).

The work presented in [7] proposes a hybrid method for future location prediction using Hidden Markov Models (HMMs). Their approach clusters location histories according to their characteristics, and latter trains HMMs for each cluster. The HMMs [21] are then used for location prediction. Their method of location clustering has the same objective as our location detection method, though different techniques are employed. For location prediction to be precise, the identification of the boundaries and the significance of the user locations are needed.

The proposed method in this work for movement prediction combines spatial and temporal analysis in order to extract the location of the user. The extracted data is then used to adapt the Neural Network according to the desired day for location prediction. This work describes the steps involved in designing a network that can obtain location prediction using mobile data collected using satellite Global Positioning System (GPS) and applications.

1.8 Methodology

In order to conduct this research the methodology shown in Figure 1.2 is used.

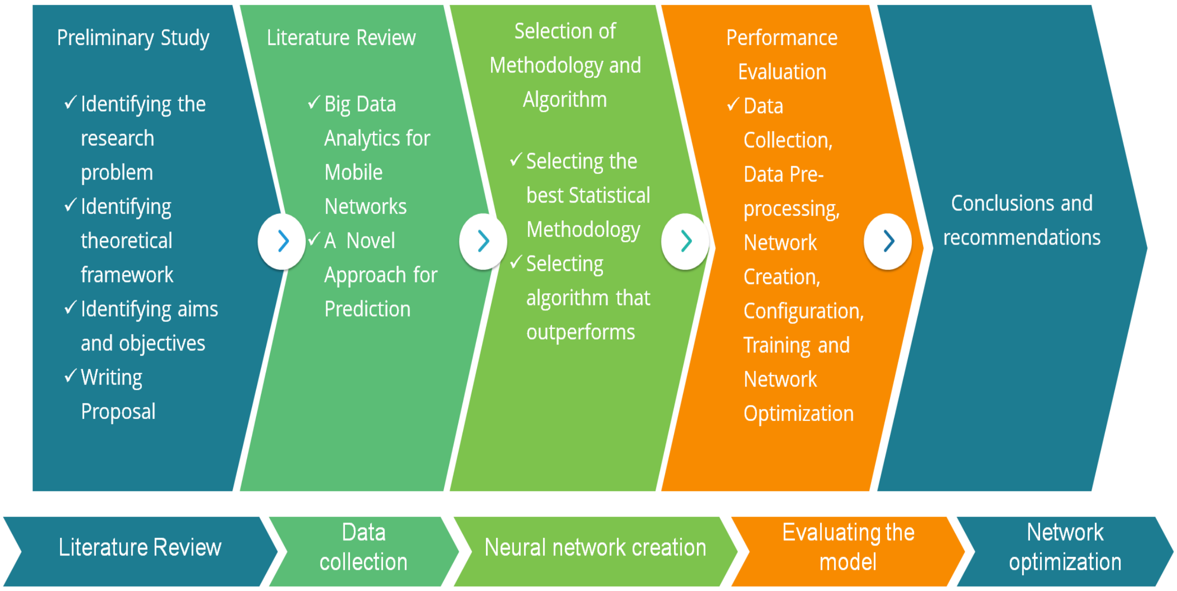

Figure 1.2: Research Methodology

Literature Review: Literature survey of theoretical bases and related works on users movement prediction.

Data Collection and Pre-Processing: The data needed for this research work are collected and pre-processed.

Neural Network Creation: The Non-Linear Autoregressive with External Input Neural Network (NARXNN) desired for this work is created.

Evaluating the Neural Network Model: Finally the network that is created is evaluated based on the performance result obtained.

1.9 Thesis Organization

Following this first Chapter, the second Chapter presents an overview of the evolution of cellular networks and LBS in cellular communications network. The third Chapter presents an overview of Big Data in telecommunication industry and one of the deep learning technique called Neural Network, which is used as a method for prediction of users movement in mobile networks. The fourth Chapter presents the experimentation procedure starting from data collection to the results obtained for prediction of users movement in mobile networks. The fifth Chapter concludes this work and it provides a recommendation for the future.

Chapter 2

Location-Based Services (LBS) over Cellular Communications Network

This Chapter provides a brief background about First Generation (1G), Second Generation (2G), Third Generation (3G), Fourth Generation (4G), Fifth Generation (5G), LBS and Mobility Prediction, giving an exposition of 3G as a Universal Mobile Telecommunications System (UMTS) architecture.

2.1 Cellular Communications Network Generations

Cellular communications network made a huge step in the way people communicate. The remarkable growth in cellular communications in the last few years has led to mobile technologies development and increase in the number of users.

The evolution of this technology is reaching 5G [71]. The previous generations were developed through research with the 1G, 2G, and 2.5 Generation (2.5G) developed simultaneously along with the 3G [59]. The 1G was developed to provide the basic voice communication with mobile ability, while the 2G and 2.5G introduced the concepts of capacity and coverage [59]. The 3G made the mobile broadband concept a reality by providing higher speeds for data transmission [59]. The 4G is developed to accommodate the QoS and rate requirements set by forthcoming applications like wireless broadband access, Multi-media Messaging Service (MMS), video chat, mobile TV, High Definition Television (HDTV) content, Digital Video Broadcasting (DVB), minimal services like voice and data, and other services that utilize bandwidth [71]. This evolutionary time-line continues to reach 5G which expands the Mobile User (MU) services and provides a higher data rate to support both low and high speed mobile applications [71].

2.1.1 First Generation (1G) Network

These phones were the first mobile phones to be used, which was introduced in 1982 and completed in early 1990 [72]. It was used for voice services and was based on a technology called Advanced Mobile Phone System (AMPS). The AMPS was frequency modulated and used Frequency Division Multiple Access (FDMA) technique with a channel capacity of 30KHz and a frequency band of 824 - 894MHz [72]. Its speed is 2.4Kbps, allows voice calls in one country, use analog signal, poor voice quality, poor battery life, large phone size, and offered very low level of spectrum efficiency [72].

It also introduces mobile technologies such as Mobile Telephone System (MTS), Advanced Mobile Telephone System (AMTS), Improved Mobile Telephone Service (IMTS), and Push to Talk (PTT) [71]. Furthermore, it has low capacity, unreliable handover, poor voice links, and no security at all since voice calls were played back in radio towers, making these calls susceptible to unwanted eavesdropping by third parties [72].

2.1.2 Second Generation (2G) Network

At the end of the 1980s, the MUs increased in number which led to a serious need for increasing the capacity of the network [60]. This was achieved by introducing a digital system that could support services such as low data rate and traditional speech. These systems are considered as the 2G cellular systems [60]. The services supported by the 2G cellular systems use digital multiple access technologies, such as Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) [60]. These digital multiple access technologies provide higher spectrum efficiency, better data services, and more advanced roaming [60]. Standards for the 2G were deployed and put into effect in European Telecommunication Service Institute (ETSI) called Global System for Mobile Communications (GSM) which enabled reliable services throughout Europe to support international roaming [60]. The support of multiple users was introduced in TDMA technology. In the years since that time, GSM technology has been continuously improved to offer better services to satisfy the needs of the network subscribers. Based on the GSM system more technologies and new services have been developed, which leads to creating new systems known as 2.5G systems [60].

Meanwhile, the United States of America (USA) started its line of development for the 2G digital cellular systems [60]. In 1991 the first digital system called the International Standard-54 (IS-54) (North America TDMA Digital Cellular) was introduced [60]. Years later a new version supporting additional services (the IS-136 Standard) was introduced. In 1993, IS-95 (CDMA One Standard) was deployed [60]. The United States (US) Federal Communications Commission (FCC) allowed the 1900MHz band spectrum to become operational for Personal Communications Service (PCS), allowing GSM1900 to enter the USA cellular communications network market [60]. In 1990, a smaller scale of the 2G was defined by the Japanese Personal Digital Cellular (PDC) system, which was known as Japanese Digital Cellular (JDC) [60].

The main elements of the GSM architecture reside in the Base Station Subsystem (BSS) [59]. The BSS is built up using Base Transceiver Stations (BTSs) and Base Station Controllers (BSCs). The Network Switching Subsystem (NSS) consists the Mobile Switching Centers (MSC), the Visitor Location Register (VLR), the Home Location Register (HLR), the Authentication Center (AC), and the Equipment Identity Register (EIR) which were developed during that time [59].

2.1.3 Third Generation (3G) Network

The evolution in technologies created the need to develop a more enhanced cellular communications network than the one in the 2G or 2.5G to have faster data rate, higher data capacity and better QoS [59]. International Mobile Telecommunication for the year 2000 (IMT-2000) as specified by the International Telecommunication Union (ITU) involved the newly developed standards and specifications of the communications network [61]. The 3G system solved the differences in the standards such as in GSM and CDMA by introducing new solutions such as in roaming services [61]. These standards were grouped into families to solve several issues and expand the set of services such as voice-video calls, and broadband wireless data transfer within the cellular environment [59].

The Third Generation Partnership Project (3GPP) worked on a wideband system based on the Wideband CDMA (W-CDMA) technology, referred to as UMTS, that uses the core of GSM technology [61]. This solution improved the data rate transmission from the mobile device over the network by different frequencies than 2G with a downlink rate up to 14.4 Mbps and uplink rate up to 5.8Mbps as in High Speed Packet Data Access (HSPDA) [59]. The 3G QoS standards assure subscribers to have the highest data rate provided by the operator [61]. The architecture of the 3G system is described in the UMTS architecture Section.

2.1.4 Fourth Generation (4G) Network

The rapid increase in the users demand to have a higher data rate than the previous technology encourages the wheel of development and research towards new technology to keep turning and deliver new user experiences. The 4G introduced a new platform that combined all kinds of known mobile technologies such as GSM, CDMA, UMTS, Wireless Fidelity (Wi-Fi), and even Bluetooth to deliver what is called the IMT-Advanced multi-service [62, 63]. Multi-service means integrating all the cellular services that the user has or would expect to have with the highest service quality [62]. The IMT-Advanced services provide fast interactive cellular services with enhanced data rate access [64, 65], roaming capability as well as broadband multi-media. Long Term Evolution (LTE-Advanced) is one of the IMT-Advanced items already provided by 3GPP [65]. Another item is Worldwide Interoperability for Microwave Access Multiple-input Multiple-output communications (WiMAX MIMO) which is provided by the Institute of Electrical and Electronics Engineers 802.16 Standard (IEEE 802.16).

2.1.5 Fifth Generation (5G) Network

5G was started from late 2010s [71]. Facilities that might be seen with 5G technology includes far better levels of connectivity and coverage [71]. The main focus of 5G will be on Wireless World Wide Web (WWWW). It is a complete wireless communication with no limitations [71]. The main features of 5G are: it is highly supportable to WWWW; high speed, high capacity; provides large broadcasting of data in Gbps; multi-media newspapers, watch TV programs with clarity (HD Clarity); Faster data transmission than that of the previous generation; large phone memory, dialing speed, clarity in audio/video; support interactive multi-media, voice, streaming video, Internet and; more effective and attractive [71].

2.1.6 Universal Mobile Telecommunications System (UMTS) Architecture

UMTS is 3G system promoted by ETSI and provides a vital link between today's multiple GSM systems and the ultimate single worldwide system for all cellular communications network. It is also referred to as W-CDMA and it is one of the most significant advanced evolution of communications network from 3G networks [67]. It addresses the growing demands of mobile and Internet applications in the overcrowded cellular telecommunications network. It increases the network speeds and establishes a global roaming standard.

A simplified architecture of a UMTS system is shown in Figure 2.1. UMTS is divided into three main components: MU, Universal Terrestrial Radio Access Network (UTRAN) and Core

Network (CN), with the corresponding interfaces among them [67]. The CN, which is responsible for connecting UMTS to external networks, provides functionalities of switching calls for voice communications and Packet Switched (PS) services for data transmissions [68].

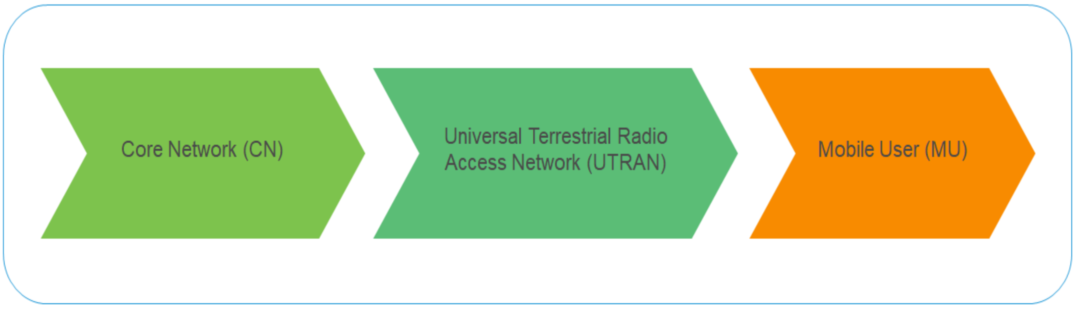

Figure 2.1: A Simplified Architecture of UMTS System [68]

The NodeB which is also called Base Station (BS) and the Radio Network Controller (RNC) are collectively known as the UTRAN [68]. From the UTRAN to the CN, the RNC is responsible for handling radio resources of UTRAN [68]. NodeB is the lowest element of the UTRAN, which connects to the MU directly [68]. The RNC decides on where the traffic should be transmitted. Packet traffic is sent to a new component, first to the Serving General Packet Radio Service (GPRS) Support Node (SGSN) and then to the Gateway GPRS Support Node (GGSN) [69]. The function of the GGSN is very similar to the normal Internet Protocol (IP) gateway, which transfers the receiving packets to the appropriate Internet [69].

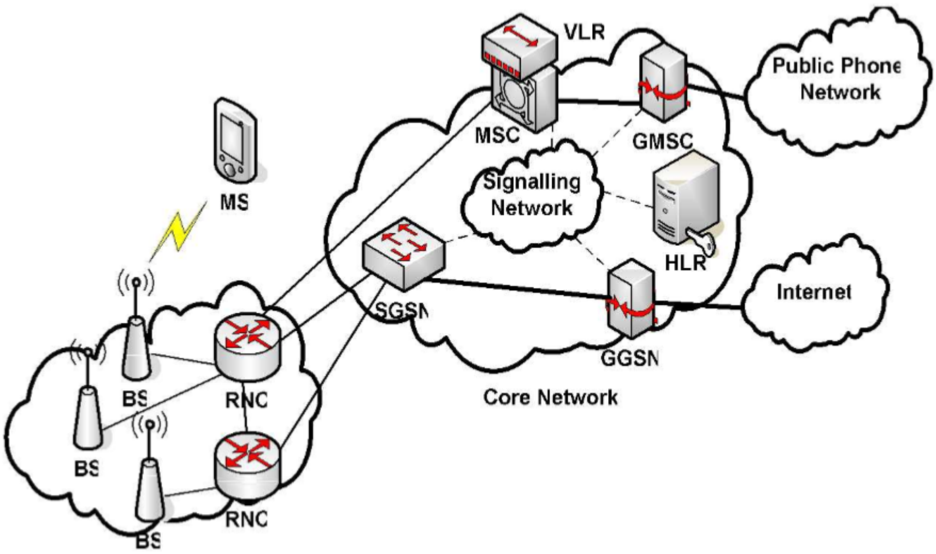

Figure 2.2: The Architecture and Connections between UMTS Components [69]

If there is a voice call from a subscriber, the RNC will transmit the traffic to the MSC. If the subscriber is authenticated before, the MSC switches the phone call to another MSC. The call is switched to the Gateway MSC (GMSC) if the called end is in the public fixed phone network. The MU which is the terminal of the UMTS [68], interfaces with the radio interface of UTRAN and user applications. A diagrammatic illustration of the UMTS architecture is shown in Figure 2.2 [69].

Figure 2.3 shows the infrastructure of the UTRAN. The components that composes the UTRAN are the Radio Network Subsystems (RNS). A UTRAN contains one or more RNS, each of which is connected to the CN respectively. RNS can be divided further into two entities: RNC and BS called NodeB in standards. One RNS contains only one RNC and one or more NodeB's [69].

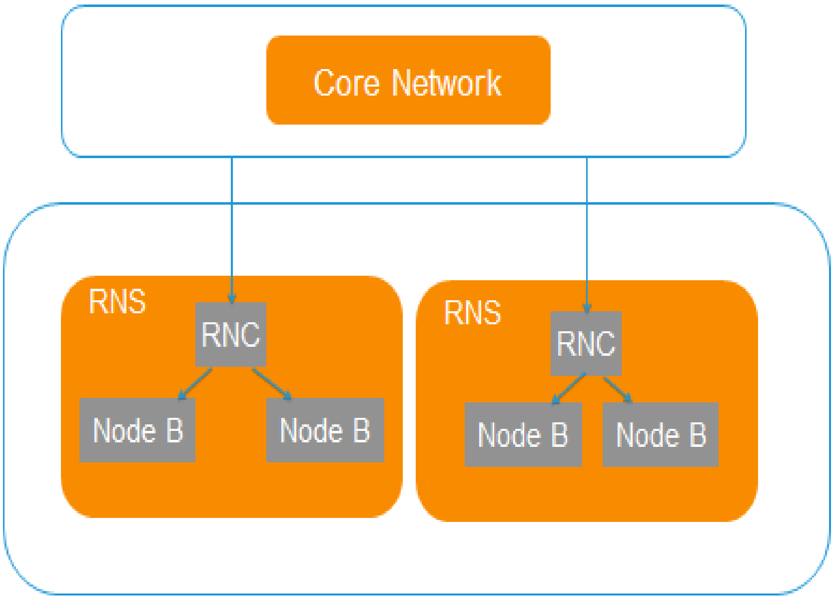

Figure 2.3: UTRAN Architecture [69]

RNC is responsible for resource allocation and transmission/reception in a group of cells [69]. For each connection between the MU and the UTRAN, there exists a RNC, namely Serving RNC (SRNC) to control the establishment and the release of specific radio resources to the connection [69]. If the connection state changes because of the move of the MU, the connection may be handed over to a different RNC, namely Drifting RNC (DRNC) [69]. The SRNC decides based on the parameters given by the MU and the UTRAN, whether a handover is necessary or not and performs the initial handover if it is necessary. The SRNC is responsible for the Macro-diversity, the Macro-diversity determines the threshold for the MU to connect more than one cell and the serving cell [69].

2.1.7 Comparison of Generations of Mobile Technologies

Table 2.1 compares the five generations of mobile technologies starting from deployment year, followed by data bandwidth, technology used, core network, type of multiplexing used, switching mechanism applied, primary service, key differentiator and finally weakness of the technologies.

Table 2.1: Comparison of Mobile Generation: 1G to 5G [71]

Technology	**1G**	**2G**	**3G**	**4G**	**5G**
Deployment	1970-80	1990-2004	2004-10	Now	Soon (probably by 2020)
Data Bandwidth	2Kbps	64Kbps	2Mbps	1Gbps	Higher than 1Gbps
Technology	Analog	Digital	CDMA 2000, UMTS, EDGE	WiMAX, Wi-Fi, LTE	WWWW
Core Network	PSTN	PSTN	Packet	Internet	Internet
Multiplexing	FDMA	TDMA/CDMA	CDMA	OFDMA	OFDMA
Switching	Circuit	Circuit, Packet	Packet	All Packet	All Packet
Primary Service	Analog Phone Calls	Digital Phone Calls and Messaging	Phone Calls, Messaging, Data	All-IP Service (including Voice Messages)	High speed, High capacity and provide large broadcasting of data in Gbps
Key Differentiator	Mobility	Secure, Mass Adoption	Better Internet Experience	Faster Broadband Internet, Lower Latency	Better coverage and no dropped calls, much lower latency, Better performance
Weakness	Poor spectral efficiency, major security issue	Limited data rates, difficult to support demand for Internet and E-mail	Real performance fail to match type, failure of WAP for internet access	Battery use is more, Requires complicated and expensive hardware	?

2.2 Location-Based Services (LBS)

LBS are services offered through a mobile phone and take into account the device's geographical location. Many services can be considered as LBS such as navigation assistance, emergency location detection applications, disaster aid, finding friends in social networking, locating point of interest using map applications, etc [70]. In order to use LBS the correct positioning technology is required. Using a Geographical Information System (GIS) that shows the areas and through mapping the environment where a mobile operator serves is important [70]. LBS faced a lot of challenges in the past such as insufficient positioning technologies, limited standards that support it amongst operators and networks, inappropriate infrastructure such as lack of GPS enabled handsets [70]. Recently, there has been a modification on the business model by considering location as the subject of operators. LBS are now supported by most mobile phone applications. LBS applications design features are started by collecting positioning data through GIS, and these data are mapped with sufficient information about the road network and point of interest [70].

2.2.1 Components of LBS

LBS architecture is built up upon several layers, each of which contains many components. The first layer is User Interface (UI) which includes; LBS application, application server data and smart phone devices [66, 70]. The second layer is LBS middleware. The third layer holds the core LBS features which are: Location Tracking, GIS provider with data and Location Collection Services (LCS), see Figure 2.4 [70].

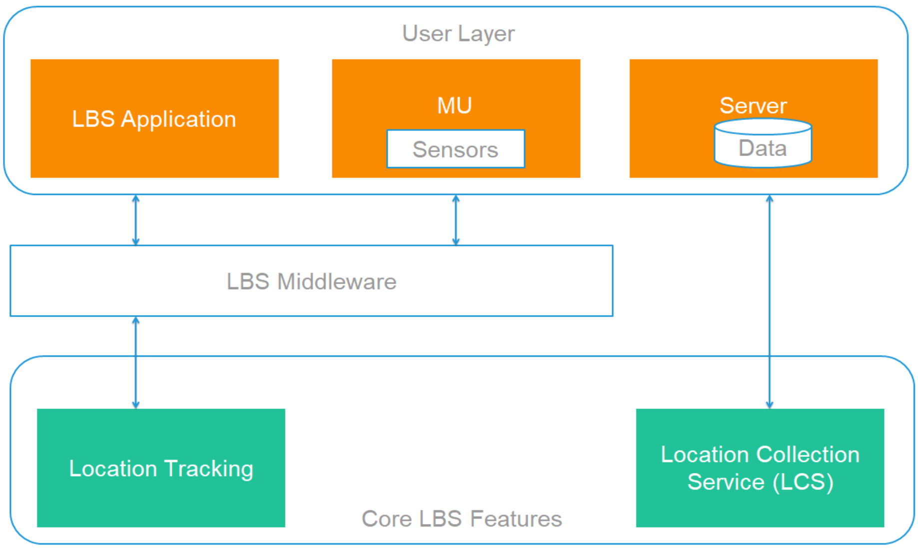

Figure 2.4: Diagram Showing Components of LBS [70]

LBS components and their functionality are explained below [70]:

User Side (US): LBS application is installed on a smartphone component and consists of a number of sensors connected to a server component that stores the application's data.

LBS Middleware: Provides a medium between LBS application and core LBS features layer. One example of such middleware is the Open Location Services (OpenLS) which specifies interfaces that enable companies in the LBS value chain to provide their pieces of applications.

Core LBS Features: Location Tracking to store traces of the MU locations. It works by keeping records of MUs' locations, current and previous. It also detects the MUs location within a defined location in addition to generating MU's movement model. The GIS provider is mostly responsible to provide geo-spatial functionality that includes information such as maps, and visualization of maps in addition to directory services. This can be found in "Google Maps". LCS is used to detect latitude and longitude of specific users by collecting their locations.

2.2.2 Benefits of LBS

LBS provide a number of benefits for MUs, these benefits can be reached through the services served by the cooperation chain to enable the MUs to get every relevant material on the Internet, that aids him/her in taking decisions by filtering the important information to fit the position that he/she is located at, for example, choosing the best hotel or restaurant in the area.

Another benefit of LBS is to provide MUs with instant information that is not usually known by MU to speed up their activities and decisions. This information can be served when reaching certain areas - road closures for example.

Furthermore, local information is available for all MUs who are within certain areas by sharing and tagging locations. This makes the set of MU movements with an associated tagged information a very helpful source of information for service providers and operators to build models that would enhance services.

2.3 Movement (Mobility) Prediction

The reason why we move are many. There are activities performed to satisfy physiological needs like eating and sleeping, institutional demands like work and school, personal preferences like childcare, shopping, and leisure. The actual and desired activity of an individual are also dependent on the composition, roles, and tasks of the household to which he/she belongs and on the significance of the home as a place of living, a place to which one returns, and the prime location of people's time spending. It is not only the mandatory tasks in life that makes people want to move but also movement itself can constitute an activity. A sense of speed, motion, control, and enjoyment may motivate people to undertake "excess" movement.

With the current growth rates of mobile and high speed wireless communication networks, mobile users expect to have a high QoS anywhere and anytime for accessing information [73]. Telecommunication service providers are expected to provide high-speed data, superior quality voice and LBS [73]. It has also been evident that mobile communications technology has developed very rapidly over the past few decades [74]. To achieve a high data rate, the concept of mobility is considered as a very important feature of wireless networks. Several researches have been carried out on mobility [74], and it is seen that in all of the technologies that have been proposed, the Mobile Node (MN) has a point of attachment with the BS which serves its mobility needs. When a MN is active in a network, there is always a continuous exchange of radio signals between the MN and BS to which the antennas are attached [74].

Mobility prediction has remained an area of research seeking to come up with an optimal solution to track and locate mobiles in cellular networks [74]. Mobility has become the order of the day with mobile users changing their positions at varying velocities and directions. MUs, thus, move from the proximity of one cell to another. Locating users as they move from one cell to another in a cellular computing environment is the key to provide continuous services with unrestricted mobility [75].

Therefore, the data management in this environment involves challenges in the need to process information during the move, to cope with resource limitations and to deal with heterogeneity [75]. One of the applications of cellular data management is LBS which have been identified as one of the most promising areas of research and development [75].

Two of the techniques used for mobility prediction in cellular communications network are discussed below.

2.3.1 Cell-Based Technique

In the cell-based techniques, a service area is partitioned into several cells [76]. The cell covering the MU pages his/her device to establish a radio link to track the changes in the location of MUs. The cells broadcast their identities and the MU periodically listens to the broadcast cell identity and compares it with the cell identity stored in its buffer [76]. If the comparison indicates that the location has been changed then the MU sends a location update message to the network [76]. The knowledge of the history (habits) of the users and their current location is useful in determining their probable future location.

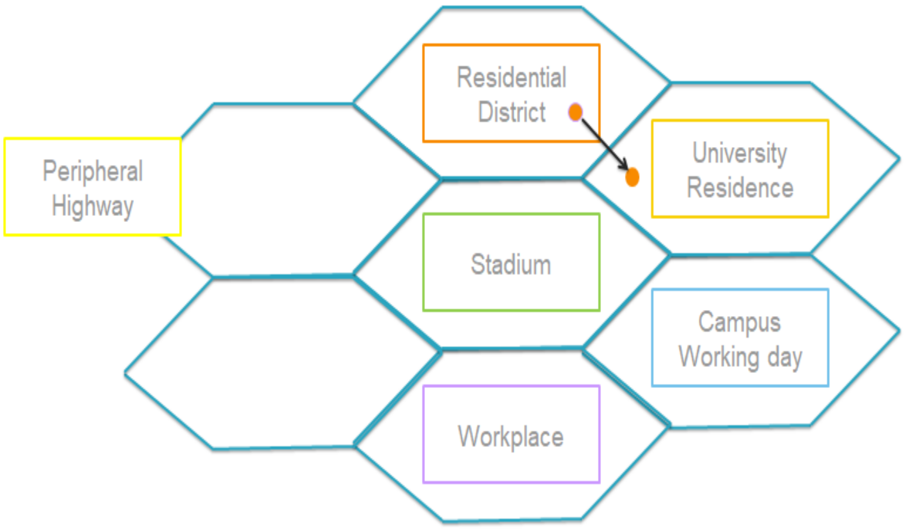

Figure 2.5: Cell-Based Technique in Cellular Networks

Prediction techniques that are based on a cell technique can be enhanced by heuristic methods and Neural Network [77, 78]. In [77] the cell is divided into two areas, edge and non-edge. The edge areas have neighboring cells, while the remaining areas are considered as non-edge areas. When the MU is in a cell's edge area, the information is passed to the Neural Network which predicts the next cell that will be visited [77]. Another technique captures some of the MU activity and paths. These paths are progressively recorded, giving a history record which is used as an input to the Neural Network to predict the next cell that will be visited [78].

Let us assume that we have a set of cells, each one respectively covering residential district, stadium, university residence, university campus, workplace, and segments of highway as shown in Figure 2.5. If the users with history (that is wherein we already know the practices of displacements) are located in residential district on working day, and if it is known that they usually take the highway to go to work, then it is very probable that the future cells are those that cover the highway and the workplace. In the same manner, if the users are in the workplace, we can predict that the future cells will be those covering the highway in direction of their residential district. The same assumption applies to students with history who spend the majority of their time between the university residence and the campus. If a student is in the university residence, there is a strong chance that his/her next destination is the campus. Except on Sundays, when he/she may prefer to go to the stadium, for example, instead of the campus.

This type of mobility prediction technique in cellular communications network is used in this work.

2.3.2 Map-Based Technique

The map matching algorithm has been used for mobility prediction [79]. In [79], the map matching algorithm has been developed by its dependence on the Markov Chain Model (MCM) and GPS sensor. The distance and the direction between the points which are recorded by GPS are used. In map matching, using fuzzy logic and data gathered from GPS is considered as one of the techniques used to navigate the users in sidewalk areas [79]. The GPS data and the map of the target area are stored in the server-side and the analysis of the data is performed by the fuzzy logic. Therefore, the incorrect direction will be eliminated and thus it will advise the users how to reach the best destination [80].

The drawback of using the map-based technique is anyone who needs to use them must have GPS sensors. However, the GPS sensors lead to extra physical cost bearing in mind that they may not be applicable for all mobile devices. Moreover, GPS suffers from inaccurate data on narrow roads, in high buildings, so higher-end GPS is used to improve the signal instead of lower-end GPS [80].

Chapter 3

Big Data in Telecommunications Industry and Neural Networks

This Chapter provides a summary of Big Data in telecommunication industry and Neural Network which is one of the deep learning techniques used for mobile users movement prediction.

3.1 Big Data

The term Big Data is used for many years [13]. Big Data is a data set that is too large or complex to analyze by using traditional data analytics tools [13]. At the beginning of that era, companies took notice of the increasing amount of data. They started thinking about what they should have done with that data, that Volume. The first applications were developed around 1980s and the first attempts to process such Volume were carried out at the same time [13]. About the year 2000 Google came up with a new idea on how to process, store and manage a lot of information [13]. In the 2000's Google was the first who started to generate a huge amount of data from the Internet and the first company which started to use the term Big Data [13]. The latest trend is shown in Figure 3.1, which shows an exponential increase in the amount of data over time [14].

Google created a new concept of working with big sets of data, denoted as BigTable [13, 14]. This concept is divided into three parts: Google's data storage system, Google file system and MapReduce [14]. However, BigTable is the first framework that was created in connection with Big Data, its concept survives till today. Today's Big Data framework includes a wide range of software and hardware [14].

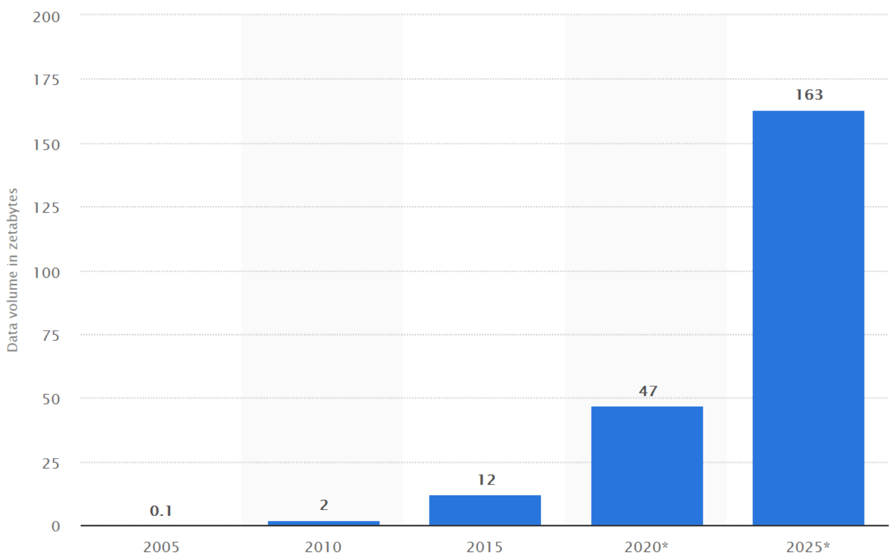

Figure 3.1: A Time-Line of Big Data Year's verses Data Volume in Zettabytes [14]

3.2 Big Data Use Cases for Telecommunications

Nowadays, almost all people have their own mobile phone. These devices provide Communication Service Providers (CSPs) with a lot of information about their customers [19]. This information can be turned into valuable insight by using appropriate Big Data tools. Today, it is not only about mobile devices but also smartphones, tablets, sensors, and TVs come to play [15]. All these devices produce so much information that it is almost impossible to process and analyze it all together at one time. There are a lot of possible use cases for the telecommunications industry sector and it can be divided according to countless amounts of characteristics [15]. One way is to split up use cases into 4 parts: marketing, customer experience, exogenous influencer's and technical use cases [15, 19]. All the cases have one thing in common – they describe basic facts about selected use cases and all are focused on generating profit and reduce overall costs connected with customer activities.

In the following Sections, we attempt to present the most relevant and efficient Big Data use cases in the telecommunications industry.

3.2.1 Improved Network Security

Telecoms have large networks and ensuring the security of these networks is a daunting task [38]. Now, this has been made a lot easier with the ability to analyze streaming events in real-time [33]. Telecoms define the rules to identify security issues, predictive analytics, and machine learning-based solutions to analyze patterns, to identify any threats and automated escalations, and actions segregate or resolve the issue before it can cause any damage [38].

3.2.2 Better Customer Service

Telecoms are in the service business for phone, Internet, cable TV and others [15]. On one hand they have to ensure excellent customer service and on the other hand, they have to take steps to avoid customer churn. Big Data helps with both these areas by pro-actively working on issues, sometimes even before they impact the customer [39]. The customer service team is given visibility to 360° customer data for understanding the customer history and the current state [33]. This results in better decision making when interacting with customers. Big Data solutions are used to aggregate and analyze customer data from multiple sources and results in a better customer experience thus reducing churn [33]. It also lowers the operational costs as calls are handled faster.

3.2.3 Reduced Fraud Losses

Fraud is a big problem in the Telecom world. A recent survey estimates global Telecom fraud losses at $40.1 Billion United States Dollar (USD) or approximately 1.88% of revenue [33]. This number is based on only the known fraud transactions, adding the unknown fraud i.e. the fraud that could not be identified, could easily double the fraud losses. The different types of fraud that can be minimized by using Big Data solutions are [33]:

- **Toll Number Fraud:** Real-time call analysis reduces this type of fraud as a Telecom can lose thousands of dollars very quickly by someone calling toll numbers that cost $5/ minute (or higher).

- **Bypass Fraud:** This mainly involves unauthorized traffic in Telecom's network. This is prevented by using Big Data to review the source of the transactions, the destination number and the cost of the call, in real-time.

- **Credit Card Fraud:** This is the usual charge-back fraud and card not present fraud, that impacts other verticals and can be minimized by correlating real-time transactions with historical activity.

3.2.4 Predictive Maintenance

The large network of Telecom has thousands of devices and to ensure no impact to customers, these devices have to be up and running all the time [33]. Telecoms use sensors to send data of the device's current state and utilize predictive analytics solutions to resolve the issue before it causes any service issues [41].

3.2.5 Targeted Campaigns

Big Data solutions help understand the customer better by reviewing how they use the services offered by Telecom. As an example, a customer who calls a specific country more often or watches on-demand movies can be targeted with a campaign for these two areas [33]. This targeting can also be done in real-time. For example, after the customer has finished the call, a real-time campaign could say – sign up for the monthly plan and the customer could have saved 40% on the last call [33]. This results in better conversion and more revenue for Telecom. This also increases customer satisfaction and reduces churn.

3.2.6 Real-Time Network Analysis

Telecoms need to continuously monitor their network to avoid any issues and Big Data solutions have made this possible [40]. They can now get alerts if a part of the network is experiencing more traffic than usual and could soon impact the customer experience [40]. They can also determine the right time to upgrade the network to add more capacity as they on-board more customers. This results in better use of their investment dollars by focusing on areas that will deliver the best return [40].

3.2.7 Intelligent Transportation System (ITS)

Operators with their Big Data analytics can help the government or individual subscriber to improve their transportation service [41]. Operators can visualize the traffic situation in a specific area and let the subscribers know about it when they require [41]. They can also provide some pop-up services which will send the subscribers' traffic information [41]. For example, if there is road blocking on their way. Operators can also offer services such as vehicle tracking with GPS services for security purposes, route mapping, suggesting the nearest gas station when gas is running low and tracking of driven kilometers for taxation purposes [41]. These types of use cases will typically require data types, such as location data from the handset, and mapping data from the data bases.

3.2.8 Product Innovation

All the real-time data from multiple sources can be used to improve the products offered by Telecom [41]. They can also analyze customer usage to come up with new product bundles that will serve the customer's needs and also save money. A great example of one such innovative feature offered by a Telecom, after analyzing customer data, is the ability to use their wi-fi service from anywhere [41]. The customer only needs to log in and they can use the wi-fi whether they are at home, in a restaurant, a coffee shop or at the airport [41].

3.2.9 Call Detail Record (CDR) Analysis

CDR has rich information about each call that can further improve the customer experience but this analysis used to take much longer before Telecoms started using Big Data solutions [43]. Now Telecoms can review metrics like call latency, packet loss, and call quality in real-time and take appropriate action [43]. They can also collaborate with third-parties to identify any suspect conversations or unusual call activity [43]. And the data can be combined with historical data to define usage patterns and understand customer behavior better.

3.2.10 Optimized Pricing

Big Data solutions review several metrics in real-time to come up with the price for a product offering [42]. This price is tested with different segments of customers in different regions to come up with the most optimized price [42]. This is a win-win for the customers and the Telecom as customers get the price they are happy with and the Telecom gets a steady revenue stream with low customer churn.

3.2.11 Contextual Location-Based Promotions

Telecoms can detect customer location in real-time based on detecting the location of their mobile device [41]. This information is then used to send contextual promotions by partnering with different merchants. An example would be 20% off at a restaurant when the customer is within 100 feet of that restaurant around lunchtime [33]. These promotions have a higher conversion rate as they are contextualized and end up generating more revenue for the Telecom by getting a cut on each transaction.

All the mentioned use cases above can be combined to create a new use case. The rest of this work is focused on the mobile users movement prediction use case. Mobile users movement prediction use case is primarily an adjusted combination of location-based analysis and reaction to an event use cases [19], which are parts of marketing use cases. Because CSPs pay a lot of attention to this topic as it can bring new revenue income and extended customer services.

3.2.11.1 Location-Based Analysis

Location-based marketing is a very interesting way how to approach customers at the right place and at the right time even without personal contact [15]. The best example could be a shopping mall. There are a lot of small shops. According to the customer's actual geo-location position, CSPs can send him/her a message with sales and promotions based on positions of shops [15]. The customer is then informed about actual events in his/her near location. These real-time analyzes are appropriate mainly for marketers and business strategy planners.

3.2.11.2 Reaction to an Event

People want to get information as fast as possible and be informed about all happenings. CSPs can connect business data coming from Customer Relationships Management (CRM) and marketing applications with the technical data coming from network probes to react nimbly to customer demands, changes and technical problems like drop calls, Short Message Service (SMS) fail delivery and others [15, 19]. It is used primarily by technicians and business strategists.

3.3 Artificial Neural Network (ANN)

ANN in short Neural Network is composed of simple elements operating in parallel [34]. These elements are inspired by biological nervous systems built to perform tasks in a similar manner that the human brain works [34]. This predictive method is a key part of data mining methods examined by many scientists and engineers for prediction purposes. It brings a new view of data processing [16]. Neural Networks are not as accurate as of the two methods tree and rules, and statistical models and the learning process is even slower at the beginning [17, 19]. But after being trained they are faster and they can learn the mutual relationships among input data. The Neural Network, in general, is depicted in Figure 3.3 [17].

ANN consists of a number of very simple and highly interconnected processors which are called neurons [34]. These neurons are like the biological neurons in the brain. The neurons are connected by weighted links passing signals from one neuron to another [34].

The input comes into the neuron that is weighted (each input can be individually multiplied with a weight). The neuron then sums the weighted inputs, bias and processes the sum with a transfer function. In the end, a neuron passes the processed information via output. The benefit of neuron model simplicity can be seen in its mathematical description below [35]:

$$y_d = \varphi\left(\sum_{i=0}^{m} w_i x_i + b\right) \tag{3.1}$$

where y_d is the desired value, b is the bias, w_i is the weight value, φ is the transfer function and x_i is the input value [35].

3.4 Transfer Function

The essential and important unit in ANN structure is a scalar function for network inputs called transfer function, or activation function and the result value of the output is called the unit's activation [49]. A transfer function limits the amplitude of the output of a neuron [49]. Some of the most commonly used transfer functions are for solving non-linear problems. The transfer function denoted by $\varphi : R \to R$ defines the output of a neuron, which is bounded, monotonically increasing, continuous, differentiable, and satisfies [48]:

$$lim_{x\to+\infty}\varphi\left(x\right) = 1 \ and \ lim_{x\to-\infty}\varphi\left(x\right) = 0. \tag{3.2}$$

The sigmoid function is by far the most common transfer function used in construction of ANNs [50]. An example of the sigmoid function is the logistic function defined in the range from 0 to +1 . In [48] one of the most commonly used transfer function the log-sigmoid (logsig.) transfer function is described. This transfer function takes the input that may have any value between $+\infty$ to - ∞ and squashes the output in the range from 0 to 1 .

It is sometimes desirable to have the transfer function ranging from -1 to $+1$ allowing the transfer function of the sigmoid type to take negative values, in which the transfer function assumes an anti-symmetric form with respect to the origin, satisfying $\varphi\left(-x\right) = -\varphi\left(x\right)$ [50]. For example, the hyperbolic tangent transfer function (tansig.) has particularly simple derivative [48]:

$$\varphi\left(x\right) = tanh\left(x\right), \ \varphi^{'}\left(x\right) = 1 - \varphi^2 \tag{3.3}$$

Throughout this work, we take tansig. as a transfer function, depending on the results of [26, 27], which evidence that an anti-symmetric transfer function tansig. enables the training algorithm to learn faster, and in ANN it is related to a bipolar sigmoid having an output in the range of -1 to $+1$. As can be seen in Figure 3.2, this function is, mathematically equivalent to $tanh\left(n\right)$ [49, 50]. It differs from logsig., because it runs faster, but the results can have very small numerical differences. This function is a good tradeoff for ANN in which the speed is more important than the exact shape of the transfer function.

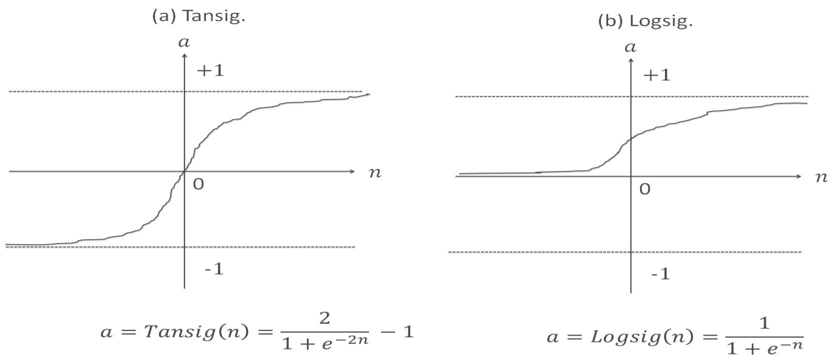

Figure 3.2: Sigmoid Functions (a) Tan-Sigmoid, and (b) Log-Sigmoid [28]

3.5 Architecture of Neural Networks

The architecture of the Neural Network defines its structure including number of hidden layers, number of hidden nodes (neurons), and number of input and output nodes [35]. Typically, the network consists of an input layer of source neurons, at least one hidden layer of computational neurons, and an output layer of computational neurons [35]. The input signals are propagated in a forward direction on a layer-by-layer basis.

The hidden layers provide the network with its ability to generalize. In theory, ANN with one hidden layer and a sufficient number of hidden neurons is capable of approximating any continuous function [34]. In practice, ANN with one and occasionally two hidden layers are widely used and performs well. In general, ANNs with one hidden layer, a non-linear transfer function and a sufficient number of hidden neurons can approximate any function with arbitrary precision [25].

In [26, 27] they determined the optimal network architecture for approximation problems. There is no magic formula for selecting the optimum number of hidden neurons. Let n be the number of input nodes. From Kolmogorov existence theorem (the theorem guarantees that a suitably consistent collection of finite-dimensional distribution will define a stochastic process) we know that a three-layer perceptron (linear classifier) network with $n\,(2n+1)$ nodes can compute any continuous function of n variables [25]. However, [26] provides the number of hidden nodes in the Feed-Forward Neural Network (FFNN) (in FFNN information must flow from input to output in only one direction with no back-loops). [26] Proved that any bounded continuous function in

R^n can be uniformly approximated within any error by using one hidden layer network having $2n + 1$ units (nodes) [26].

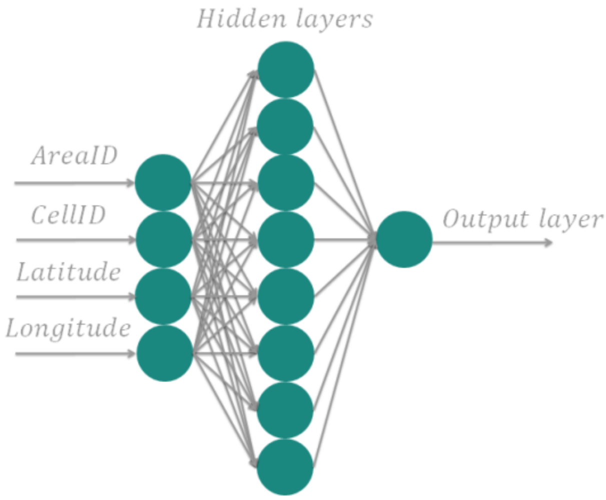

Figure 3.3: General Architecture of Neural Network

For the architecture of FFNN with multi hidden layers [26] shows that the number of hidden nodes in the first hidden layer is $2n + 1$ nodes, while the number of hidden nodes in the second hidden layers is $2\,(2n + 1) + 1$ units. In general, the number of hidden nodes equal to double hidden nodes contained in the previous layer plus one. The numbers of input neurons are equal to the dimension of the domain [27]. The numbers of output neurons are equal to the dimension of the range [27].

3.6 Training of Neural Networks

The training of any ANN operations is an adaptive process of the network by adjusting the weights associated with each neuron [28]. The training is conducted by specialized algorithms to change the weights. This is called conditioned weight which happens through a process called the training phase [28]. The initial weights are selected randomly.

All training algorithms used in ANNs are classified into four distinct types [47]:

Supervised Training: Which incorporates an external teacher (trainer), so that each output neuron is told what its desired response to input signals ought to be [28]. During the training process, global information may be required.

Unsupervised Training: Uses no external teacher and is based upon only local information

[28]. It is also referred to as self-organization, in a sense that it self-organizes data presented to the network and detects their emergent collective properties [28].

Semi-Supervised Training: Combines supervised and unsupervised training. Part of the weights is usually determined through supervised training while others are obtained through unsupervised training [28].

Reinforcement Training: Aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk [47]. It continuously learns from the environment in an iterative fashion [47].

Throughout this work we used supervised training algorithms.

3.7 Training Algorithm for Neural Networks

In this Section, the description of the algorithms that are commonly used for prediction problems is discussed. These algorithms are used for evaluating the network final results performance by the Mean Square Error (MSE) and the gradient method [36]. The Gradient Descent with Adaptive Learning algorithm, the Bayesian Regularization (BR) algorithm, and the Modified Levenberg-Marquardt (LM) algorithm are selected because the first one mentioned has the adaptive learning technique to obtain better results faster, the second one is used when having large data set but it takes a lot of time to obtain training results and the third one is used for larger data set, it takes a few number of epochs (the number of iterations) to obtain training results and a more accurate result is obtained.

3.7.1 Gradient Descent with Adaptive Learning Algorithm

The Gradient Descent with Adaptive Learning algorithm is used to minimize the error function [36]. The error function (3.4) is used to calculate and evaluate the differences between the output signals and the required target signals. This method is based on calculating the partial derivative of the error function [36]. In connection with each weight $w_{i,j}$ the steepest descent of a direction is calculated. It all results in obtaining a gradient vector giving the steepest increasing direction. The newly updated values of $\Delta w_{i,j}$ are derived from the obtained gradient vector with a negative sign. The error function is calculated as follows [36]:

$$E = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{N_L} (t_j - a_j)^2 \tag{3.4}$$

where t_j and a_j are the target and the output signals of a neuron j and N_L defines the number of output neurons. The parameter L represents the numbers of hidden layers.

The gradient direction along with a step size is calculated as follows [36]:

$$\Delta w_{i,j}(n) = -\eta \frac{\partial E(n)}{\partial w_{i,j}(n)} \tag{3.5}$$

where the parameter η defines the learning rate which indicates the speed of the learning process. The $w_{i,j}$ parameter represents the network weights from value i in layer l to value j in layer $l+1$.

Each iteration can be divided into three parts [55]:

1) The forward pass

   In this part, a pattern is introduced to the input layer. The pattern goes through all the layers until it reaches the output layer and creates the obtained result. The activation function is dependent on every i step of the iteration process in layer l and is calculated using the sigmoid function.

2) The generalized delta rule

   This step describes the calculation of the local gradients.

3) The final updating of the weights

   Finally, all the adjusted weights from the previous step are updated in the whole network and a new iteration process is carried out again from step 1.

The iteration process can be carried out in two variants, on-line training and batch training [19]. Both are the same approaches except for the fact that on-line training processes all the three steps mentioned above in all iterations. On the other hand, in batch training, only the first two steps mentioned above are used during the iteration process. It means that the final updating of the weights, the third step, is not performed at all iterations but at the end of an epoch. The final results are calculated from the sum of the collected local gradient values.

3.7.2 Bayesian Regularization (BR) Algorithm

BR is able to obtain lower MSEs than any other algorithms for functioning approximation problems [44]. BR has an objective function that includes a residual sum of squares and the sum of squared weights to minimize estimation errors and to achieve a good generalized model [45, 46]. This algorithm typically requires more time but can result in good generalization for difficult, small or noisy datasets. Training stops according to adaptive weight minimization (regularization). The network training function *trainbr* updates the weight and bias values according to LM optimization.

Typically, training aims to reduce the sum of squared errors. However, regularization adds an additional term: the objective function becomes $F = \beta E_D, +\alpha E_W$, where E_D, is the sum of squared error E_W, is the sum of squares of the network weights, and α and β are objective function parameters [57]. The relative size of the objective function parameters indicates the emphasis on training. If $\alpha \ll \beta$, then the training algorithm will drive the errors smaller. If $\alpha \gg \beta$, training will emphasize weight size reduction at the expense of network errors, thus producing a smoother network response [44]. The main problem with implementing regularization is setting the correct values for the objective function parameters. In [45] an extensive work on the application of Bayes' rule to Neural Network training and to optimizing regularization has been done. In the following, we will show the main results as they apply to our problem. In the Bayesian framework, the weights of the network are considered random variables. After the data is taken, the density function for the weights can be updated according to Bayes' rule [57]:

$$P\left(w|D,\alpha,\beta,M\right) = \frac{P\left(D|w,\beta,M\right)P\left(w|\alpha,M\right)}{P\left(D|\alpha,\beta,M\right)} \tag{3.6}$$

where D represents the data set, M is the particular Neural Network model used, and w is the vector of network weights [57]. $P\left(w|\alpha,M\right)$ is the priori density, which represents our knowledge of the weights before any data is collected. $P\left(D|w,\beta,M\right)$ is the likelihood function, which is the probability of the data occurring, given the weights w. $P\left(D|\alpha,\beta,M\right)$ is a normalization factor, which guarantees that the total probability is 1.

If we assume that the noise in the training data set is Gaussian and that the prior distribution for the weights is Gaussian, the probability densities can be written as [57]: $P\left(D|w,\beta,M\right) = \frac{1}{Z_D(\beta)}exp\left(-\beta E_D\right)$ and $P\left(w|\alpha,M\right) = \frac{1}{Z_w(\alpha)}exp\left(-\alpha E_w\right)$, where $Z_D\left(\beta\right) = \left(\frac{\pi}{\beta}\right)^{\frac{n}{2}}$ and $Z_w\left(\alpha\right) = \left(\frac{\pi}{\alpha}\right)^{\frac{N}{2}}$. If we substitute these probabilities into (3.6), we obtain [57]:

$$P\left(w|D,\alpha,\beta,M\right)=\frac{\frac{1}{Z_w(\alpha)}\frac{1}{Z_D(\beta)}exp\left(-\left(\beta E_D+\alpha E_w\right)\right)}{Normalization\ Factor}=\frac{1}{Z_F\left(\alpha,\beta\right)}exp\left(-F\left(w\right)\ \right) \tag{3.7}$$

In this BR, the optimal weights should maximize the posterior probability $P\left(w|D,\alpha,\beta,M\right)$. Maximizing the posterior probability is equivalent to minimizing the regularized objective function $F=\beta E_D+\alpha E_W$. The Bayesian optimization of the regularization parameters requires the computation of the Hessian matrix H of $F\left(w\right)$ at the minimum point w^{MP}.

Here are the steps required for Bayesian optimization of the regularization parameters [45]:

Step 0: Initialize α, β and the weights. We choose to set $\alpha=0$ and $\beta=1$ and use the Nguyen- Widrow [57] method of initializing the weights. After the first training step, the objective function parameters will recover from the initial setting.

Step 1: Take one step of the LM algorithm to minimize the objective function $F\left(w\right)=\beta E_D+\alpha E_W$.

Step 2: Compute the effective number of parameters $\gamma=N-2\alpha tr\left(H\right)^{-1}$ (where γ is the effective number of parameters and N is the total number of parameters in the network) making use of the Gauss-Newton approximation to the Hessian available in the LM training algorithm: $H=\nabla^2F\left(w\right)\approx 2\beta J^TJ+2\alpha I_N$, where J is the Jacobian matrix of the training set errors and I is the identity matrix.

Step 3: Compute new estimates for the objective function parameters $\alpha=\frac{\gamma}{2E_W(w)}$ and $\beta=\frac{n-\gamma}{2E_D(w)}$.

Step 4: Now iterate steps 1 through 3 until convergence.

We should bear in mind that with each estimate of the objective function parameters the objective function is changing; therefore, the minimum point is moving. If traversing the performance surface generally moves toward the next minimum point, then the new estimates for the objective function parameters will be more precise. Eventually, the precision will be good enough that the objective function will not significantly change in subsequent iterations. Thus, we will obtain convergence. The drawback of the BR algorithm is that it requires the computation of the Hessian matrix of the performance index and it takes too much time to obtain the training result.

3.7.3 Modified Levenberg-Marquardt (LM) Algorithm

Although BR based training is accurate it is very slow as well and even more complications appear at the solution. On the other hand, the LM algorithm [28] which is derived from Hessian based algorithms [31] can investigate data more in detail and provide even more accurate and subtle features. LM was specially developed for faster convergence in back-propagation algorithms.

The Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field [31]. It is closely connected with the Jacobian matrix. The LM method uses the following approximation to the Hessian matrix: $H \approx J^T J + \mu I$, where μ is always non-negative, called the combination coefficient, I is the identity matrix. From the Hessian matrix it is known that the elements on the main diagonal of the approximated Hessian matrix, will be larger than zero. The gradient can be computed as $g = J^T E\left(w_k\right)$, where J is the Jacobian matrix which contains the first derivatives of the error function (performance function) with respect to the weights and a vector of network errors, $E\left(w_k\right)$ [28].

It solves problems with the local curvature of a function of many variables. Hessian matrices are widely used in large-scale optimization problems. The convergence of the training process is faster because the Hessian does not vanish at the solution [31]. The general performance of errors and the regularization process are both, in a small modification, involved in this algorithm and that is also the reason why this algorithm is selected for this work.

The general training process of LM algorithm is as follows [26, 27]:

Step 0: Weights initialization

Randomly initialize weights and biases, respectively, on all the connections (paths) in the ANN.

Step 1: Distributing inputs feed forwarding

Each inputs $(x : x_1,\ x_2, x_3, \ldots, x_n)$ and targets $(t : t_1,\ t_2, t_3, \ldots, t_n)$ are sent to all neurons in the hidden layer.

Step 2: Summation in each neuron in hidden layer

Each neuron in this layer computes the summing for the inner product of its weights and bias with the inputs.

Step 3: The inner product transferring in the hidden layer

Take suitable transfer function for summing the value in step 2. Then its output is sent to the output layer as an input.

Step 4: Summation at the adder in each neuron in output layer

The neurons in this layer, sums its weighted inputs.

Step 5:The inner product transferring in external layer

The transfer function of the neuron in this layer computes its output which is also the output of the overall network.

Step 6: Squared error computation

After the output neuron computed the output, and then the error can be calculated.

Step 7: Checking halt condition

If the error is less than a specific value, the training stops and the weights and biases are sent. Otherwise, the training proceeds to the next step until the error becomes less than a specific value.

The LM algorithm is chosen for this work because its performance is better than the other two algorithms which are the Gradient Descent with Adaptive Learning algorithm and the BR algorithm. The flow chart of the LM algorithm is shown in Figure 3.4 [28].

After a review of researches done for prediction of users movement in mobile networks the problem identified was, design and selection of best features for an accurate result. Our proposed approach aims at accurately predicting the users movement by training, validating and testing features from given data using Non-Linear Autoregressive Neural Network with External inputs (NARXNN). Before working with the NARXNN it is necessary to take a look at the core of the network which is the dynamic network.

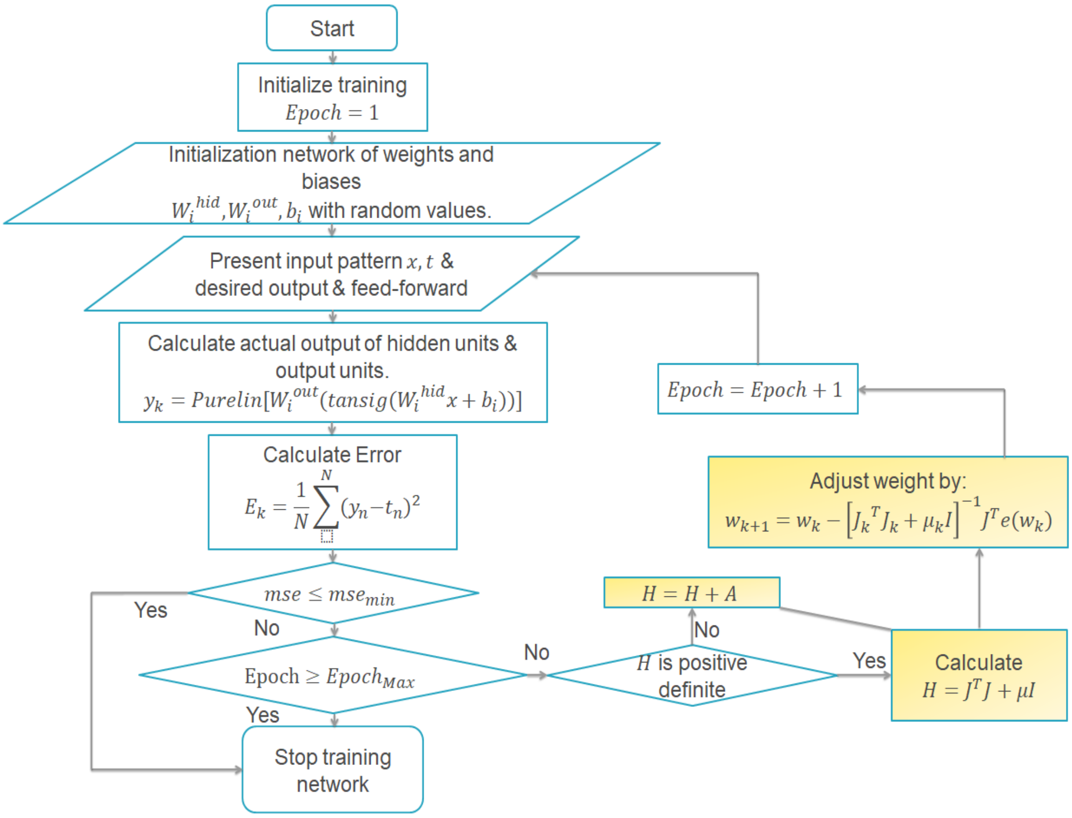

Figure 3.4: Modified Levenberg-Marquart Algorithm Flowchart [28]

3.8 Dynamic Networks

There are two groups of NARXNN network [20]: the feedforward one and the feedback one among the neurons [54]. To understand it correctly there are three types of networks: static, feed-forward dynamic, and recurrent dynamic networks.

Very briefly explained, static networks provide us with the same length of the output vector as the input vector. On the other hand, feed-forward dynamic networks give us longer response to the input data which is caused by the memory [54]. The output is not affected only by the input data but also by previous delayed values of the input vector delayed by 1 time step. Without any feedback connection, the network has only finite amount of previous values that is generally called Finite Impulse Response (FIR) filter.

Finally, feed-forward dynamic networks with recurrent connection has even longer output data response [54]. This recurrent connection allows the network to investigate the data relationships more in detail with better accuracy. The input data is not delayed and the recurrent connection is delayed by 1 time step [54]. The process is the same as in non-recurrent feed-forward dynamic networks except for the fact that the response value never reaches zero value. That is called Infinite Impulse Response (IIR) filter.

3.9 Non-Linear Autoregressive with External Input Neural Network (NARXNN)

The NARXNN provides a very accurate chaotic time series prediction [20] that perfectly fits this work because it solves the mobile users movement prediction problem. This network with delayed inputs, delayed recurrent (feedback) outputs, the nonlinearity and dynamic character allows to compute and determine tasks that are almost impossible to solve for conventional methods or linear (time-invariant) systems [19].

NARXNN architecture creates feedforward back-propagation with feedback from the output unit to the input unit [20]. The first hidden layer receives weight from input unit. Each subsequent layer receives weight from the previous layer [20]. The NARXNN can be carried out in one of the following two modes: Series-Parallel (SP) Mode and Parallel (P) Mode [20]. In SP mode, the output's regression is formed only by the actual values of the system's output [20]. In P Mode, estimated outputs are fed back and included in the output's regression [20]. The P mode NARXNN architecture is represented in Figure 3.5.

The dynamic Multi-Layer Perceptron (MLP) Neural Network consists of an input vector composed of past values of the NN input and output [25]. This is the approach by which the MLP can be considered as a NARXNN model of the system. This way of introducing dynamics into a static network has the advantage of being simple to implement [25]. To deduce the dynamic model of realized Neural Network system, NARXNN P-type model [25] can be represented as follows:

$$y\left(k+1\right) = f_{ANN} = \left(y\left(k\right), \dots y\left(k-n+1\right), u\left(k\right), \dots u\left(k-m+1\right) + \varepsilon\left(k\right)\right. \tag{3.8}$$

where $y\left(k+1\right)$ is model predicted output, f_{ANN} is a non-linear function describing the system behavior, $y\left(k\right), u\left(k\right), \varepsilon\left(k\right)$ are output, input and approximation error vectors at the time instances k, n and m the order of $y\left(k\right)$ and $u\left(k\right)$ respectively [25].

The order of the process can be predicted from experience. Modeling by Neural Network depends on the considerations of an approximate function of f_{ANN}. Approximate dynamic model is developed by adjusting a set of connection biases (b) and weight (w) via training function defined as MLP network.

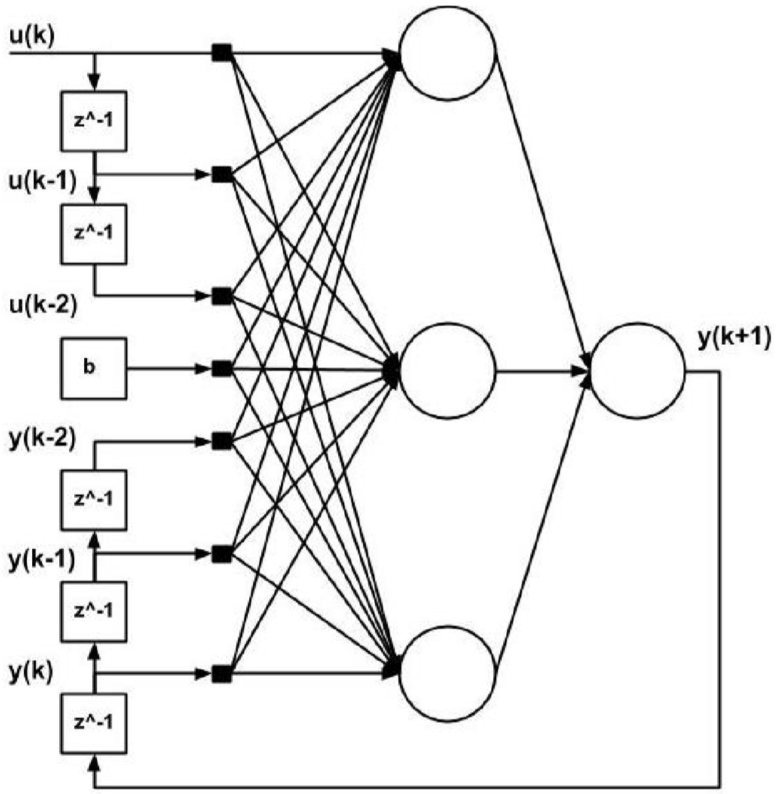

Figure 3.5: NARXNN Architecture [20]

3.10 Time Series Prediction

A time series is a sequence of vectors, $x\,(t)\,t = 0, 1, \ldots$ where t represents elapsed time [51]. For simplicity we will consider here only sequences of scalars, although the techniques considered generalize readily to vector series. Theoretically, x may be a value which varies continuously with t , such as a location [52]. In practice, for any given physical system, x will be sampled to give a series of discrete data points, equally spaced in time. The rate at which samples are taken indicates the maximum resolution of the model; however, it is not always the case that the model with the highest resolution has the best predictive power, so that superior results may be obtained by employing only every n^{th} point in the series [51].

Since it is possible to over-sample a data stream, [51] suggest computing the average mutual information at varying sampling rates, and taking the first minimum as the appropriate rate. The mutual information acts as the equivalent of the correlation function in a non-linear domain. A minimum shows a point at which sampled points are maximally decorrelated – a small change of sampling rate in either direction will result in increased correlation [51].

The work in Neural Networks has concentrated on forecasting future locations of the time series from values of x up to the current time [52]. Formally this can be stated as: find a function f N : R $\rightarrow R^N \rightarrow R$ such as to obtain an estimate of x at time $t+d$, from the N time steps back from time t, so that [52]:

$$x\left(t+d\right)=f\left(x\left(t\right),x\left(t-1\right),\ldots,x\left(t-N+1\right)\right) \tag{3.9}$$

$x\left(t+d\right)=f\left(y\left(t\right)\right)$ where $y\left(t\right)$ is the $N-$ ary vector of lagged x values. Normally d will be one, so that f will be forecasting the next value of x.

The proposed approach which is the NARXNN with dynamic network and time series prediction is used for this work.

3.11 Applications of Neural Networks

Neural Networks are applied in many areas and a few of them are discussed below [56, 58]:

Intelligent Control: A problem of concern in industrial motor control is the ability to predict system failure [58]. Neural Networks have been used in many vehicular applications including trains and automatic gear transmission in cars.

Signal Processing: In digital communication systems, distorted signals cause inter-signal interference, one of the first commercial applications of Neural Networks was to suppress noise cancellation and it was implemented to remove the noise from the telephone line signal [56].

Pattern Analysis: Real-time audit of bandwidth systems, images and data management can also be tackled with the help of Neural Networks [56].

Medicine: An auto associative network is developed to store large amount of medical records, each of which includes information on the symptoms, diagnosis and treatment of a specific case. When trained network is presented with a set of symptoms, the network finds the best diagnosis and treatment [58].

Classification: It is the assignment of each object to specific class, which is an important aspect in image classification for example recognition of printed or handwritten characters [58].

Financial Forecasting: One of the most desirable applications of Neural Networks have been in financial applications [56]. The idea if one dealer can predict market values or assess risk better than his competitors even by a very small amount, leads to an expectation of huge financial benefit. Much of the work in this area is based on the time series prediction.

Function Approximation: Many computational models can be described as functions mapping the input vectors to numerical outputs [58]. Neural Networks are employed to construct approximately the same output for the given input vector based on the available training data.

Condition Monitoring: Neural Networks can also be used in predicting the occurrence of malfunction of item [56].

Process Monitoring and Control: The techniques applied in processes such as chemical and biochemical, pharmaceutical, water purification, food and beverage production, and power distribution [58].

Speech Recognition: In recent years, speech recognition has received enormous attention and involves three modules namely the front end which samples the speech signals and extracts the data, the word processor which is used for finding the probability of words in the vocabulary that match the features of the spoken words and the sentence processor which determines if the recognized word makes sense in the sentence [56,58].

Chapter 4

Experimentations

This Chapter has five Sections; in the first Section, there is an explanation of the network that has to be set in order to achieve the users movement prediction simulation. In the second Section, a brief description of the performance parameters are given. In the third Section, the simulation results obtained for mobile users movement prediction are presented. In the fourth Section a discussion is made based on the results obtained in the third Section. In the last Section, the challenges that occurred during this work and the limitations are explained.

4.1 Evaluation Framework

This Section focuses on data collection, data pre-processing, creating network object, configuring the network inputs and outputs, training the network, optimizing and validating the network.

4.1.1 Data Collection

The prediction, analytics methods in general, is strongly dependent on a selected data set and that is the reason why the right data set has to be selected and data pre-processing should be achieved. For this work, three types of datasets were collected. These are the Massachusetts Institute of Technology (MIT) reality mining data set, Mobile Data Challenge (MDC) data set and OpenCelliD data set, and the performance of OpenCelliD was observed. The OpenCelliD data was chosen because it contains Ethiopia's CellID information particularly, Addis Ababa's, whereas the other two contain America's and Switzerland's data.

4.1.1.1 Massachusetts Institute of Technology (MIT) Reality Mining Data

The Reality Mining project was conducted from 2004-2005 at the MIT Media Laboratory [29]. The MIT reality mining data set was obtained in **.mat** format in MATLAB. The reality mining data set consists of 52 fields and 94 rows. The Reality Mining study followed ninety-four individuals using mobile phones pre-installed with several pieces of software that recorded and sent the researcher data about call logs, Bluetooth devices in proximity of approximately five meters, cell tower IDs, application usage, and phone status [29]. Of these 94 individuals, 68 were colleagues working in the same building on campus (90% graduate students, 10% staff) while the remaining 26 individuals were incoming students at the university's business school [29]. The individuals volunteered to become part of the experiment in exchange for the use of a high-end smartphone for the duration of the study. Individuals were observed using these measurements over the course of nine months and included students and faculty from two programs within a major research institution [29]. Self-report relational data was collected from each individual, where individuals were asked about their proximity to, and friendship with, others [29]. The drawback of using this data for this work is that the geo-location of the individuals was not specified in the data set and the data set is not local data.

4.1.1.2 Mobile Data Challenge (MDC) Data

The MDC data was collected on a 24/7 basis over months [30]. In the Dedicated Track of this competition, which included the task of next place prediction, the raw location data is transformed into the sequence of visits to symbolic places [30]. The users in MDC data are sampled into three separate sets. The training data set (called setA) consists of mobile phone data collected from 80 individuals during a period of time varying from a few weeks to two years [30]. The unseen data for each participant in setA is used to build the test data set (called setC), where the unseen data corresponds to the continuation of setA (in time). The setC ground truth was never released after the challenge. However, the validation set (called setB) was released and is used to evaluate the results [30]. The validation set contains visits that were randomly chosen from the last part of setA (in time). The validation set was built by filtering data in setA with time intervals corresponding to the randomly chosen visits [30]. The MDC database has 5 main types of data: environmental, personal, phone usage, phone status, and visits data [30]. This data is represented across 18 tables, with more than 130 raw attributes, and is approximately 50 GB in size [30].

The MDC data was obtained by signing license agreement between *Instituto de Investigación Agropecuaria de Panamá (IDAP) Research Institute* and *Addis Ababa University (AAU)*. But it is not used for this work because similar work is done in Switzerland and the data set is not local data.

4.1.1.3 OpenCelliD Data

The OpenCelliD project was primarily created to serve as a data source for global system for mobile communication (GSM) localization [37]. As of October 2017, the database contained almost 36 million unique GSM CellIDs [37]. More than 75,000 contributors have already registered with OpenCelliD, contributing millions of new measurements every day in average to the OpenCelliD database [37]. The OpenCelliD data set was obtained in **.csv** format in excel. We first extracted Ethiopia's OpenCelliD data and then used their own CellMapper web application to Extract Addis Ababa's data. Extracted OpenCelliD Addis Ababa's data set consists of 14 columns, 14002 rows, and 100 contributors. CellID is a unique identifier of each base transceiver station (BTS) within a Location area code displayed as areaID. This data set was selected for this particular research work.

Table 4.1: OpenCelliD Addis Ababa's Statistics [37]

City Name	Total Cells	GSM	CDMA	UMTS	LTE
Addis Ababa	14,325	7,786	0	6,390	149

The OpenCelliD data set has got its own shortcoming that is the type of antenna of each cell is not known but for this particular work, we assumed that the antenna type used is Omnidirectional Antenna. These prepared data are used to predict a mobile user's movement by the geo-location coordinates as there are enough correlations among the data. Data is time-variant, in other words, there is long-term time dependence. That makes it a suitable set of data for prediction investigation with NARXNN.

4.1.2 Data Pre-Processing

Mostly, data can be processed directly without using any pre-processing techniques and in some cases it is even more appropriate to do so. But in most applications it is necessary to select the right representation of data or remove some parts so that results are more accurate and trustworthy. Typically, removed parameters are the ones that have linear or non-linear trends (rising or decreasing values) or suffer from seasonality (patterns caused by a factor that is repeated over and over again) [19, 20].

Very important part of data selection is to find the mutual relationships among the data called correlations [19]. The idea is to pre-process the input data so that it is as predictable as possible. It is not always easy to do it and in many cases inner connections among data are hidden [20]. In that case, there are other Neural Networks like Self-Organizing Map (SOM) Neural Networks or its modifications that can solve this problem. Data correlations can be calculated as follows [19]:

$$R = \frac{\sum_{t=1}^{N} \left(y\left(t\right) - y\right) \left(y\left(t+k\right) - y\right)}{\sum_{t=1}^{N} \left(y\left(t\right) - y\right)^2)} \tag{4.1}$$

where $y\left(t\right)$ is an output vector and k stands for the time delay. To calculate the mean value of y the following equation is given [19]:

$$y = \sum_{t=1}^{N} \frac{y_t}{N} \tag{4.2}$$

Sometimes the outliers which are values that vary from the mean a lot can also cause problems for learning process. To get rid of this problem, the correlation described above is used to uncover hidden outliers and these are then deleted from the data set. Finally, the normalization process can be applied. It means that the input and output values are mapped into [0, 1]. This normalization step can significantly speed up the training and produce more accurate results.

4.1.3 Network Creation

The pre-processed data has to be loaded and then it needs to be changed into a typical Neural Network cell array form so that it can be processed by the NARXNN Matrix Laboratory (MATLAB) code. The fundamental parameters that have to be set are:

- Training algorithm
- Delays (Input and feedback delays)
- Number of neurons
- Open or closed loop NARXNN

After setting up the above parameters the created network has to be trained. Initially, the default values were used. For this research, the network is set to 4-8-1 architecture as shown in Figure 3.3. The first 4 input layers designate the columns in a given data set which are the Location AreaID, CellID, Latitude, and Longitude. The second 8 hidden layers specify the number of neurons (hidden layers) used for the network hence it will be adjusted depending on the performance of the network, and the last 1 output layer indicates the output obtained as a prediction of the mobile users movement.

4.1.4 Network Configuration

The dataset is separated into 3 parts; the training data is used by the network during training, and the network is adjusted according to its error, the validation data is used to measure network generalization, and to halt training when generalization stops improving, testing data have no effect on training and so provide an independent measure of network performance during and after training. The data set by default is divided in a proportion of 70% training, 15% validation and 15% testing. The network configuration parameters used for the Gradient Descent with Adaptive Learning algorithm are shown in Table 4.2.

Table 4.2: Network Configuration for Gradient Descent with Adaptive Learning Algorithm

Training function	*traingda*	**Training parameters**	
Architecture parameters		Maximum Epochs	200
Input delays	1 : 2	Maximum training time	Inf
Feedback delays	1 : 2	Performance	0
Number of neurons	4	Minimum Gradient	$1e-7$
Data division		Maximum validation checks	6
Training data	70%	Learning rate	0.001
Validation data	15%	Learning rate decrease	0.5
Testing data	15%	Learning rate increase	1.1
		Maximum performance increase	1.5

4.1.5 Network Training

The NARXNN can work in two modes. The first, the open-loop mode is generally used for the network training part when the output data is known [24]. The second, the closed-loop mode is convenient for prediction tasks as it uses the recurrent connection from the output layer to the input layer with given delays [24]. This mode provides a multi-step prediction based on previous open-loop mode and the external input. The closed-loop architecture can predict an arbitrary number of steps, but only as many as the input layers have time steps. Normally, the network gives $y\,(t-1)$ result at the same time when the network reads $y\,(t-1)$ input data [24]. For some applications (decision making), it would be convenient to be able to predict a time-step early so called Step-ahead prediction. So, the network would give $y\,(t-1)$ once $y\,(t)$ is ready for reading. It is done by removing one delay, in other words, by shifting the whole time-line one step to the left so that the minimum delay starts at 0 value instead of 1 .

There are two problems that should be taken into account and alleviated before training. First, static Neural Networks can be frozen in local minima [32]. We do not need to be worried about that as the NARXNN solves this problem by having dynamic character. Second, very often the number of connections and weights overcomes the number of parameters for a given task [32]. This situation can lead to “Overtraining” or “Overfitting” that can provide false or inaccurate results. It can be successfully solved by regularization techniques in combination with value performance reduction. Classical MSE error is then replaced by MSEreg as follows [32]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{4.3}$$

$$MSW = \frac{1}{n}\sum_{j=1}^{n}(w_j)^2 \tag{4.4}$$

$$MSEreg = \varrho MSE + (1-\varrho)\, MSW \tag{4.5}$$

where the number of values in the error vector is given by the N parameter for the MSE and the n parameter for the Minimum Squared Weight (MSW), t_i represents the target vector and y_i indicates the predicted vector, and the w_j parameter is the weights while the ϱ parameter is the performance ratio [32].

4.1.6 Development Tool

Many development tools are used in the process of this kind of research work. The tools that have been used include Tensorflow, MATLAB, and Waikato Environment for Knowledge Analysis (WEKA). For this work, MATLAB is chosen, since MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, c#, Java, Fortran, and Python.

4.2 Performance Matrices

Non-Linear Regression Function: Defines a non-linear regressive model of function f [53]:

$$X_R = f\,(X_i, \beta) + \varepsilon \tag{4.6}$$

where X_i are the model parameters, β nonlinear computed parameter estimates of the model and ε represents the error terms.

Autocorrelation Function: Indicates an autocorrelation model, where vector y is shifted in time and compared with itself (It is a function for finding repeating patterns) [19]:

$$R_{yy}\left(k\right) = \frac{1}{T-1}\sum_{t=1}^{T-k}\left(y_t - y\right)\left(y_{t+k} - y\right) \tag{4.7}$$

where y_t is the input signal, y_{t+k} is the lagged input signal and y is the mean of the input signal.

Mean Square Error (MSE): Measures the average of the squares of the “errors“ [28]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(t_i - y_i\right)^2 \tag{4.8}$$

where t_i defines the target value and y_i is a predicted value.

4.3 Results

This Section illustrates the results obtained for a mobile user movement prediction using NARXNN. Three types of algorithms are simulated which are the Gradient Descent with Adaptive Learning algorithm, the BR algorithm, and Modified LM algorithm and the results of each are discussed.

4.3.1 Gradient Descent with Adaptive Learning Algorithm

In this Section, the result based on the Gradient Descent with Adaptive Learning algorithm is obtained. The optimized parameters are input delays, feedback delays and number of neurons as shown in Table 4.2. These parameters are selected because they affect the network results the most when they are changed.

Figure 4.1 illustrates real movement of the user (blue line) and predicted movement (red line) in latitude. The trend of the predicted line movement follows the target line. The predicted line suffers from significant jitter (unwanted noise) which would cause visible problems with mobile user movement prediction. The two reasons why this is caused is one, due to the vanishing gradient during the learning process and two, by the fact that this algorithm is unable to predict accurately, in terms of this prediction task.

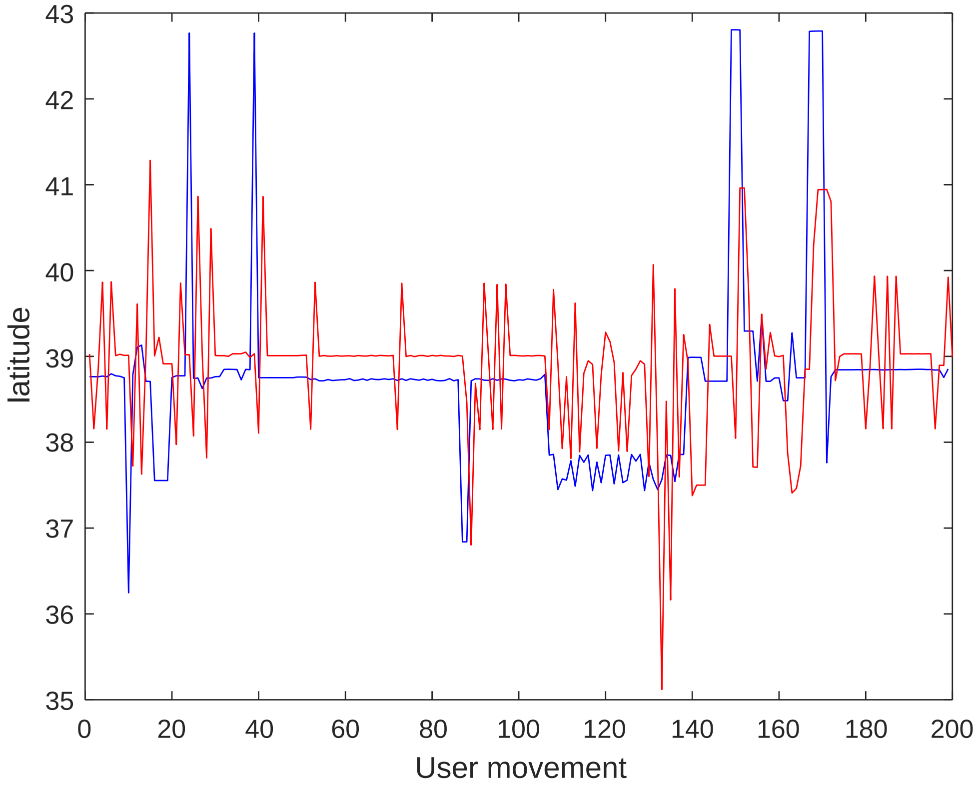

Figure 4.1: Movement Prediction in Spatial Coordinates

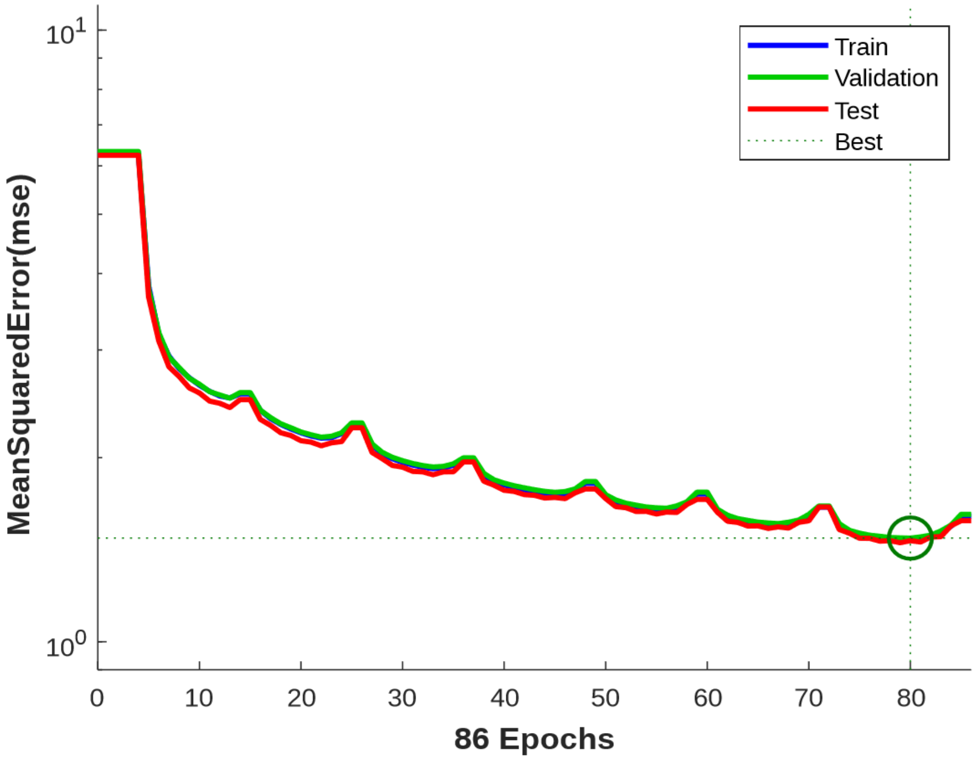

Figure 4.2: Mean Square Error (MSE) Performances

Figure 4.2 represents the MSE during the training process, the MSE of the validation process, and the MSE for the test process. The dotted line represents the minimum MSE and the number of iterations required to reach the minimum validation MSE. Figure 4.2 depicts how much iteration are desired to get the minimum validation MSE of around 1.5 which is not satisfactory result from performance point of view. The MSE should be close to zero for accurate result. And also it is abundantly clear that the network training takes a lot of time, 80 epochs (number of iterations), to obtain the best validation performance. According to the curves the result shows that their is slight decrease in the MSE during the training.

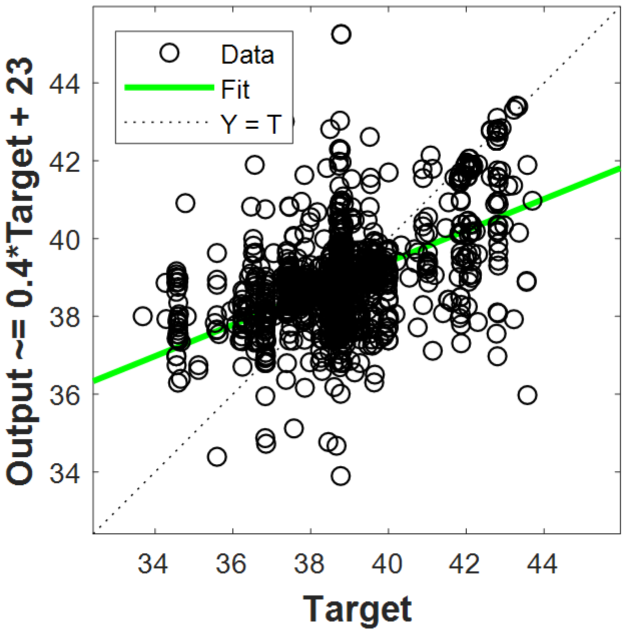

Figure 4.3: Regression Performance

Figure 4.3 shows the regression (4.6) toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (green line). It also illustrates the scattering of the real movement indicating very inaccurate result in terms of mobile users movement prediction planning depending on type of use case. For ideal case, the X_R parameter should be one and the green line should fit with the dotted line. The result of the regression is 0.5229 which is the same as 52.29% similarity between the target and the predicted output. This type of resemblance is not enough for movement prediction tasks.

4.4 represents the correlated errors in time steps (blue bar lines) and the Confidence Limit line (red dotted lines) showing limits inside which the prediction is almost accurate. Figure 4.4 explains how the predicted errors of the calculated output and the target are related in time (4.7). The correlation value at zero lag indicates that the errors are the same if there is no time delay. The non-zero lags indicate some correlations that having inaccuracies during the training process according to the non-zero blue bar lines leading to inaccurate prediction.

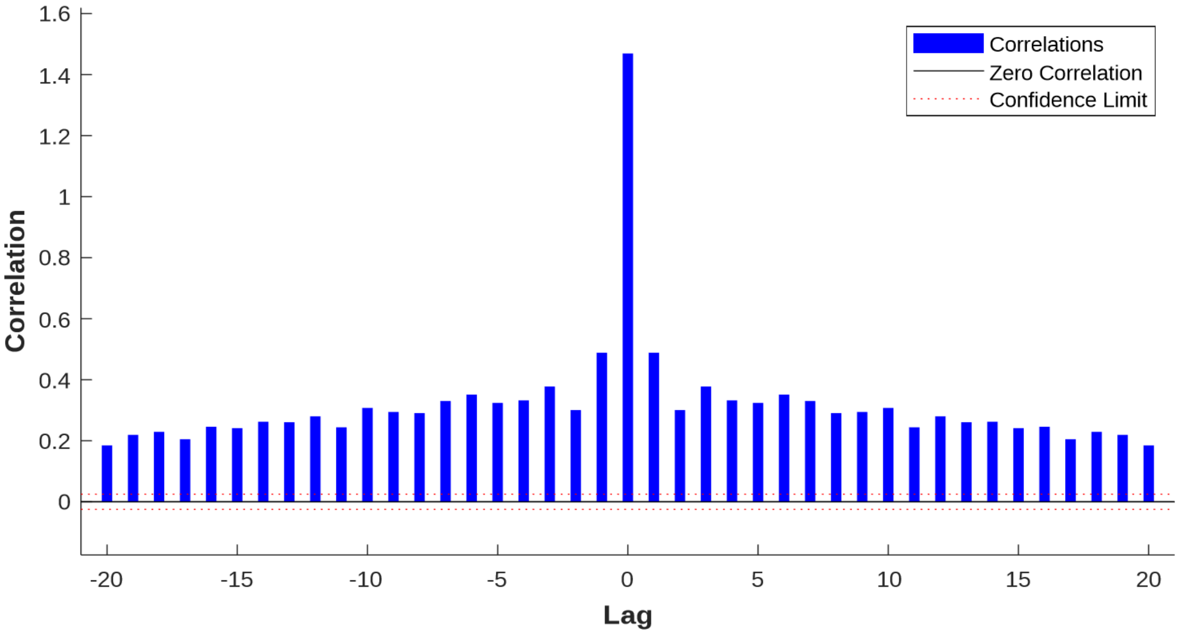

Figure 4.4: Autocorrelation of Errors

4.3.2 Bayesian Regularization (BR) Algorithm

In this Section, the BR algorithm explained in Chapter 3 is evaluated. The optimization is done using iteration process which is able to find the best possible combination of the network parameters. The optimized parameters are input delays, feedback delays and number of neurons as shown in Table 4.3. These parameters are selected because they affect the network results the most when they are changed.

Table 4.3: Network Configuration for Bayesian Regularization (BR) Algorithm

Training function	*trainbr*	**Training parameters**	
Architecture parameters		Maximum Epochs	200
Input delays	1 : 2	Maximum training time	Inf
Feedback delays	1 : 2	Performance	0
Number of neurons	6	Minimum Gradient	$1e-7$
Data division		Maximum validation checks	6
Training data	70%	Momentum learning (Mu)	0.001
Validation data	0%	Mu decrease ratio	0.1
Testing data	30%	Mu increase ratio	5
		Maximum Mu	1000

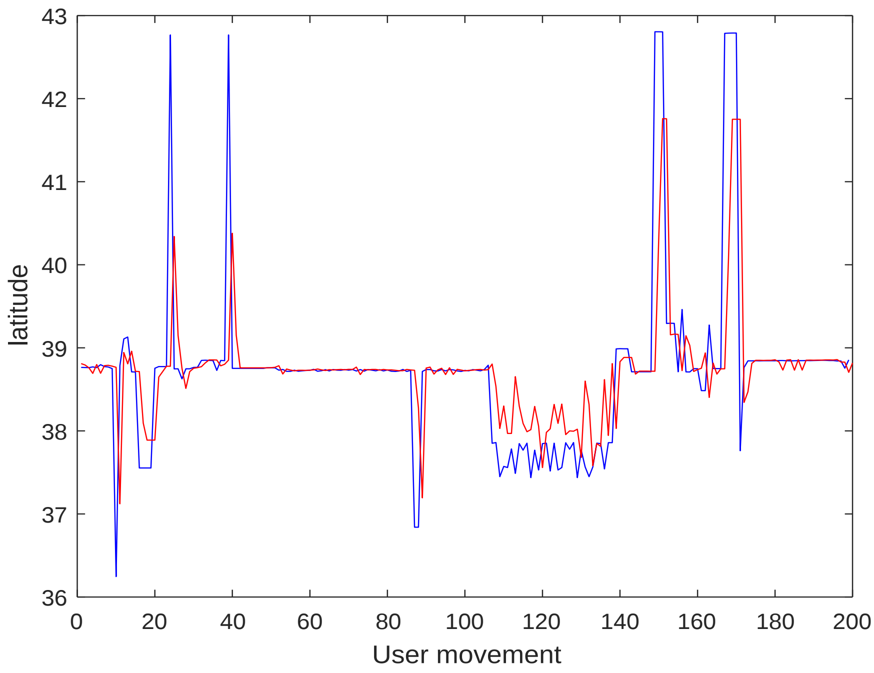

Figure 4.5: Movement Prediction in Spatial Coordinates

Figure 4.5 shows that the trend of the predicted line (red line) movement follows the target line (blue line). The predicted line does not suffer from such significant jitter like the previous algorithm which would bring less problems with mobile user movement prediction. Nevertheless, it is still not sufficient enough to use it for real industry process and secondly by the fact that this algorithm is unable to predict accurately in terms of this prediction task because it takes a lot of time to obtain the training result.

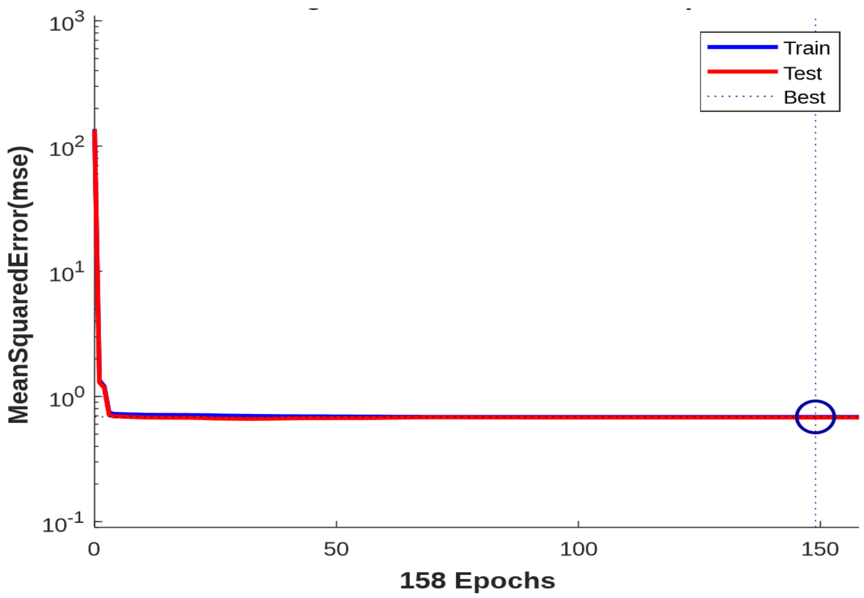

Figure 4.6: Mean Square Error (MSE) Performances

Figure 4.6 illustrates how much iterations are needed to get the minimum test MSE (4.8) of 0.7 which is significantly lower than the previous algorithm. The network does not perform any validation check in this type of algorithm because, the validation is used as a form of generalization and BR algorithm has its on built in generalization within the algorithm. It can be seen that in this case the network training to obtain the best test performance takes more time than the previous training process, 150 epochs. This method has its own benefits and limitations. Less cycles of the training could be used to get results faster but at the cost of even worse results.

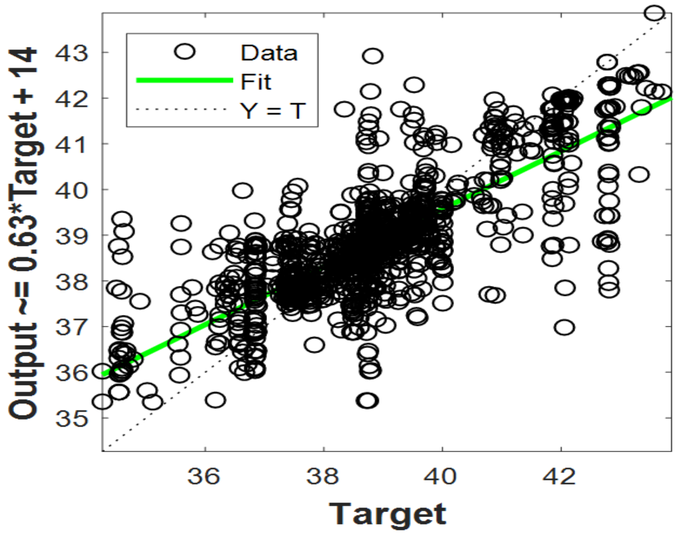

Figure 4.7: Regression Performance

Figure 4.7 shows the regression toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (green line). It also shows scattering of predicted real movement which can play a very significant role in terms of mobile users movement prediction use case. In this case, the result is not enough to be used in real industry cases but if the result is good enough in other cases it can be used. For ideal prediction use case, the green line should fit with the dotted line and the X_R parameter should be 1. The result of the regression is 0.79 which equals to approximately 79% similarity between the target and the predicted output. This similarity is better for movement prediction than the previous case.

Figure 4.8 describes how the predicted errors of the calculated output and the target are related in time. The correlation value in the zero lag indicates that the errors are the same if there was no time delay. Other non-zero lags shows some correlation that indicates some inaccuracies during training process. In this case, the blue lagged bar lines are closer to the Confidence Limit that shows an improvement in the result than the previous method.

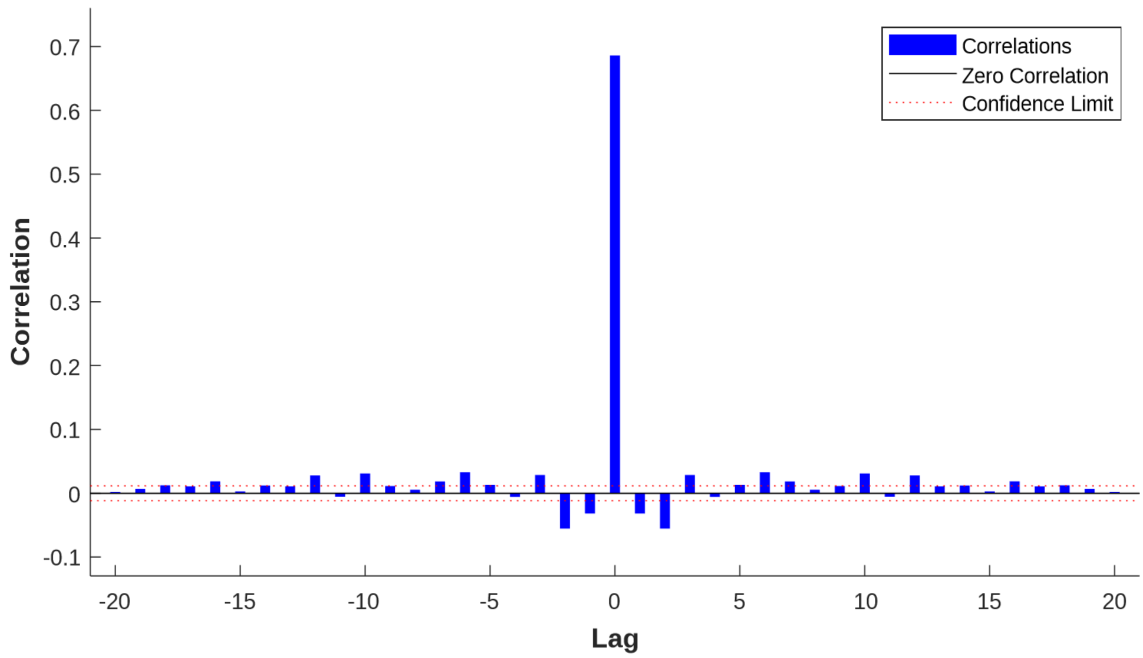

Figure 4.8: Autocorrelation of Errors

4.3.3 Modified Levenberg-Marquardt (LM) Algorithm

In this Section, the Modified LM algorithm explained in Chapter 3 is evaluated. The Modified LM algorithm uses momentum learning (indicates the speed of training data) as an alternative to the learning rate. This method is predominantly useful for obtaining optimized results in larger data sets with small number of epoch (iteration). The optimized parameters are input delays, feedback delays and number of neurons as shown in Table 4.4. These parameters are selected because they affect the network results greatly when they are altered.

Table 4.4: Network Configuration for Modified Levenberg-Marquardt (LM) Algorithm

Training function	***trainlm***	**Training parameters**	
Architecture parameters		Maximum Epochs	50
Input delays	1 : 3	Maximum training time	Inf
Feedback delays	1 : 3	Performance	0
Number of neurons	15	Minimum Gradient	$1e-7$
Data division		Maximum validation checks	6
Training data	70%	Momentum learning (Mu)	0.001
Validation data	15%	Mu decrease ratio	0.1
Testing data	15%	Mu increase ratio	5
		Maximum Mu	1000

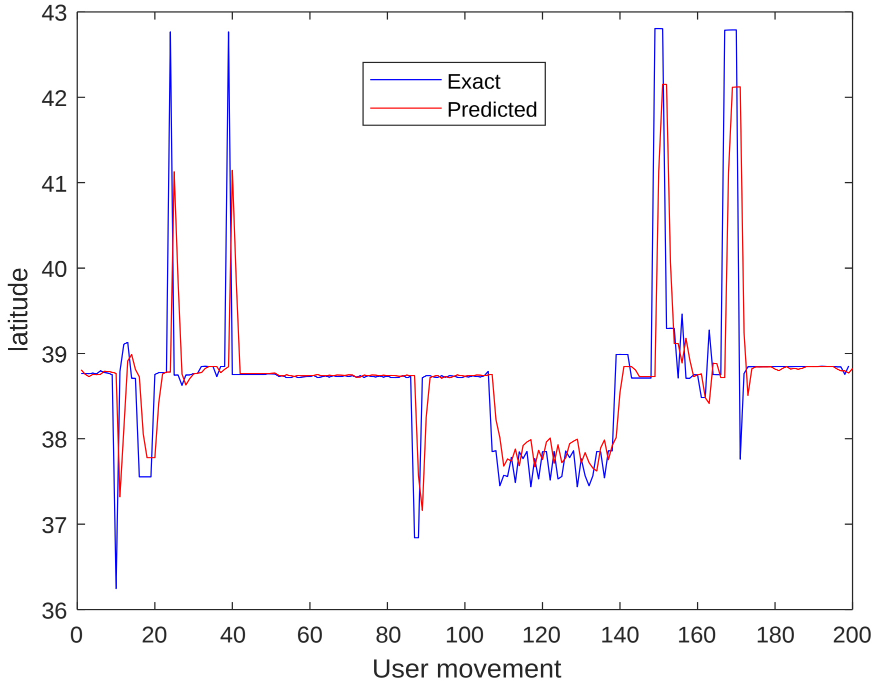

Figure 4.9: Movement Prediction in Spatial Coordinates

Figure 4.9 represents real movement of the user (blue line) and predicted movement (red line) in latitude. The trend of the predicted line movement follows the target line. The predicted line suffers from very small jitter (unwanted noise) which would cause minimum problems with mobile user movement prediction. This result is sufficient enough to be used for real industry cases. The Modified LM algorithm is apparently able to achieve the gradient disappearing difficulty and is suitable for this type of prediction tasks as can be seen from the results obtained.

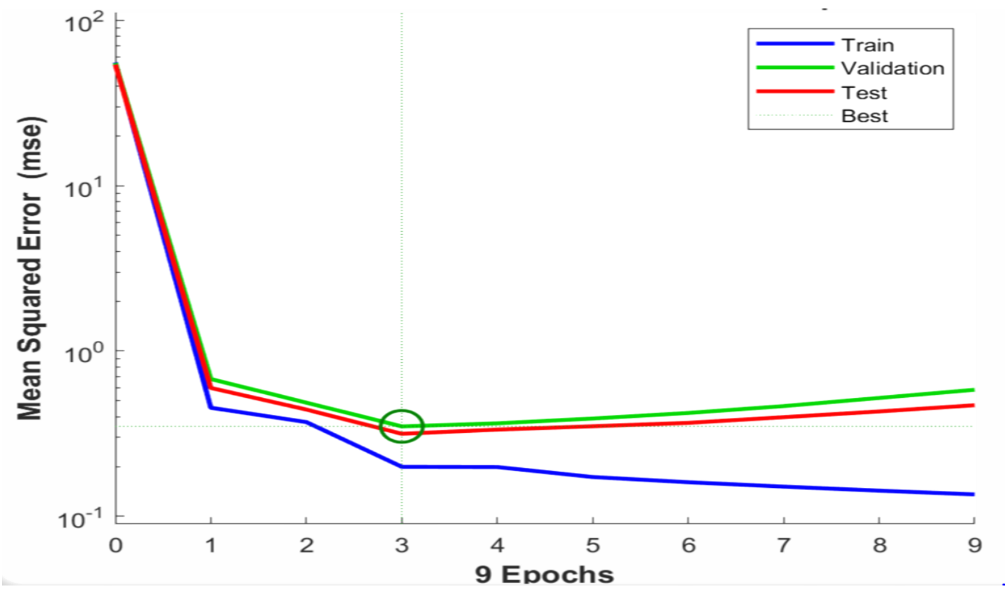

Figure 4.10: Mean Square Error (MSE) Performances

Figure 4.10 illustrates how much iteration is needed to get the minimum validation MSE of 0.32. The network doesn't indicate any overfitting or underfitting, since there is almost no difference between the validation and the train performance because they are very close to each other. This algorithm outperforms the Gradient Descent with Adaptive Learning algorithm and the BR algorithm, as seen in Section 4.3.1 and Section 4.3.2 respectively. The Modified LM algorithm terminates the training process much faster than the other algorithm, that is only 3 number of iterations are needed to obtain the best validation performance, which indicates that the result obtained is accurate compared to the other two algorithms.

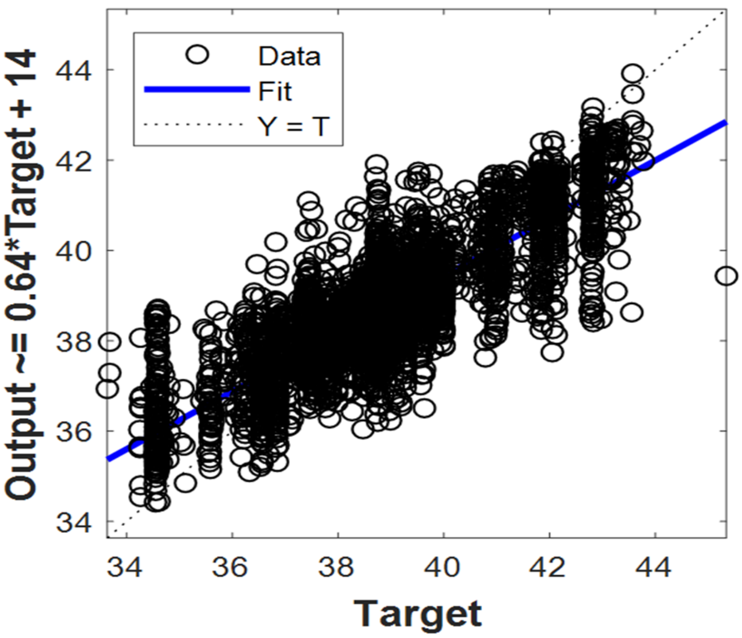

Figure 4.11: Regression Performance

Figure 4.11 shows the regression toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (blue line). It also illustrates the scattering of the predicted real movement which can play a very vital role in terms of mobile users movement prediction use cases. As it can be clearly seen the result demonstrates that a few deviations from the mean line implies the suitability of this algorithm to be used in real time prediction cases.

Ideally the blue line should fit with the dotted line and the X_R parameter should be equal to 1. The result of the regression is around 85% which indicates that there is a similarity between the target and the predicted output as compared to the previous regression results of the other two algorithms. A perfect result near to 99.9% can be obtained if we have very accurate data set and combine other conventional statistical methods.

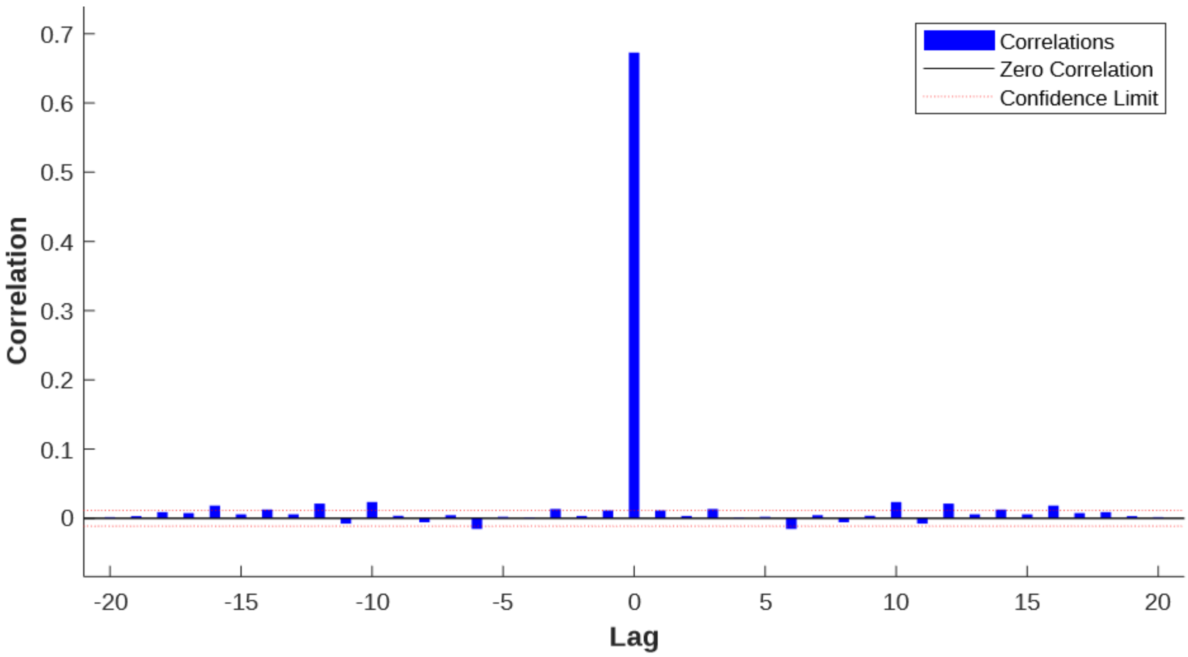

Figure 4.12: Autocorrelation of Errors

Figure 4.12 represents the correlated errors in time steps (blue bar lines) and the Confidence Limit line (red dotted lines) showing limits inside which the prediction is almost accurate. It explains how the predicted errors of the calculated output and the target are related in time. The correlation value at zero lag indicates that the errors are the same if there is no time delay. The non-zero lags indicate some correlations that having little inaccuracies during the training process according to the non-zero blue bar lines. Approximately perfect prediction is demonstrated in Figure 4.12, since blue lagged bar lines other than the zero lag are almost inside the Confidence Limit.

4.4 Discussion

In this research work three algorithms are used, which are the Gradient Descent with Adaptive Learning algorithm, the BR algorithm, and the Modified LM algorithm. The first algorithm does not show sufficient result for mobile users movement prediction as explained in Section 4.3.1 because of the gradient vanishing problem, the second algorithm needs a number of iterative process to stop the training and this is time consuming because it takes a lot of iterative processes to halt the training and the third one, the Modified LM algorithm, is widely used for mobile users movement prediction in telecommunications network in all aspects and also for wider type of prediction tasks. The summary of all the results which are train MSE, validation MSE, test MSE, regression and number of epochs (iterations) for the three algorithms used is shown in

Table 4.5.

Table 4.5: Results Discussion

	Results		
	Gradient descent algorithm	**Bayesian regularization algorithm**	**Levenberg-Marquardt algorithm**
Train MSE	1.52000	0.76173	0.19930
Validation MSE	1.50000	0.00000	0.34984
Test MSE	1.50000	0.68740	0.31564
Regression	0.5229	0.79000	0.84560
Number of iterations	80	150	3

Table 4.5 shows the differences between the used algorithms. Performance results displayed in the Table show the efficiency of the train, validation and test process respectively. The Gradient Descent with Adaptive Learning algorithm has no good results because the algorithm itself is not the best choice for mobile users movement prediction even though it is generally used for prediction tasks. The training takes 80 epochs to obtain the results; the regression is around 52% and the overall performance of the MSE is not sufficient for industry cases. The BR algorithm outperforms the Gradient Descent with Adaptive Learning algorithm in terms of the overall performance of MSE and the regression is about 79% for finding the best possible combination of the network parameters. Nevertheless, it needs more time to obtain results, 150 epochs. The third algorithm which is the Modified LM algorithm outperforms both previous algorithms in terms of MSE 0.31564 **,** regression 84.56% and only 3 iterations are needed to train the network. This result can be used for real telecommunication industry cases.

4.5 Challenges and Limitations

The challenge of doing such type of research is the exact type of data that can be applied for this work can't be easily accessed. And the prediction analytics is strongly dependent on the data that we use. Table 4.6 summarizes the challenges that appear at each step of prediction analytics research work.

The limitation of this work is selection of the right data and classical statistical methods which are very critical for high accuracy of the prediction result. So, the data that we used is limited to only OpenCelliD data set and the accuracy of our result is directly related to the type of data that we used.

Table 4.6: Challenges

Steps	**Challenges**
Access, explore and analyze data	• Data diversity Numeric, images, signals, text- not always tabular • Accessing the right data
Pre-process data	• Lack of domain tools Filtering and feature extraction, selection and transformation
Train models	• Time consuming Train many models to find the best result
Assess model performance	• Avoid pitfalls Over fitting, speed-accuracy-complexity trade-off
Iteration	• A whole lot of iterative processes to go through

Chapter 5

Summing up

5.1 Conclusion

In this work, an overview of LBS and Big Data are introduced in connection with the telecommunications industry. Big Data solution offers deeper insight to customer's behavior in telecommunications network and QoS through analytics tools. The work also gives a look at customer needs of the current market and comprehensive use cases for telecommunications industry. In this research, common prediction using Neural Network and its optimization for mobility prediction is used. The data that has been collected and the challenges that appear while doing prediction analytics researches are discussed in this work.

In this work, prediction using NARXNN and its optimization for mobility prediction has been also analyzed. In the first step, the Gradient Descent with Adaptive Learning algorithm has been observed in order to analyze its suitability to predict user movement. The results show that this type of algorithm is not efficient because the regression is 52.29% similarity between the target and the calculated output. It could lead to considerable problems in built-up areas and prediction would be inaccurate. The average train, validation and test MSE performance equals to 1.5 so we can conclude that the performance of this algorithm is not good.

Further, the BR has been proposed. This approach improves MSE performance to 0.7 which is more acceptable overall system performance for real industry usage but at the cost of slower learning process and the fact that the iteration process can take a lot of time for relatively small data set in comparison with a real telecommunications data flow. The accuracy of the prediction grows up to almost 79% that would be applied for specific use cases.

Lastly, the Modified LM algorithm (iterative selection of parameters for Neural Network) has been investigated in order to analyze its suitability to predict mobile users movement. The results obtained which are, the system average train, validation, test MSE and final regression were compared with the Gradient Descent with Adaptive Learning algorithm and the BR algorithm, and shown that the Modified LM algorithm, is perfectly convenient for mobile users movement prediction because it outperforms the previous solution in all aspects. The system average MSE equals to 0.32 which supports the fact that the accuracy performance of the prediction reaches almost 85%. The network needs only 3 epochs to be trained and obtain the best validation performance. This research assures quality optimization in terms of the selected algorithm. The gained results that are obtained can be used as a basis for Big Data processing in telecommunications using Neural Networks.

5.2 Recommendation

This work can be further extended if Ethio-telecom data scientists work on an independent research similar to MDC work, and see the benefit of the data for Addis Ababa, Ethiopia. Other researchers should use the data of Ethio-telecom for accurate result. In the future, more algorithms for prediction experiment and the results could be combined with classical statistical methods to obtain possibly more accurate results in shorter time. This work and the gained results can be used as a basis for Big Data processing in telecommunications using Neural Networks.

Chapter 6

Bibliography

[1] Z. Sun, "10 Bigs: Big Data and Its Ten Big Characteristics," *PNG Univ. of Tech.,* Bais 17010, January 2018.

[2] M. Jeevan. (2018, October 24). *10 Vs of Big Data,* [online]. Available: https://www.edvancer.in/10-vs-big-data/

[3] V. Ondryhal *et al.,* "Utilization of Selected Data Mining Methods for Communication Network Analysis," *Radio Engineering 20,* no. 2, 2011.

[4] Feng, Huifang *et al.,* "Location Prediction of Vehicles in VANETs Using A Kalman Filter," *Wireless Personal Communications,* pp. 1-17, 2014.

[5] F. Péter *et al.,* "Accurate mobility modeling and location prediction based on pattern analysis of handover series in mobile networks," *Mobile Information Systems 5,* no. 3, pp. 255-289, 2009.

[6] Z. Lei *et al.,* "Improving Location Prediction by Exploring Spatial-Temporal-Social Ties," *Mathematical Problems in Engineering,* 2014.

[7] B. Martins *et al.,* "Predicting future locations with hidden Markov models," *In Proceedings of the 2012 ACM Conference on Ubiquitous Computing,* pp. 911-918, ACM, 2012.

[8] R. Jain *et al.,* "Location prediction algorithms for mobile wireless systems," *In Wireless internet handbook,* pp. 245-263, CRC Press, Inc., 2003.

[9] S. Parija *et al.,* "Location Prediction of Mobility Management Using Soft Computing Techniques in Cellular Network," *International Journal of Computer Network and Information Security (IJCNIS) 5,* no. 6, 2013.

[10] F. Mériaux *et al,* "Predicting a user's next cell with supervised learning based on channel states," *In Signal Processing Advances in Wireless Communications (SPAWC), Institute of Electrical and Electronics Engineers (IEEE) 14th Workshop on*, pp. 36-40, 2013.

[11] M. Ozer *et al.,* "Location prediction of mobile phone users using apriori-based sequence mining with multiple support thresholds," 2014.

[12] H. Jin, "A location prediction algorithm for mobile communications using directional antennas," *International Journal of Distributed Sensor Networks (IJDSN),* 2013.

[13] S. Rosenbush and M. Totty. (2018, May 10). *The Evolution of Big Data in U.S. Corporations* [online]. Available: http://whatsthebigdata.com/2018/03/10/the-evolution-of-big-data-in-u-s-corporations/

[14] Statistica. (2018, May 10). *Volume of data created worldwide from 2005 to 2025 (in zetabytes)* [online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

[15] IDC company (2018, June 25). *Unlocking Big Data, Business Values, Drivers, and Challenges* [online]. Available: https://www.unlockingbigdata.com/industries.cfm.

[16] S. Haykin, "Neural Networks –A Comprehensive Foundation," McMaster Univ., Ontario Canada, Pearson Education Singapore Pte. Ltd., 1999.

[17] M. T. Koné. (2018, May 10). *The future of human evolution* [online]. Available: http://futurehumanevolution.com/artificial-intelligence-future-human-evolution/artificialneural-networks

[18] S. Akoush and A. Sameh, "Movement Prediction Using Bayesian Learning for Neural Networks," Systems and Networks Communications, Second International Conference, *Institute of Electrical and Electronics Engineers (IEEE),* 2007.

[19] J. Kudlacek, "Big Data Analytics for Mobile Networks," M.S. thesis, Dept. Elect. Eng., Czech Univ., Prague, 2015.

[20] I. Stojmenovic *et al.,* "Autoregression Models for Trust Management in Wireless Ad Hoc Networks," *Global Telecommunications Conference (GLOBECOM 2011), Institute of Electrical and Electronics Engineers (IEEE),* pp. 1-5, 2011.

[21] L. Cristian *et al.,* "Significant Location Detection and Prediction in Cellular Networks Using Artificial Neural Networks," *Computer Science and Information Technology (CSIT),* vol.03, no. 3, pp.81-89, 2015.

[22] H. Gao *et al.,* "Mobile location prediction in spatiotemporal context," *In the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing,* Nokia, June 2012.

[23] S.Rajagopal *et al.,* "Gps based predictive resource allocation in cellular networks," *In Networks 10th IEEE International Conference*, *Institute of Electrical and Electronics Engineers (IEEE),* vol. 10, pp. 229–234, 2002.

[24] M. Hudson *et al.,* "Neural Network Toolbox: *Getting Started Guide,*" The MathWorks, Inc. March 2016.

[25] Tawfiq M. and Eqhaar H., "On Multilayer Neural Networks And Its Application For Approximation Problem," *3rd scientific conference of the College of Science,* Univ. of Baghdad, March 2009.

[26] Jabber A. K., "On Training Feed Forward Neural Networks for Approximation Problem," MSc. Thesis, Univ. of Baghdad, College of Education ± Ibn Al-Haitham, 2009.

[27] H. Eqhaar, "Comparison Between Artificial Neural Networks with Radial Basis Function and Ridge Basis Function for Interpolation Problems," MSc. Thesis, Univ. of Baghdad, College of Education ± Ibn Al-Haitham, 2007.

[28] L. Naji and F. Faisal, "New Technique to Estimate the Concentration of Heavy Metals in Soil: On Contamination in Soil," June 2016.

[29] Nathan Eagle *et al.,* "Inferring Social Network Structure using Mobile Phone Data," *Proceedings of the National Academy of Sciences (PNAS),* Vol 106 (36), pp. 15274-15278, 2009.

[30] J. K. Laurila *et al.,* “The mobile data challenge: Big data for mobile computing research,” *In Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf. on Pervasive Computing*, Newcastle, UK, 2012.

[31] S. Srihari. (2019, April 15) *The Hessian Matrix Machine Learning* [online]. Available: http://www.cedar.buffalo.edu/∼ srihari/CSE574/Chap5/Chap5.4-Hessian.pdf

[32] M. T. Hagan *et al.,* “Training Recurrent Networks for Filtering and Control,” in (editors) L.R. Medsker, L.C. Jain, CRC Press, 2001.

[33] KDnuggets News. (2019, April 2). *Top 10 Data Science Use Cases in Telecom: Data Science, Telecom, Use Cases* [online]. Available: https://www.kdnuggets.com/2019/02/top-10-data-science-use-cases-telecom.html

[34] A. I. Galushkin, “The Structure of Neural Network,” *Neural Networks Theory,* ch. 2, pp. 33-40, 2007.

[35] A. K. Jain, “Artificial Neural Networks: A Tutorial,” *IBM Almanden Research Center, Institute of Electrical and Electronics Engineers (IEEE),* vol. 18, pp. 31–44, March 1996.

[36] A. Tsakonas *et al.,* “Towards a Comprehensible and Accurate Credit Management Model: Application of Four Computational Intelligence Methodologies,” Univ. of the Aegean, 2017.

[37] OpenCelliD. (2019, January 4). *The world’s largest Open Database of Cell Towers* [online]. Available: https://opencellid.org/

[38] G. M. Weiss, “Data Mining in Telecommunications,” in Data Mining and Knowledge Discovery Handbook, pp. 1189-1201, 2005.

[39] J. van der Lande, “How CSPs can can generate profits from their data,” Analysis Mason, 2013.

[40] J. Middleton. (2019, January 6) *Turkcell using Big Data to track faults* [Online]. Available: http://telecoms.com/173682/turkcell-using-big-datato-track-faults/

[41] R. I. Jony *et al.,* "Big data Use Case Domains for telecom operators," *IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom and SC, Institute of Electrical and Electronics Engineers (IEEE),* 2015.

[42] T. Sonera. (2019, January 9) *Spotify music on the computer, mobile, TV and almost anywhere!* [Online]. Available: http://www.teliasonera.com/en/innovation/entertainment/2011 /4/draft/

[43] H. Windsor. (2019, February 4) *Mobile Revenue Assurance & Fraud Management, Juniper Research* Online. Available: https://www.juniperresearch.com/press/pressreleases/mobile-industry-lost-over-$ 58-billion-in-revenue-i

[44] H. Demuth and M Beale, "Neural Network Toolbox User's Guide," *The Math Works Inc.,* Natick, MA, USA, pp. 5–22, 2018.

[45] H. Okut *et al.,* "Prediction of body mass index in mice using dense molecular markers and a regularized neural network," Genet. Res. Camb. Vol. 93, pp. 189–201 2011.

[46] F. Burden and D. Winkler, "Bayesian regularization of neural networks," *Methods Mol. Bio.* Vol. 458, pp. 25–44, 2008.

[47] D. Fumo. (2019, February 15) *Types of machine learning algorithms you should know* Online. Available: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

[48] Dorofki M. *et al.,* "Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data," *International Conference on Environment, Energy and Biotechnology IACSIT Press,* IPCBEE, Singapore, Vol. 33, pp. 39- 44, 2012.

[49] Tawfiq M., "Improving Gradient Descent method For Training Feed Forward Neural Networks," *International Journal of Modern Computer Science & Engineering,* Vol. 2, No. 1, pp. 1-25, 2013.

[50] Wang, X. G. *et al.,* "An improved back propagation algorithm to avoid the local minima problem," *Neurocomputing,* Vol. 56, pp. 455-460, 2017.

[51] H. Ababarnel *et al.,* "The analysis of observed chaotic data in physical systems," *Reviews of Modern Physics*, Vol. 65, No. 4, pp.1331-1392, 1993.

[52] N. Gershenfeld and A. Weigend, "In Time Series Prediction: Forecasting the Future and Understanding the Past," The Future of Time Series, pp 1-70. 1993.

[53] K. Nanda *et al.,* "Location Prediction of Mobility Management using Neural Network Techniques In Cellular Network," *International Conference on Embeded Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System, Institute of Electrical and Electronics Engineers (IEEE),* 2013.

[54] MathWorks. (2019, February 10) *How dynamic neural networks work* [online]. Available: http://www.mathworks.com/help/nnet/ug/how-dynamic-neural-networks-work.html.

[55] M. Moreira and E. Fiesler, "Neural Networks with Adaptive Learning Rate and Momentum Terms," *Institut Dalle Molle D'Intelligence Artificielle Perceptive,* no. 95-04, October 1995.

[56] F.S. Panchal1 and M. Panchal, "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network," *International Journal of Computer Science and Mobile Computing,* Vol.3 Issue.11, pp. 455-464, India, November 2014.

[57] Foresee F. Dan and Martin T. Hagan, "Gauss-Newton approximation to Bayesian learning," *Proceedings of the International Joint Conference on Neural Networks*, *Institute of Electrical and Electronics Engineers (IEEE),* Vol. 5, pp. 1930-1935, June 1997.

[58] A. Tesfay, "Neural Network based 3G Mobile Sites Fault Prediction: A Case Study in Addis Ababa, Ethiopia ," M.S. thesis, Dept. Elect. Eng., Addis Ababa Univ., Ethiopia, November 2018.

[59] K. Amit *et al.,* "Evolution of mobile wireless communication networks: 1G to 4G," *International Journal of Electronics and Communication Technology,* vol. 1, pp. 68–72, 2010.

[60] Y. Chen, “Soft handover issues in radio resource management for 3G WCDMA networks,” Ph.D. dissertation, Queen Mary, Univ. of London, 2017.

[61] A. Pashtan, “Wireless Terrestrial Communications: Cellular Telephony,” Buffalo Grove, Illinois, USA: Elose Publisher, September 2012.

[62] J. Govil, “4G: Functionalities development and an analysis of mobile wireless grid,” *in First International Conference on Emerging Trends in Engineering and Technology (ICETET),* pp. 270 –275, July 2008.

[63] D. Raychaudhuri and N. Mandayam, “Frontiers of wireless and mobile communications,” *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE),* vol. 100, no. 4, pp. 824 –840, April 2012.

[64] S.Parkvall *et al.*, “LTE-advanced evolving LTE towards IMT-advanced,” *in Institute of Electrical and Electronics Engineers (IEEE),* 68th Vehicular Technology Conference, pp. 1-5, 2008.

[65] V. Osa *et al.,* “Implementing opportunistic spectrum access in LTE-advanced,” *EURASIP Journal on Wireless Communications and Networking,* vol. 2, pp. 1–17, 2012.

[66] A. Cupper *et al.,* “TraX: a device-centric middleware framework for location-based services,” *Communications Magazine, Institute of Electrical and Electronics Engineers (IEEE),* vol. 44, no. 9, pp. 114-120, September 2006.

[67] M. Moustafa *et al.,* “Game based dynamic resource scheduling in QoS aware radio access networks,” Soft Comput., vol. 9, no. 2, pp. 101–115, February 2005.

[68] M. I. Wardlaw, “Intelligence and mobility for BT’s next generation networks,” BT Technology Journal, vol. 23, no. 1, pp. 28–47, 2005.

[69] M. Mackaya *et al.,* “Modelling location operations in UMTS networks,” *in Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems,* ser. MSWiM’02. New York, USA: ACM, pp. 69–73, 2002.

[70] IAMAI-Indicus, “White paper : Location based services (LBS) on mobile in India,” *INDICUS,* vol. 14, pp. 5 –8, 14 April 2008.

[71] Lopa J. Vora, "Evolution of Mobile Generation Technology: 1G to 5G and review of upcoming wireless technology 5G," *International Journal of Modern Trends in Engineering and Research (IJMTER),* vol. 02, pp. 281–290, October 2015.

[72] A. Kumar *et al.,* "5G Technology – Redefining wireless Communication in upcoming years," *International Journal of Computer Science and Management Research (IJCSMR),* Vol 1 Issue 1, August 2012.

[73] S. Venkatachalaiah, *et al.*, "Improvement of Handoff in Wireless Networks using Mobility Prediction and Multicasting Techniques," *Centre for Advanced Technology in Telecommunications (CATT),* vol. 4, no. 2, pp. 195-208, 2005.

[74] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *Institute of Electrical and Electronics Engineers (IEEE), Transactions on Networking,* vol. 11, no. 5, pp. 718-732, 2003.

[75] D. Barbar, "Mobile computing and databases-a survey," *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Knowledge and Data Engineering,* vol. 11, pp. 108–117, 1999.

[76] S. H. Shah and K. Nahrstedt, "Predictive location-based QoS routing in mobile Ad-Hoc networks," in *Institute of Electrical and Electronics Engineers (IEEE), International Conference on Communications,* vol. 2, New York, USA, pp. 1022–1027, May 2002.

[77] S. Lu, "Applied Neural network for location prediction and resources reservation scheme in wireless networks," *in International Conference on Communication Technology Proceedings (ICCT) Institute of Electrical and Electronics Engineers (IEEE),* vol. 2, 9-11, pp. 958 – 961, April 2003.

[78] J. Capka and R. Boutaba, "Mobility Prediction in Wireless Networks Using Neural Networks," I. I. F. for Information Proceeding, Ed. Springer Heidelberg Berlin, vol. 3271, 2004.

[79] M. Ren and H. A. Karimi, "A chain-code-based map matching algorithm for wheelchair navigation," *Transactions in Geograhpical Information System (GIS),* vol. 13, no. 2, pp. 197–214, 2009.

[80] M. Ren and H. A. Karimi, "A fuzzy logic map matching for wheelchair navigation," *GPS Solutions,* vol. 15, pp. 1–10, 12 June 2011.

Appendices

Appendix A

A.1 Sample Raw Data

UMTS	636	1	1601	6977357	0	39.777033	10.060619	18613	12	1	1459751568	1463394399
UMTS	636	1	1702	7307199	0	37.367935	11.646194	1000	1	1	1459751568	1454722314
UMTS	636	1	1105	6912218	0	38.83873	9.0218353	1000	1	1	1459751568	1456714860
UMTS	636	1	1105	6912202	0	38.840356	9.013042	1174	8	1	1459751569	1519293153

A.2 Snapshot of Training Result

Month	Data category	No of data	Training data	Validation data	Test data	Num. of Hidden Neurons	Num. of delays	Training MSE	Validation MSE	Test MSE	Training R	Validation R	Test R	Regression	Number of Iteration
Feb. 1, 2019	Week 1 day 1	1000	700	150	150	10	2	8.04816e-1	1.18130e-0	7.66523e-1	8.54453e-1	7.64245e-1	8.50742e-1	0.71*t+2.8	9
Feb. 2, 2019	Day 2	1000	700	150	150	15	4	3.46369e-1	2.02898e-1	2.33431e-1	5.54543e-1	2.32770e-1	4.15132e-1	0.29*t+6.5	9
Feb. 3, 2019	Day 3	1000	700	150	150	11	3	4.22795e-1	4.67778e-1	4.17022e-1	8.69188e-1	8.37871e-1	8.47570e-1	0.77*t+9.1	15
Feb. 4, 2019	Day 4	1000	700	150	150	12	4	0.869261	0.88133	0.879373	0.802453	0.803626	0.769185	0.61*t+15	11
Feb. 5, 2019	Day 5	1000	700	150	150	13	4	0.0480497	0.0746423	0.561732	0.664406	0.349812	0.242465	0.44*t+22	11
Feb. 6, 2019	Day 6	1000	700	150	150	13	4	0.199304	0.349842	0.315649	0.911213	0.853498	0.864954	0.78*t+8.8	9
Feb. 7, 2019	Day 7	800	560	120	120	13	3	0.56312	0.604634	0.363291	0.928507	0.912140	0.955481	0.83*t+6.7	13
Feb. 8, 2019	Day 8	14000	9800	2100	2100	14	4	0.716255	0.662096	0.721394	0.783902	0.782476	0.786466	0.59*t+16	12
Feb. 9, 2019	Day 9	14000	9800	2100	2100	14	4	0.60032	0.668682	0.712565	0.802273	0.800302	0.768669	0.64*t+14	13
Feb. 10, 2019	Day 10	14000	9800	2100	2100	15	4	0.704750	0.665639	0.635322	0.788991	0.799520	0.787428	0.62*t+15	12

A.3 Snapshot of Training Model