CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

MASTER’S THESIS

2015

Bc. Jakub Kudláček

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

Department of Electromagnetic Field

Big data analytics for mobile networks

May 2015

Author:	Bc. Jakub Kudláček
Supervisor:	Doc. Ing. Zdeněk Bečvář, Ph.D.

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Electromagnetic Field

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Jakub Kudláček**

Study programme: Communications, Multimedia, Electronics
Specialisation: Wireless Communication

Title of Diploma Thesis: **Big data analytics for mobile networks**

Guidelines:

Study problem of big data for various use-cases and analyze possible exploitation of big data for mobile networks. Then, investigate methods suitable for prediction of big data obtained from mobile networks. Implement and assess selected methods for big data analytics using appropriate methodology and tool.

Bibliography/Sources:

[1] S. Parija, R.K. Ranjan, P.K. Sahu, "Location Prediction Of Mobility Management Using Neural Network Techniques In Cellular Network," International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System, 2013.

[2] Z. Zheng, J. Zhu, and M.R. Lyu, "Service-generated Big Data and Big Data-as-a-Service: An Overview," IEEE International Congress on Big Data, 2013.

[3] D. Agrawal et al., "Challanges and Opportunities with Big Data," A community white paper developed by leading researchers across the United States, 2012.

Diploma Thesis Supervisor: doc. Zdeněk Bečvář Ing., Ph.D.

Valid until: SS 2015/2016

prof. Ing. Pavel Pechač, Ph.D.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 26, 2015

I hereby declare that this master’s thesis is completely my own work and that I used only the cited sources in accordance with the Methodical instruction about observance of ethical principles of preparation of university final projects.

Prague, May 5, 2015

………………..……………………
Signature

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 5. května 2015

………………..……………………

podpis

Acknowledgements

Specially, I would like to thank the thesis advisor Doc. Ing. Zdeněk Bečvář, PhD. for his support, guidance, comments which contributed to the improvement of this work and last but not least for his continuous interest in my work.

Then I would like to thank the IBM solution architect Ing. Jiří Slabý, PhD. for his supportive comments and all people who contributed to my work whether much or little.

Finally, I would like to give a special thanks to my parents. Without them I would not be here at all and I would never reach up to here without their help and understanding.

Abstrakt

V této práci je představen obecný pohled na analýzu velkého objemu dat, a to především v telekomunikacích. Práce se zaměřuje na zpracování dat pro predikci pohybu mobilního uživatele v mobilní síti. Cílem práce je zpracovat a zanalyzovat získaný vzorek telekomunikačních dat pomocí neuronové sítě a poskytnout co možná nejpřesnější predikci pohybu mobilního uživatele. Získaný výsledek z predikce je dále optimalizován pomocí iterační metody vyhledávání nejvhodnější možné kombinace parametrů neuronové sítě. Efektivita predikce pohybu uživatele je ověřena simulací v prostředí MATLAB. Výsledky simulace ukazují úspěšnost predikce až 97%, což je přesnost dostatečná pro široké využití predikce pro optimalizaci mobilních sítí nebo pro služby spojené s predikcí pohybu uživatele. Získané výsledky plně reflektují reálné řešení pro telekomunikační průmysl a mohou pomoci při plánování aktivit spojených s pohybem mobilního uživatele v dané lokalitě.

Klíčová slova

Velká data, telekomunikace, komerční subjekty, predikce, neuronová síť, konfigurace, trénink, optimalizace, mobilní sítě

Abstract

This work introduces a general view on analysis of big data, especially in telecommunications. The work is focused on analytics of data for mobile users movement prediction in telecommunications network. The objective of this work is to process and analyze obtained samples of telecommunications data by means of neural network and provide as accurate mobile users movement prediction as possible. Obtained results from prediction are then 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 MATLAB. Simulation results show success rate of prediction up to 97%, which is 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.

Key words

Big data, telecommunications, commercial subjects, prediction, neural network, configuration, training, optimization, mobile networks

"I have not failed. I've just found 10,000 ways that won't work."

Thomas A. Edison

CONTENTS

List of Figures

List of Tables

List of Acronyms

SQL	Structured Query Language
NoSQL	Not only SQL
IT	Information Technology
ETL	Extract, Transform and Load
HDFS	Hadoop Distributed File System
OLTP	Online Transaction Processing
OLAP	Online Analytical Processing
BI	Business Intelligence
GPFS	General Parallel File System
MDM	Master Data Management
SIMD	Single Instruction Multiple Data
CSP	Communication Service Provider
CDR	Call Detail record
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
QUEST	Quaternion Estimator
FACT	Fuzzy art with Add Clustering Technique
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
PDA	Pitch Detection Algorithm
SOM	Self-Organizing Map
BPG	Back-propagation algorithm
NARX	Nonlinear Autoregressive with External Input
BTS	Base Transceiver Station
QoS	Quality of Service

1 INTRODUCTION

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 have already been thought and realized by many companies. They copied the right data into their warehouses and selected the right strategy how to analyze and display the data. Every piece of the data is organized and categorized in data warehouses [1]. The problem is that recently the companies' ability to generate the data exceeds the ability to store, process or analyze it.

There are a lot of sources that can generate a huge amount of data. For business value, it is very important to understand the power that is hidden in modern data sources like sensors, mobile phones, emails, videos, social networks, etc. [2, 3]. For business, the goal is to store, process, visualize and analyze the data in a new way. All the mentioned steps are not new for information technologies but a problem arises if data is massive or complex or just is coming to us so fast that it is very complicated or almost impossible to work with it using classical old relational databases and techniques. Arrival of concept of NoSQL databases [4, 5] makes working with big data more efficient and easier. Big data can be described by three main characteristics, denoted as 3V [6]: Volume, Velocity and Variety, see Fig. 1.

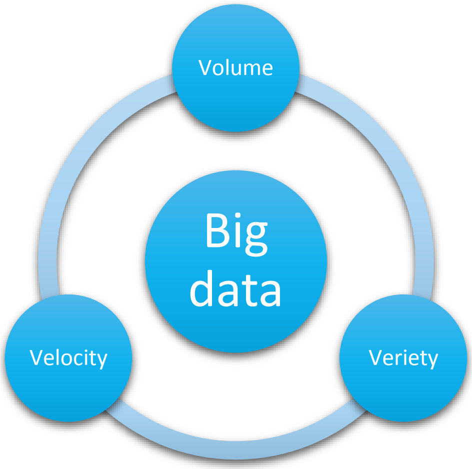

Fig. 1: Big data summary

The volume means how much data is coming to Information Technology (IT) systems. The input can differ from company to company and recently it is not a special situation that some companies generate terabytes of data and in many cases more than terabytes, petabytes. As the amount of data is soaring up day by day it can be expected that companies will face this issue more often than before [7, 8]. For example, every month Facebook generates more than 955 million new accounts in 70 languages, 125 friends set for new friendships and 140 billion updated photos and uploaded videos in total length of 48 hours [7].

The variety is actually the reason why the inputs of data are so massive. In general data can be structured, unstructured or semi-structured. In terms of structured data all the process steps are relatively simple because of tags and structured hierarchy. On the other side the unstructured data is without tags and this data is saved with no organization at all. Therefore more sophisticated and complex systems have to be used to analyze this data. Finally the semi-structured data have no logical hierarchy but it contains tags so that the particular elements can be found [2, 9].

The velocity is understood as the speed of processing and data generation. The processing of the data as fast as possible is needed to get information immediately so that companies can keep up with data generation speed and react according to the data from their clients or devices [7, 8].

Some other sources speak about different characterization of big data to 6 parts: variety, volume, velocity, variability, complexity, value [6]. The variability can be understood as a data flow that is changing over time. The complexity is defined by the amount of data sources. Generally, the more sources the more complicated data processing. The last part, the value, can be described as the ability to filter the data and make right business decisions on the basis of the filtered data.

This thesis is focused on exploitation of big data analytics for prediction of the user's movement in mobile networks. Results obtained from prediction can bring significant improvements in the field telecommunications industry in terms of marketing activities, customer quality of service and infrastructure planning. The contribution of this thesis consists in:

- Well-arranged definition of big data concept for telecommunications and provision of a deeper insight to big data industry and customer’s behavior in network and quality of service.
- Analysis of various types of algorithms for prediction of user's movement exploiting processing of big data.
- Prediction of mobile user's movement using a 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.

The rest of this work is organized as follows. In the next three sections, there is a definition of big data and summary of main differences of both approaches. In section 5, market customer needs are presented. Summary of hot topic use case for telecommunications is presented in section 6. Next section provides a view into statistical methods including neural network theory. Scenario definition and final results are commented in section 8. Final section sums up major conclusions and possible future extension of this work.

2 BIG DATA OVERVIEW

This chapter provides a brief summary of big data history and building on that, the Hadoop concept is presented with its main parts: Hadoop Distributed File System (HDFS), MapReduce and auxiliary Hadoop parts.

2.1 GLIMPSE OF HISTORY

The term big data is known for many years. At the beginning of that era companies took notice of increasing amount of data. They started thinking about what they should have done with that data, that volume. The first applications were developed around 1980's and the first attempts to process such volume were carried out at the same time. About the year 2000 Google came up with a totally new idea how to process, store and manage a lot of information. Tab 1 with the largest corporate data since 1950's until 2010's demonstrates how fast the data are growing [10]. The same trend is observed in Fig. 2, which shows exponential increase in the amount of data over time. There is an obvious exponential increase in data generated from companies' internal systems.

	Company	Industry	Data
1950's	John Hancock Mutual Life Insurance Co.	Insurance	600 Megabytes
1960's	American Airlines	Aviation	800 Megabytes
1970's	Federal Express Cosmos	Logistics	80 Gigabytes
1980's	CitiCorp's NAIB	Banking	450 Gigabytes
1990's	Walmart	Retail	180 Terabytes
2000's	Google	IT	25 Petabytes
2010‘s	Facebook	Internet information provider	100 Petabytes

Tab. 1: The largest corporate data since 1950's until 2010‘s

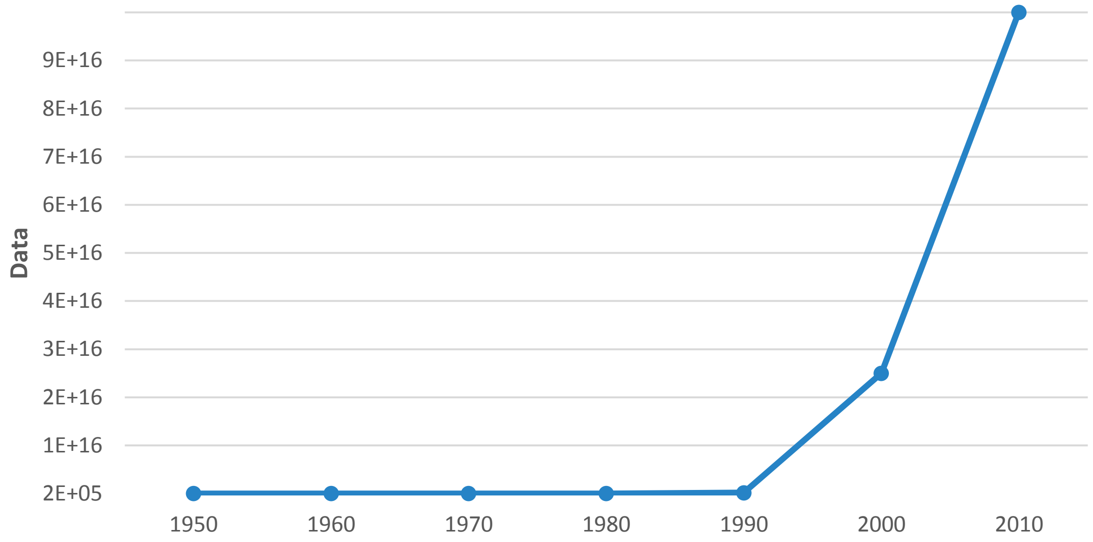

Fig. 2: A timeline of big data in year's decade

In 2000's Google was the first who started to generate a huge amount of data from the internet. Google was the first company which started to use the term big data. They created a totally new concept of working with big sets of data, denoted as BigTable [11]. This concept can be divided into three parts: Google's data storage system, Google File System and MapReduce. 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. It is not the objective of this work to mention the whole concept but to describe essential parts for data processing framework called Hadoop in order to understand sufficiently the resulting analytics parts.

2.2 HADOOP

Hadoop was inspired by Google's data storage system and Google File System. It is an open source platform which is based on Java framework. It is a framework which uses a simple programming model and is able to connect thousands of computers to store a massive amount of data. It was neither designed for real time based systems like stream nor as a system that should supersede warehouses, databases or ETL (Extract, Transform and Load). Hadoop can consist of many parts but two of them create the system core as seen on Fig. 3, Hadoop distributed file system (HDFS) and Google's MapReduce [11, 12]. Others parts are briefly mentioned in the last paragraph.

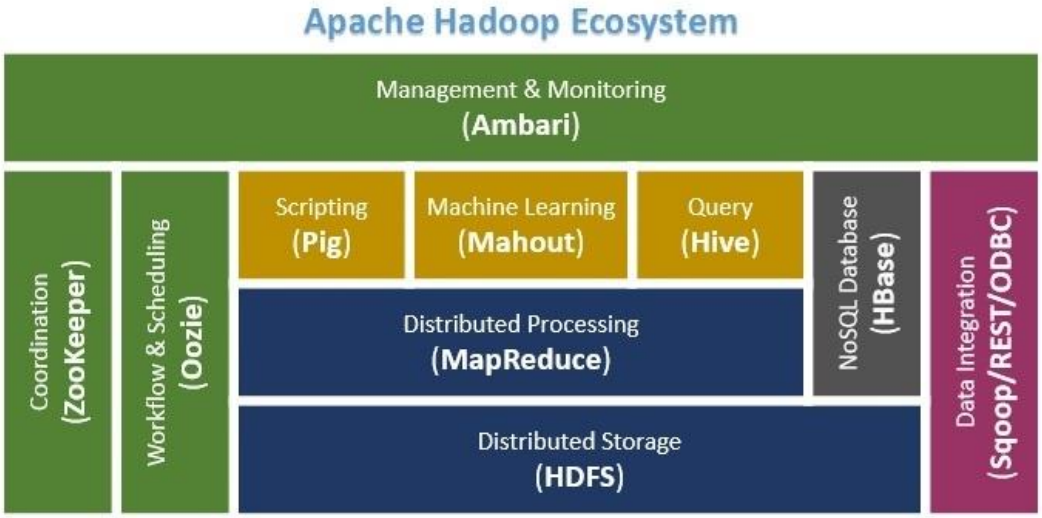

Fig. 3: Hadoop parts [16]

Hadoop Distributed File System (HDFS) is able to connect a thousand of servers, create one big cluster and store a huge amount of data. These servers communicate and use the MapReduce framework to simplify problems. They cut big set of data into smaller chunks that are afterwards automatically processed by different nodes in the Hadoop cluster in a minimum period of time [2, 8, 13].

Processing of big data are generally very complex and complicated tasks. That is why Google developed in 2004 a programming framework for distributed computing which is called MapReduce. This framework is designed to work on clusters of computers so that it could manage real massive data problems by dividing them into smaller units which can be processed more quickly in parallel. MapReduce process can be divided into two steps [14, 15]:

1. Map Step: As the first step master nodes come into play. They take a massive set of data and divide it into smaller chunks. In the second step there are worker nodes. A worker node can either do all the division process again and create a multi-level tree structure or process the divided set of data and send it back to its master node as it can be seen on Fig. 4.

2. Reduce Step: In this step there is a tree hierarchy created from the Map Step. All the smaller chunks are grouped back to the master node. Then they are processed in parallel by the reduce algorithm to a set of value in the same domain as shown on Fig. 4.

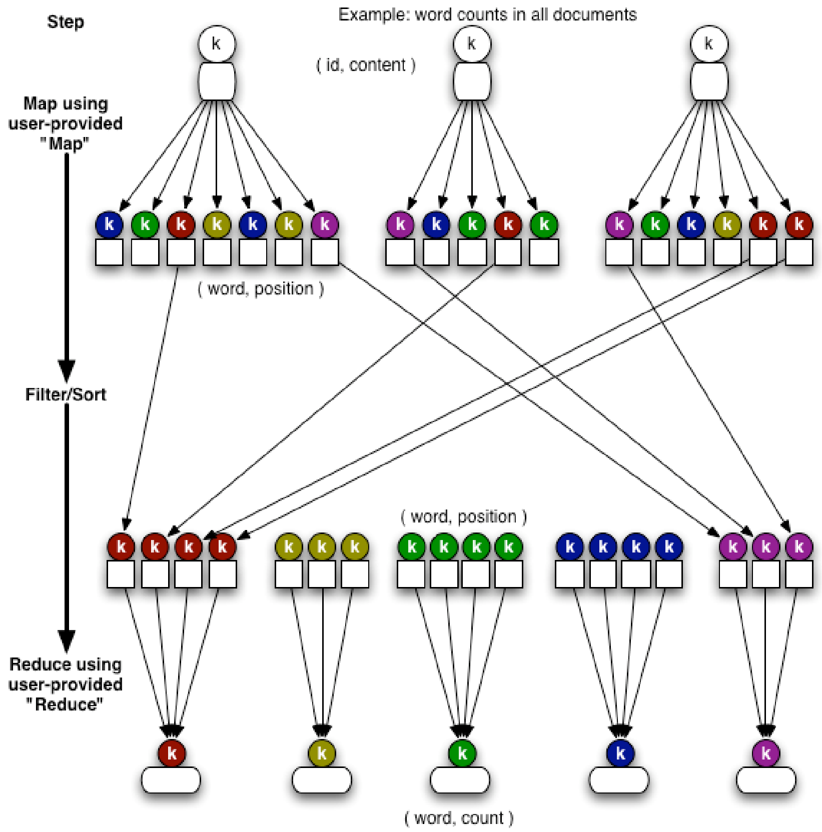
Step
Example: word counts in all documents
k
(id, content)
Map using user-provided "Map"
(word, position)
Filter/Sort
(word, position)
Reduce using user-provided "Reduce"
(word, count)

Fig. 4: MapReduce step process [17]

There are more auxiliary parts that could be meant but only the ones from Fig. 3 are discussed. For more information see [4]. These parts complement the Hadoop ecosystem and provide supportive functions in overall big data processing.

- Sqoop/Flume: Sqoop is an application that allows to transfer data between Hadoop and relational databases. The whole set of data is divided into different partitions and then transferred.
- Pig: a procedural data processing language. Instead of writing logic-driven queries, Pig works with a series of steps like an everyday scripting language. It solves common data processing issues.
- Hive: an application (data warehouse solution) that let users use a SQL like interface and relational model while working with Hadoop.
- Oozie: a very complex application that allows users to define, control and describe job flows.
- Mahaout: an open source library and machine learning tool that is used when the set of data is too large for one machine.
- Zookeeper: an open-source coordination service that controls overall maintenance and synchronization.
  HBase: a key-value (NoSQL) data store. It behaves just like any other database and it does not support SQL. It provides more complex querying and processing through MapReduce with the ability to be connected with Hive.

3 BIG DATA IN INDUSTRY

There are a lot of options how to solve issues regarding big data on today's market. Companies try to extend their product portfolios to satisfy all customer needs. With the arrival of cloud combining with analytics and big data concepts there are many new ideas how to process data. New strategies are formed to be connected with current relational databases at the data warehouse's core. The biggest players in data management systems are shown on Fig. 5.

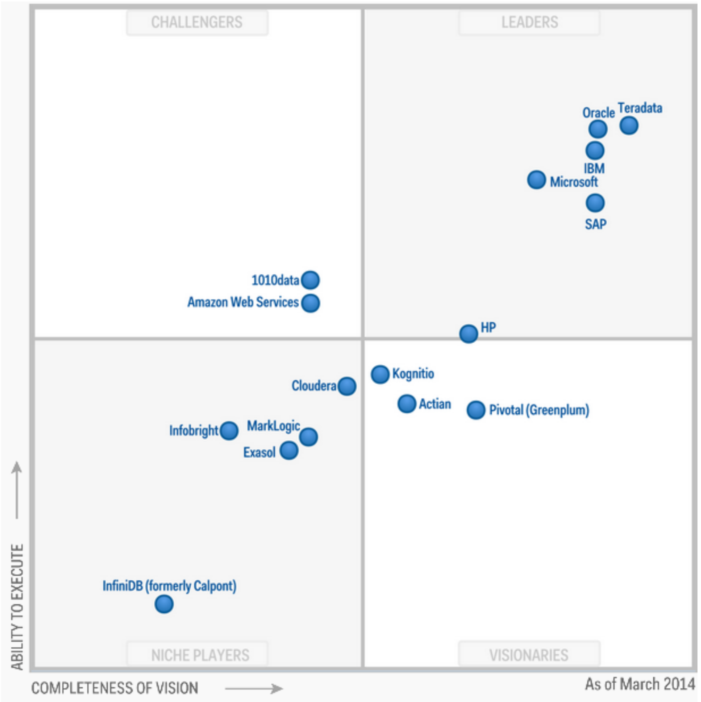

Fig. 5: Big data biggest players in data management systems on the market [18]

According to Gartner's Magic Quadrant for Data Warehouse Database Management Systems released in March 2014 [18] can be seen the biggest players on the field of big data. The market leaders displayed on Fig. 5 positioned at the top right corner are discussed with special emphasis on IBM key differentiators in the next chapter.

3.1 TERADATA

Teradata [19] is a big player on the market in terms of big data. They define four new different phenomenona on the market. First phenomenon is that the amount of data is soaring and therefore companies have to think how to store it economically and how long should it stay on servers. The next one describes how to connect new big data concept with non-relational databases in order to get better insights. The third one is about processes that have changed because clients need more sophisticated insights from their data. The last different phenomenon about big data is that it is not enough to have just one process engine. It is necessary to develop more powerful systems of connected platforms.

Architecture

An architecture of the IT system for data processing shows a general concept of how particular IT system parts communicate and work with each other. Teradata's analytics architecture, shown in Fig. 6, contains Hadoop as a core part of the infrastructure for big data analytics. The Teradata system further includes following parts:

- Teradata Aster Discovery Platform – is presented as an appliance Hadoop combined with analytics layer and Business Intelligence & Planning analytics tools [21]
- Teradata Integrated Data Warehouse – is a relational data warehouse appliance based on OLTP, which is able to store data and analyze it very fast by using parallel processing methods [20]
- Hadoop – Teradata has Hadoop environment from Hortonworks.

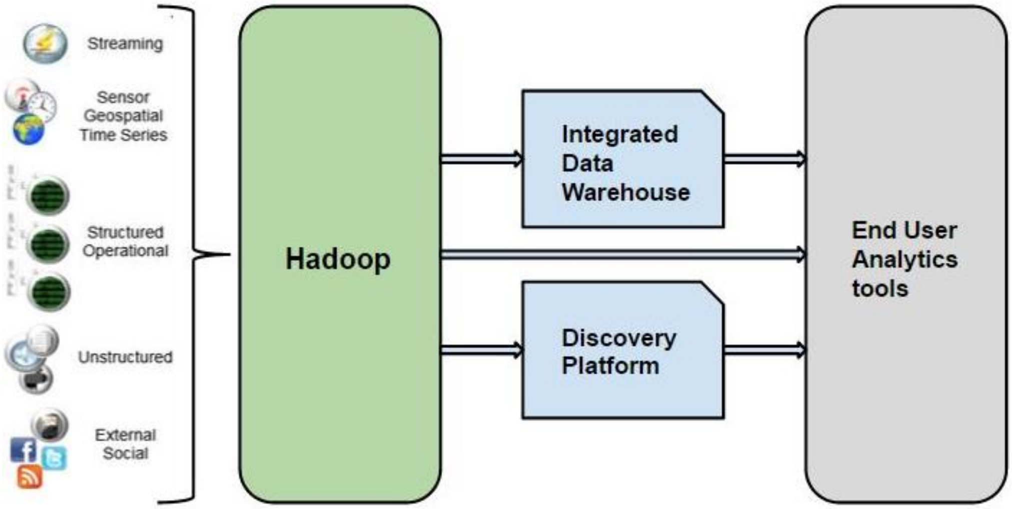

Fig. 6: Teradata big data architecture

3.2 ORACLE

According to a survey [22] which Oracle did with Economist intelligent units, just 12 per cent of current companies understand and can imagine the impact of big data over the next 3 years. From Oracle point of view, big data can bring many important benefits into all companies. For example, technical data, business data and more sources can be used and analyzed together. Companies will be able to predict more situations and response more quickly to all their issues that are business critical. Also, they will be able to look back at the old data to compare if there are some differences or not from the new data.

Architecture [23]

An architecture of the IT system for data processing shows a general concept of how particular IT system parts communicate and work with each other. This chapter presents Oracle's analytics architecture on Fig. 7 where Hadoop, here called Big Data Appliance, is a core part of the infrastructure when speaking about big data analytics. The Oracle system further includes following parts:

- Big data Appliance – is a high performance platform that runs many servers, combines NoSQL and HDFS
- Exadata – is a special oracle database appliance that can behave like OLTP DB on one hand and OLAP on the other hand
- Exalytics (appliance in memory for Business Intelligence & Planning analytics tools) – is an appliance that provide an extreme in-memory Business Intelligence & Planning analytics performance. This technology pumps information from Exadata.

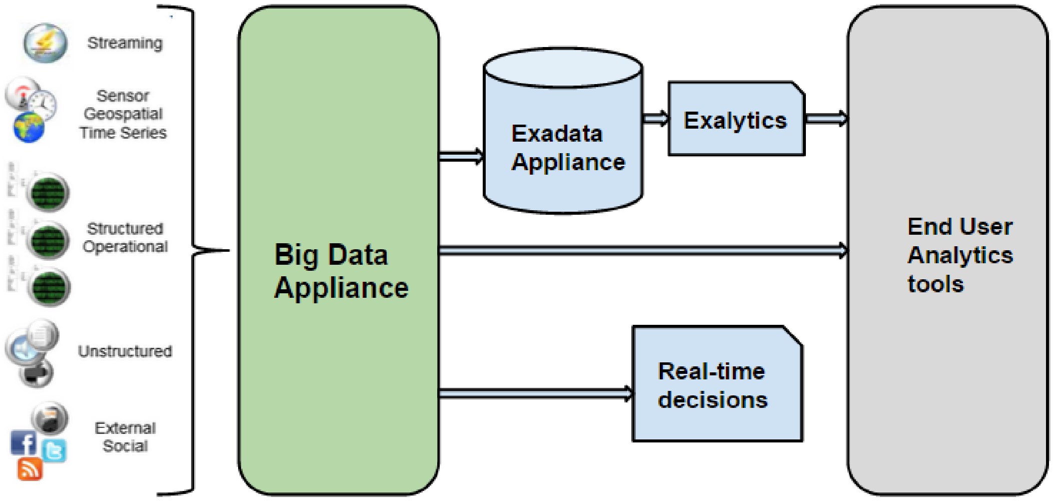

Fig. 7: Oracle big data architecture

3.3 SAP

For SAP using big data is a new way of working, playing and living. [24] It is the way how to be always well-informed by capturing and analyzing all the signals within digital noise. Big data can become involved in almost every aspect of living from shopping and sensors to medical treatment and nature renewal.

According to SAP big data has a huge future in helping people to have better life and mainly for those in needs [25] whether we speak about diseases or the poor without water resources. Nowadays a lot of people and their families suffer from various diseases. SAP has decided to use their infrastructure and software to store all the data from different medical sensors and get better analyses that can save human life.

Architecture

An architecture of the IT system for data processing shows a general concept of how particular IT system parts communicate and work with each other. This chapter presents SAP's analytics architecture on Fig. 8 where Hadoop is a core part of the infrastructure when speaking about big data analytics. The SAP system further includes following parts:

- Hadoop – SAP does not use Hadoop as a standalone analytical engine but as a transformation tool to process unstructured data for SAP HANA.
- SAP HANA – is presented by SAP as an appliance but it can be also a software. HANA stands in the middle of SAP big data platform. It provides in-memory processing for structured data only.
- SAP BW – is a relational database software that sits on one of certified databases for SAP.
- Sybase IQ – is a server or platform that is used as an additional database to SAP HANA which work with MapReduce programming engine.

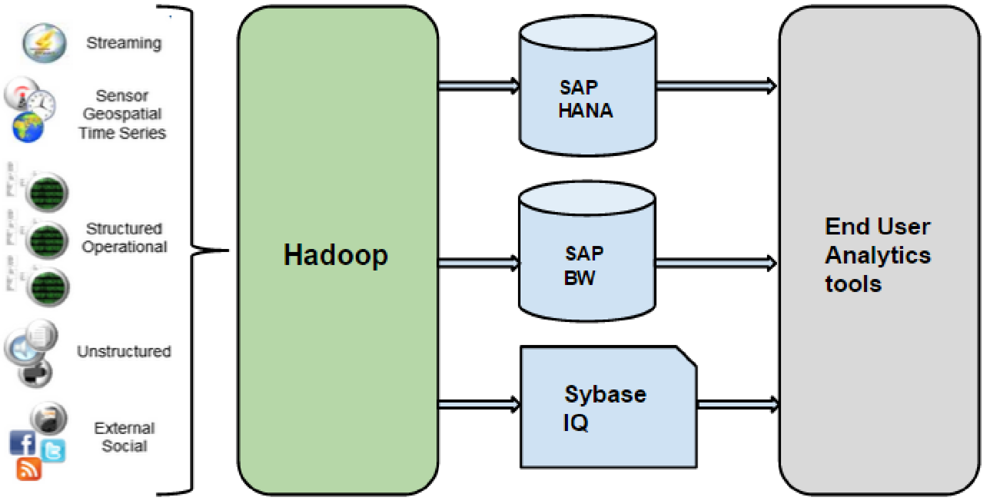

Fig. 8: SAP big data architecture

3.4 MICROSOFT

Microsoft believes there will be approximately 5.2 gigabytes per person data by 2020. Look at this amount in different unit of measurement, Microsoft says: [26] "there are twice as many bytes of data in the world than liters of water in our oceans." Microsoft sees a huge opportunity in Microsoft Office tools. These tools are widely spread all around the world and a lot of people are used to working with it. Microsoft also points out the importance of big data in telecommunications as still more and more devices are connected to networks. It will create a tremendous volume of data and also other hot topics will pop up like security, master data management, analytics and so on. Microsoft speaks about $235 billion to be at stake [27].

Architecture

An architecture of the IT system for data processing shows a general concept of how particular IT system parts communicate and work with each other. This chapter presents Microsoft's analytics architecture on Fig. 9 where Hadoop on Windows Server or on Windows Azure cloud platform, is a core part of the infrastructure when speaking about big data analytics. The Microsoft system further includes following parts:

- Hadoop On Windows Server – Microsoft cooperates with Hortonworks and uses Hortonworks data platform (HDP) in connection with Windows Server which is known as HDInsight Server for Windows. [28]
- Hadoop On Windows Azure – is a SaaS implementation of HDP called HDInsight Service [28]
- Microsoft EDW – is a classical data warehouse solution created for querying data prepared for subsequent analysis.
- SSRS - SQL Server Reporting Services [29] is a server-based reporting platform. It let SSRS users to analyze wide variety of data, create and manage reports.
- SSAS - SQL Server Analytical Services [30] is an online data engine that supports business decision making and BI tools. It can be connected with tabular models, OLAP cubes and Microsoft Office tools.

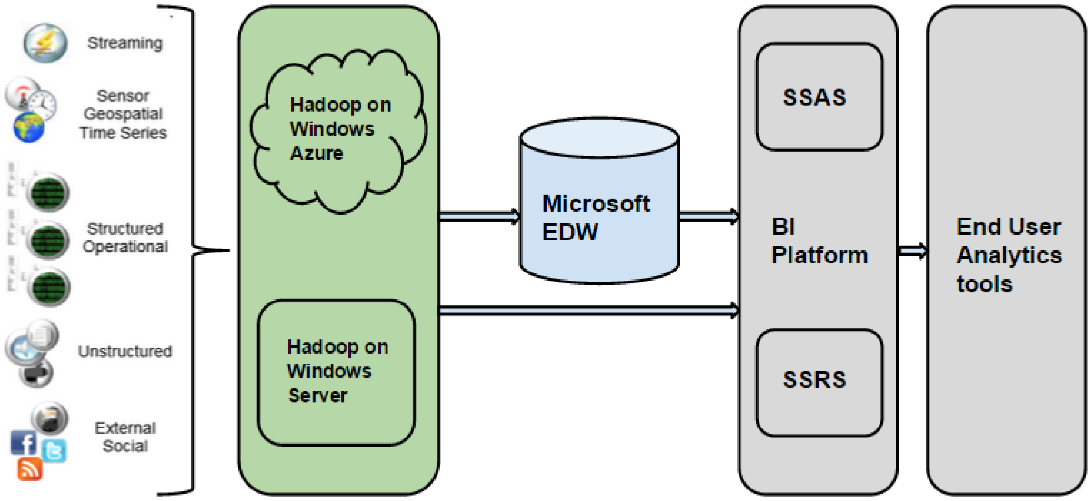

Fig. 9: Microsoft big data architecture

3.5 IBM

IBM sees the biggest potential in harvesting all sources together and combining them with people knowledge and experience in given industries. According to IBM, creating experience and knowledge patterns can lead to an absolutely new way of collective thinking and subsequently to make better decisions. In addition, cognitive analytics, connection of analytics and human experience, is a huge and very hot topic for IBM and can bring totally new insights that cannot be observe by classical approaches. Moreover, all that can be shared and spread by social networks in order to have fresh and clear information at the right time and the right place. IBM tries to help making nimble and critical decisions in healthcare, transportation, sport and many others at places where every second can play a significant role.

Architecture description

An architecture of the IT system for data processing shows a general concept of how particular IT system parts communicate and work with each other. This chapter presents IBM's analytics architecture on Fig. 10 where Hadoop, here called InfoSphere BigInsights, is a core part of the infrastructure when speaking about big data analytics. The IBM system further includes following parts:

- IBM InfoSphere BigInsights [31] – it is an IBM version of Apache Hadoop. This version brings more than just Hadoop. Besides many advantages, it includes unique GPFS storage and better data governance, embedded analytics and complex SQL access (Big SQL)
- IBM PureData for Operational Analytics [32] – it is an integrated data system (IBM enterprise warehouse) and complex solution created for high-performance operational analytics workloads. It can process 1000+ concurrent operational queries at a glimpse.
- IBM PureData for analytics [33] – this solution is focused on very fast analytical processing of huge amount of data. It is powered by IBM Netezza. It is an embedded and purpose-build analytics platform. It connects benefits of data warehousing and in-database solutions together to get extremely high-performance massively parallel platform.
- IBM DB2 BLU [34] – it is a special database that offers in-memory columnar processing along with multi-core and single instruction multiple data (SIMD) parallelism and unique compression technique. It is especially powerful on IBM Power hardware.

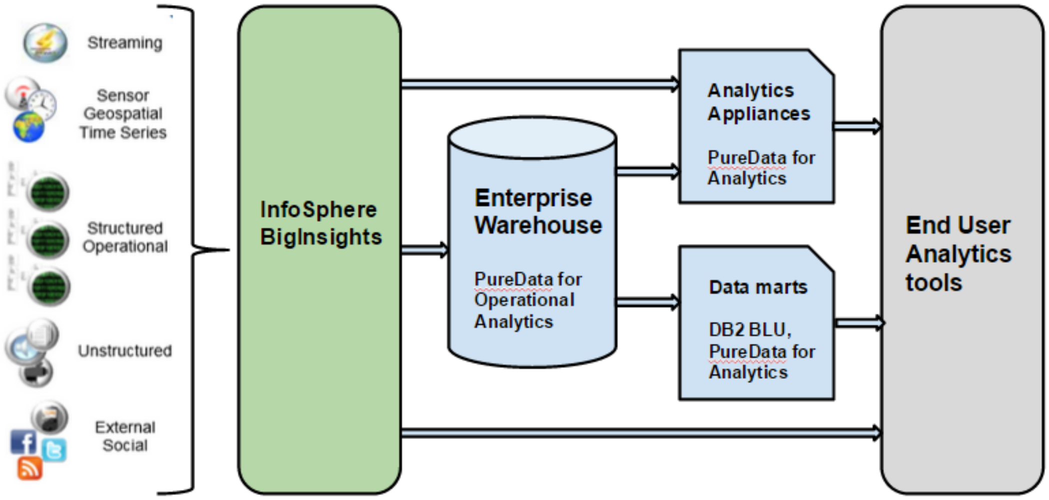

Fig. 10: IBM big data architecture

4 BIG DATA USE CASES FOR TELECOMMUNICATIONS

There are a lot of industries all around the world and each industry has its specific requirements according to what is the target of the business [35]. Some are focused on customer acquisition while others are more interested in creating as catchy advertising as possible to engage attention of their customers. Different industries, such as, healthcare industry, power industry, transportation industry, Governance industry, financial industry, media industry, or retail industry generate an enormous amount of data every single second.

Nowadays almost all people have their own mobile phone. These devices provide communication service providers (CSPs) with a lot of information about their customers. This information can be turned into a valuable insight by using appropriate big data tools. Today it is not only about mobile devices but also smart phones, tablets, sensors and TVs come to play. 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 amount of characteristics. One way is to split up use cases into 4 parts: marketing, customer experience, exogenous influencers and technical use cases. All the cases have one 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.

4.1 MARKETING USE CASES

This chapter is focused on marketing use cases derived from real industry problems on the current market. The selected use cases in this chapter are: Market & channel monitoring, Micro-segmentation, Reaction to an event.

Market & channel Monitoring

Marketing as it is generally known has evolved a lot. Today, the biggest emphasis is focused on measuring market "mood" and having product feedbacks across multiple channels. The goal is to get an overall view of the local market so that all events and promotions can be well-targeted. Most information comes from the Internet and call centers and is mainly processed ex post (not in real time). These information is valuable for people from the marketing department and salesmen.

Micro-segmentation

Segmentation is all about having the right data at the right time and as today there are a lot of data sources companies can do deeper analyses and insights. For example, it is possible to target on people from a given city that are under 25 and use smart phones with monthly payments lower than 25$ and are very likely to skip to another CSP in a short time. Marketers and salesmen can get all these information from the Internet, network probes, CDR (Call Detail Record) and CRM applications in ex post mode.

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 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, SMS fail delivery and others. It is used primarily by technician and business strategists. These data is generally analyzed in real time.

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. The best example could be a shopping mall. There are a lot of small shops. According to customer's actual geolocation position CSPs can send him a message with sales and promotions based on positions of shops. The customer is then informed about actual events in his near location. These real time analyses are appropriate mainly for marketers and business strategy planners.

4.2 CUSTOMER EXPERIENCE USE CASES

This chapter is focused on customer experience use cases derived from real industry problems on the current market. The selected use cases in this chapter are: Customer usage preference, Customer acquisition, Customer churn prevention.

Customer usage preference

To provide customers with the best service possible CSPs should know what their customers really want, how they use their mobile phones and other devices, how often, how much data they need, etc… These type of information obtained from network probes can help business strategists and salesmen create the next best offer solutions for customers.

Customer acquisition

The telecommunications environment is very dynamically changing and customers' needs are changing as well. That is the reason why it is necessary to keep acquiring new customers. Subsequently, new customers can very likely lead to revenue growth. This use case builds primarily on marketing and customer experience use cases. The both support CSPs brand and customer mind to change a service provider. Opinion makers and business strategists appreciate an overall insight to new customers. Main data stream should be from the Internet and CDR.

Customer churn prevention

This is a very hot topic for CSPs as on the local market CSPs compete with each other by offering special packages for their customers. Generally, the customers churn is the most loss-making event for CSPs. Having as much information as possible about customers can significantly reduce this phenomenon. Data does not need to be processed in real time and the more information about customers the better chance to reduce churn.

4.3 EXOGENOUS INFLUENCERS USE CASES

This chapter is focused on exogenous influencers use cases derived from real industry problems on the current market. The selected use cases in this chapter are: Fraud prevention, External Social Network.

Fraud prevention

This is a very complex topic and for CSPs it primarily means to have customer mobile devices under control in terms of security issues. Typical use case can be if someone steals one's mobile phone then the usage of that phone changes. It is all based on creating usage patterns that are then compared with current behavior in the network stream and analyses are created to prevent fraudulent actions. This requires data from CDR and network probes and this data is mainly used by security leaders.

External Social Network

Social networking is also one of the biggest topic of today's world. Facebook, Twitter, LinkedIn and many others contain enormous amount of data. Connecting all the information from CRM, CDR and network probes with social networks can provide 360° view of customers and can show customers sentiment and the way they behave. Building on what have been written about social networks, sometimes it is not the only way to observe how people behave but to observe how social network leaders (opinion makers) behave. These people can significantly affect groups of interest and show current trends. The data is obtained primarily from social networks and is used by marketers and business strategists.

4.4 TECHNICAL USE CASES

This chapter is focused on technical use cases derived from real industry problems on the current market. The selected use cases in this chapter are: Call routing and network optimization, Web traffic analysis and Location based analysis.

Call routing and network optimization

One of very important sources of revenue is also network infrastructure development. With the knowledge about customer needs and geolocation movements CSPs can optimize and plan their infrastructure so that they can provide their customers with up-to-date service coverage while reducing costs. These data comes from network probes and is used by infrastructure engineers and radio network planners.

Radio network traffic analysis

This use case is focused on customer behavior in radio network and is very often combined with an extension of current user profiles by rating of their internet usage. These analyses can be connected with all the use cases mentioned above to get better and more precise insight about customers. Data is processed mainly in ex post mode and comes from network probes. The group of interest is formed by radio network planners and marketers.

All the mentioned use cases above can be combined together to create new use cases. The rest of this thesis is focused on the mobile users movement prediction use case, described in Section 7, that is primarily an adjusted combination of Location based analysis and Reaction to an event use cases because CSPs pay a lot of attention to this topic as it can bring new revenue income and extended customer services.

5 STATISTICAL METHODS FOR PREDICTION

There are a lot of predictive methods and they can be compared on many aspects and parameters. Prediction methods giving an insight to what will happen in the future generally are based on historical data. To provide reasonable prediction, it is necessary to have as much data as possible. This implies requirements on storing and even transforming a huge amount of historical data. In other words, the more data patterns, based on historical data, the more efficient prediction. Going deep into all predictive methods is not the subject of this work. Thus, only the most promising methods are briefly discussed. After discussing all predictive methods, the theory of neural network is given in general with the emphasis on the NARX network in the next chapter. Last section informs about statistical algorithms that are used for simulating the mobile users movement prediction use case in Section 6.

5.1 PREDICTION METHODS AND THEIR COMPARISON

There is no hard-set boundary that would divide all methods into exactly the same boxes. This work divides predictive methods into 3 parts [36]: Trees & rules algorithms [37], Statistical models and neural networks. They all can be investigated from many points of view with different parameters. This chapter shows 3 of them: the velocity of algorithm, the accuracy and input data.

5.1.1 Trees & rules

These models are generally very fast and are capable of processing a huge amount of data. They are very simple to create and very effective with basic rules inside. They can be accurate but only with input data without missing values and with no relationships among them. The Trees & rules basic architecture is depicted in Fig. 12.

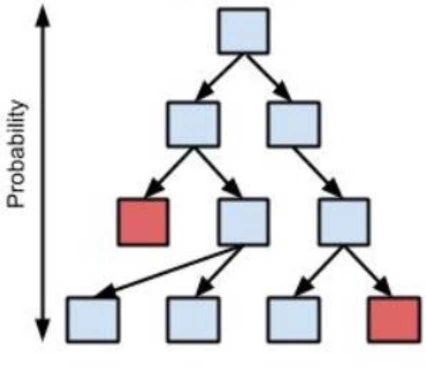

Fig. 12: Trees & rules basic visualization

The basic algorithm QUEST [36] is based on a selection of attributes and this special feature in this technique works with negligible bias. This algorithm can be used for solving problems with non-invariant or linear combination splits due to the fact that all attributes have approximately the same change to split a node. Next algorithm, called C4.5 [36], was derived from Quinlan's earlier ID3 algorithm and works with the concept of information entropy. Decision trees generated by this algorithm are widely used for classification. That is why it is called a statistical classifier. The last algorithm is called FACT [36] and works with statistical tests. Every node is split by attributes selected by statistical tests and then discriminant analysis is applied to get the split point. FACT algorithm is based on fuzzy statistics that makes it very suitable for complex real problems.

5.1.2 Statistics models

Statistical models have almost the same accuracy as Trees & rules models according to [36]. The biggest difference is in the velocity. In many application it is the most essential requirement to process data as fast as possible. They are many times slower than Trees & rules models. Thus, these algorithms fit to tasks and problems that are not sensitive to the process velocity. The statistics models basic architecture is depicted in Fig. 13.

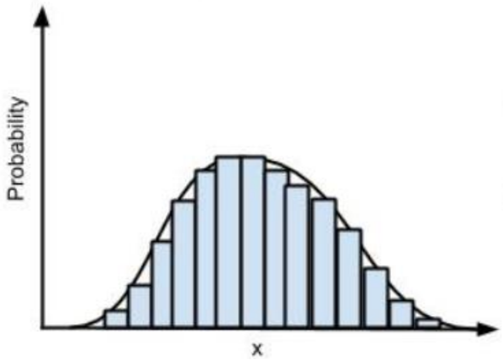

Fig. 13: Statistics models basic visualization

Linear Discriminant Analysis [36] as widely accepted basis of statistical models works with linear discriminant functions. Each class is specified by its instances that are normally distributed with a common covariance matrix. Linear discriminant analysis is a method very commonly used in various real statistical problems from bankruptcy prediction to face recognition. Next algorithm is called Quadratic Discriminant Analysis [36]. It is very similar with LDA in terms of having normal class distribution except for covariance matrix. It is estimated by corresponding sample covariance matrix and it leads to quadratic discriminant functions. QDA generally solves the same problems as LDA only with the difference that the number of parameters increases significantly. The last algorithm stands for penalized LDA [36]. If there is a lot of highly correlated attributes then this algorithm is used. A penalized regression framework solves the classification problem by optimal scoring method. It is widely used for speech and handwritten character recognition.

5.1.3 Neural networks

This predictive method is a key part of data mining methods examined by many scientist and engineers. It brings a totally new view on data processing. Neural networks are not as accurate as the two methods mentioned above and learning process is even slower at the beginning. But after being trained they are faster and they can learn the mutual relationships among input data. The Trees & rules basic architecture is depicted in Fig. 14.

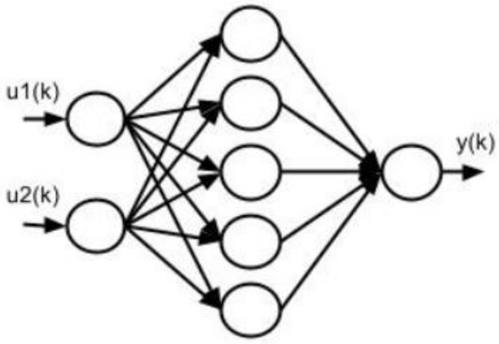

Fig. 14: Neural networks basic visualization

SOM [39] or Self-organized maps or also Kohonen Maps is a commonly used neural network for classification problems or data clustering of unknown data. There is a given topology of neurons at the beginning of the process. Then the network tries to find out some similar features within the input data and created clusters (bounded places with similar features). These clusters are formed by using Euclidean metrics. Finally, the clusters are displayed in maps for next analyses. The next neural network is called BPG [36] or Back-propagation of gradient algorithm. It is a feedforward network, commonly consists of 3 layers (input, hidden and output layer), with a "teacher", exogenous set of data, that helps to find out the best weights for the learning process in the direction of negative gradient. In the output layer there is a calculation in each neuron compared to the "teacher". The arisen differences (errors) are sent back to the previous layers and the synaptic weights, providing the network memory, are adjusted. Then the input data are again presented to the network and the process goes over and over again unless the given conditions are met. The last neural network is called Hopfield neural network [40] and has a different topology from the 2 networks mentioned above. It has a circle topology consisting of formal neurons [38] that are fully connected. The output gives only 2 results {-1, 1}. All neurons or some of them are set to one of the output values which create a pattern. Series of repetitive processes and a dynamic character of the network are able to reproduce the inserted pattern commonly used for pictures restoration. This is called content addressable memory.

5.2 GENERAL THEORY OF NEURAL NETWORKS

Neural networks are able to solve very difficult and profound problems. This extremely powerful method is inspired by the human brain. This method should not exchange actual conventional methods but rather be a complement part of these methods. Neural network consists of simple elements operating in parallel. The elements connected among each other create a neural network, see Fig. 15, and are called neurons. This essential element can be described by "Formal neuron" depicted in Fig. 16 that consists of input (signals) vectors, the weights creating synaptic weights (network's memory), the bias and an activation (transfer) function.

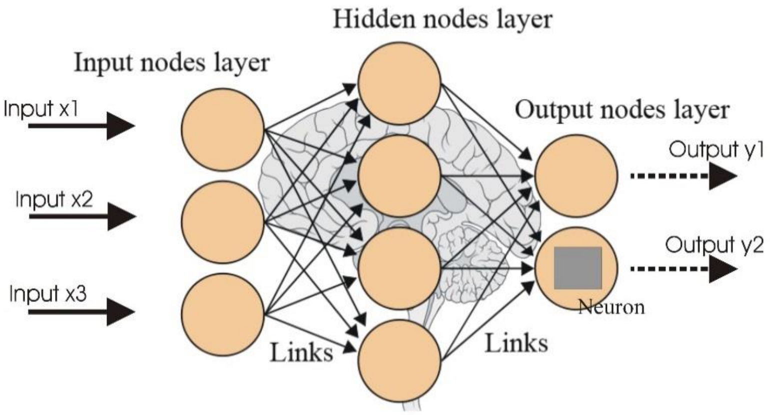

Fig. 15: A neural network in general [57]

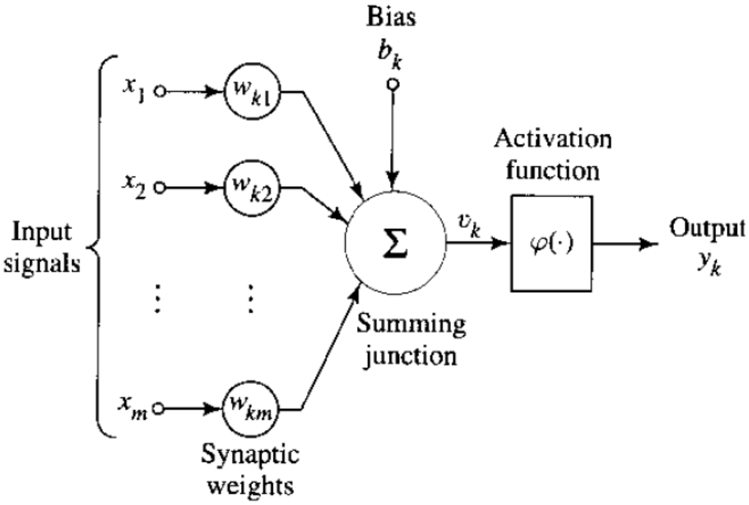

Fig. 16: Essential element of neural network, a neuron [41]

As it can be seen, the input vector *x* is multiplied by the weights *w* and then the bias *b* is added. The bias can shift the activation function and change the range of value going towards the output. Generally, the activation function can be linear, hard-limit, tangent or sigmoid and many others [42] depending on what type problem is solved.

Typically, a neural network starts with input data multiplied by weights (network memory) going through the hidden layer and is finally formed to an output. The hidden layer can have various amount of neurons generally depending on accuracy of the network. The more neurons in the hidden layer the more time spent on calculation and the better accuracy of the network. The output is then compared with a target (a teacher) and the weights are adjusted. This all is called the training process as depicted in Fig. 17. This iteration process is repeated over and over again until the difference between the target and the output reaches a predetermined value (typically error value) [43], but can be stopped by different conditions.

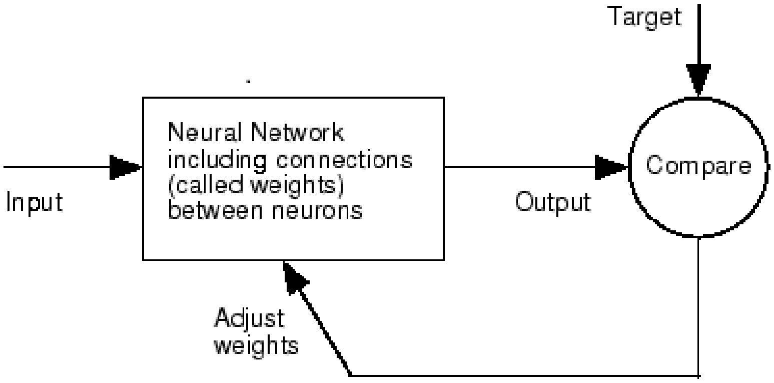
Target
Neural Network including connections (called weights) between neurons
Input
Output
Compare
Adjust weights

Fig. 17: A complete training process visualization [43]

To sum up the most important benefits [41], neural networks has, first, a **structure that is** very similar to the human brain and is **massively distributed in parallel**, second, the possibility to understand the environment and the relationships among the data called **generalization or learning**. Many useful capabilities are provided by neural networks, see the most important ones [11] in Tab. 2.

Neural network		
#	Capability	Description
1	Nonlinearity	Very important for description of particular processes
2	Input-output mapping	Learning with a given target (teacher) and synaptic weights modification
3	Adaptivity	Neural network can be adjusted according to a given environment
4	Contextual information	Every neuron in the network is affected by the global set of other neurons

Tab. 2: Neural network capabilities

5.3 NARX NETWORK

The nonlinear autoregressive network with exogenous inputs provides very accurate chaotic time series prediction [55, 68], that perfectly fits to this work because it solves 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.

5.3.1 Dynamic networks

Before working with the nonlinear autoregressive [44] network with exogenous inputs (NARX) it is necessary to take a look at the core of the network. There are two groups of this network: the feedforward one and the feedforward one with recurrent connection among the neurons. To understand it correctly there are shown three types of network providing the differences between static, feedforward-dynamic, and recurrent-dynamic networks.

Very briefly explained, static networks provide us with the same length of the output vector as the input vector [45]. The architecture looks as follows in Fig. 18.

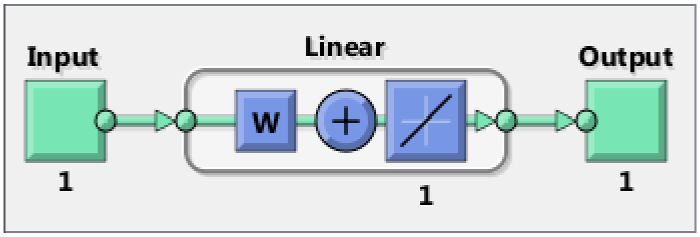

Fig. 18: Static network

On the other hand, feedforward-dynamic networks give us longer response to the input data which is caused by the memory [45]. 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 FIR (finite impulse response) filter as seen in Fig. 19.

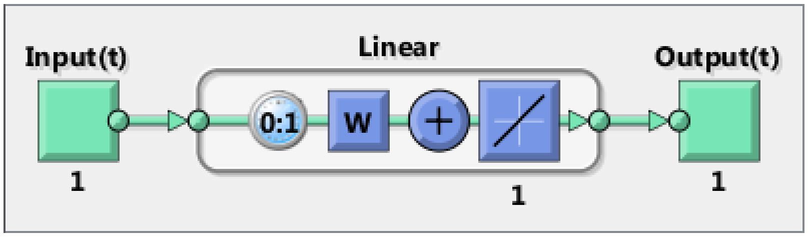

Fig. 19: feedforward-dynamic network

Finally, feedforward-dynamic networks with recurrent connection has even longer output data response [45]. This recurrent connection let the network investigate the data relationships more in detail with better accuracy. As it can be seen, the input data is not delayed and the recurrent connection is delayed by 1 time step. The process is the same as in non-recurrent feedforward-dynamic networks expect for that the response value never reach zero value. That is called IIR (infinite impulse response) filter as follows in Fig. 20.

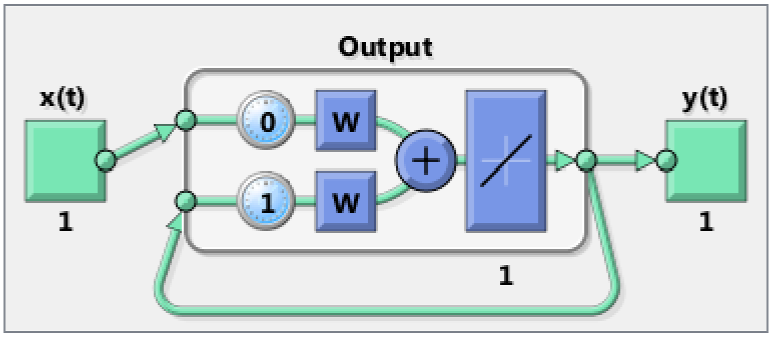

Fig. 20: Feedforward-dynamic networks with recurrent connection

5.3.2 NARX network architecture

Nonlinear autoregressive [19] network with exogenous inputs works on the assumption of the recurrent delayed feedback principle that is widely used for time-series prediction tasks. It is the third type of dynamic networks, recurrent-dynamic networks as described above in Fig 20 and typically consists of 3 layers (input, hidden, output) as BPG networks in chapter 5.1.3. As it has already been mentioned, the NARX network contains a recurrent connection enclosing several layers of the network as seen in Fig. 21. Functionality of the basic equation for the NARX network can be given as follows:

$$y(k+1) = f(y(k-1), y(k-2), \dots y(k-n_y), u(k-1), u(k-2), \dots u(k-n_u)) \qquad (1)$$

where the output vector *y(k+1)* is computed as a nonlinear function of the input vector *u(k), u(k-1), ..., u(k-du)* which has a predetermined delay. Besides, the output is affected by a recurrent connection *y(k), y(k-1), ..., y(k-dy)* that goes from the output back to the input layer.

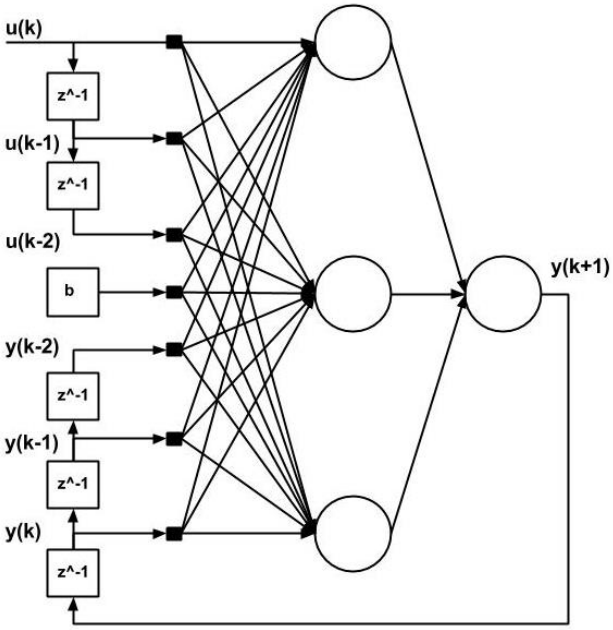

Fig. 21: NARX network architecture [46]

5.3.3 Long-term dependence problem

A big problem in prediction of any values and not only in prediction tasks is the time dependence of input and output values. It has already been proven by many researches that common gradient-descent learning algorithms are not so effective in solving long-term dependence problems or even cannot solve them at all [56, 73]. The problem is the fact that the network forgets the influence of past values to the actual output. In other words, the gradient vanishes after a given amount of steps and the network's memory is lost. A classical recurrent network has problems with long time dependencies, especially when calculating prediction of nonlinear tasks.

$$z_k(k+1) = \begin{cases} \Phi(u(k), z_i(k)), & i=1, \\ z_i(k), & i=2,3,\dots N \end{cases} \qquad (2)$$

where the output $y(k) = z_1(k)$ and state variables $z_i, i = 2,,3,\dots,N$, represent a recurrent neural network. Vanishing gradient or forgetting of network's behavior comes into play when

$$\lim_{m\to\infty} \frac{\partial z_i(k)}{\partial z_j(k-m)} = 0 \qquad (3)$$

where *z* represents state variables, *j* denotes input neurons and *i* denotes output neurons respectively. The *k* a *m* parameters refer to the time index set.

Researches have already suggested a solution for the vanishing gradient in training of recurrent networks. All of them finally agreed about either to include memory in neural network or using more convenient learning algorithms like Kalman filter algorithm, Newton type algorithm, etc. [47]. The vanishing gradient can be significantly reduced by applying embedded memory that can be created by using recurrent relations between neurons, time delay across all layers and applying activation function that can sum input over time. NARX network has the advantage of having the first two mentioned improvements that can reduce or almost eliminate the effect of vanishing gradient.

6 MOBILE USERS MOVEMENT PREDICTION

People are used to get all mobile services and information while moving which is a new trend of this century. To be able to provide all customers with this information there is a need to know the position of a mobile user [50]. All that together creates one part of communication chain in technical terms. A prediction (Latin *prœ-*, "before," and *dicere*, "to say") is one of data mining methods thanks to that people can forecast future happenings from past values, experiences or knowledge. Creating patterns is the key to the right prediction as it contains behaviors, habits and daily activities of people. As mentioned in section 1, there is a tremendous amount of data around us and it applies even more for telecommunications branch where sensors send rich information but mostly in unstructured form. These data hides valuable information that can be turned into money and the possibility to get the insight from that unstructured data, for example by using prediction algorithms, is the way to new intelligent services.

This chapter describes algorithms used for prediction tasks: the Gradient descent with adaptive learning algorithm, the Optimized gradient descent with adaptive learning algorithm and the Levenberg-Marquardt algorithm. Further describes scenario definition, performance matrices and final results with a comparison of the selected algorithms. This chapter is focused on mobile users movement prediction in mobile networks. The attempt is to simulate and estimate the next steps of a mobile user in geolocation coordinates. Such information can be exploited by mobile service providers for allocating resources more effectively, efficient location update procedures and location search techniques [50]. To go even a bit deeper we can take a look at mobile service providers' bandwidth. It is a very scarce and expensive natural resource and it could be used more efficiently thanks to prediction. Location updates and paging are messages moving between BTS (base transceiver station) and mobile user device carrying information about locations sent by users (updates) and the necessity of the core network to look for the user positions (paging). These messages take a significant part of the bandwidth and their reduction would lead to a possibility to use other services, Quality of service (QoS) would be higher and the resource prediction could bring interesting view of users' habits at the right place and even at the right time.

6.1 ALGORITHMS FOR PERFORMANCE EVALUATION

In this section, there is a description of algorithms that are commonly used for prediction problems. These algorithms are used for evaluating the network final results performance by the MSE and the gradient method. The Gradient descent with adaptive learning algorithm, the Optimized gradient descent with adaptive learning algorithm and the Levenberg-Marquardt algorithm are selected because the first one mentioned has the adaptive learning technique to obtain more accurate results faster and the second one is the optimized modification of the first algorithm. The last algorithm uses the same adaptive technique with the ability to overcome the vanishing gradient problem described in 5.3.3.

6.1.1 The gradient descent with adaptive learning algorithm

The backprogation algorithm [56] is used for many applications solving real life problems. This training procedure is based on the error function by which is a network performance evaluated. The whole purpose of the error function is to calculate and evaluate differences between the output signals and the required target signals given as follows:

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_L}(t_j - a_j)^2 \qquad (4)$$

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 training process depends on an iteration process where the error is reduced as much as possible to obtain the desired mapping. An input set of data is presented to the network sequentially. The network tries to remember all connections among each value of the input signal and creates patterns. These patterns serve as a network memory and then newly coming data is compared with the pattern. An iteration process progressively optimize connection weights (network memory) until the desired mapping is obtained.

A gradient descent technique is used to minimize the error function. This method is based on calculating the partial derivative of the error function. 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 negative sign. The gradient direction along with a step size is calculated as follows:

$$\Delta w_{i,j}(n) = -\eta \frac{\partial E(n)}{\partial w_{i,j}(\mathrm{n})} \qquad (5)$$

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

Each iteration (epoch) can be divided into three parts:

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 a_j is dependent on every *j* step of the iteration process in layer *l* and is calculated using a sigmoid activation function:

   $$a_i = f(net_j) \equiv \frac{1}{1+e^{-net_j}} \qquad (6)$$

   $$net_j = \sum_{i=1}^{N_{l-1}} w_{i,j}a_i + \theta_j \qquad (7)$$

   where each index *i* from layer *l-1* is connected with index *j*. θ_j means w_{0j} or is called the bias.

2. The generalized delta rule
   This step describes the calculation of the local gradients. An update in each iteration process is defined as follows:

   $$\Delta w_{i,j}(n) = \eta\delta_j a_i \qquad (8)$$

   The parameter δ_j [48] helps to finally go through the whole iteration process solving the problem with the local minimum.

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*. 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. 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.

6.1.2 The optimized gradient descent with adaptive learning algorithm

This algorithm is the same as the Gradient descent with adaptive learning algorithm in terms of mathematical core. This algorithm is able to do the same calculations. The algorithm is adjusted so that it works with adaptive changing of network parameters over the training time. Once a network is trained, it is necessary to repeat the training process with different parameters in order to make sure that the calculated results are the same or similar in more training modifications. In terms of neural network the parameters of the network in an iteration process can be changed either manually or using this type of algorithm.

The first option, manual setting of parameters, is not so time-consuming and is easier to be carried out. The first option is appropriate only if a data input is well-known, the required results are obtained and only few parameters need to be changed in order to verify the correctness of the network. The second option, using this type of algorithm, is more time-consuming and brings more troubles to be carried out. If a data input is unknown and results are unpredictable at the beginning of the training process it is the right time to use this type of algorithm that goes through all possible combinations of preselected parameters to obtain as accurate result as possible. One parameter is selected to determine the best possible combination of parameters. In this work, the *r* parameter, which stands for the regression, is calculated in every iteration and if a new regression value is greater than the previous one the network parameters are changed and saved. The variable parameters are the ones that affect network results the most: the number of hidden layers, maximum epochs, maximum validation checks, learning rate, learning rate increase, learning rate decrease. Parameters are adjusted as follows:

1. The default value of a selected parameter is taken as the center value for an optimization vector.
2. The optimization vector has its given minimum and maximum derived symmetrically from the center value. For example: [1 3 5 7 9] or [100 150 200 250 300], where the value 5 and the value 200 are the default center values.
3. The training process takes the first combination of all selected parameters.
4. The regression value is calculated, the best parameters combination is adjusted based on the best regression result in each iteration step.
5. Finally, new parameters are saved and the optimization process goes again from step 3 for a new combination until it goes through all possible combinations of selected parameters.
6. The final regression value and network training parameters are set.

6.1.3 The Levenberg-Marquardt algorithm

Although gradient based training is accurate it is very slow as well and even more complications appear because of the fact that the gradient vanishes at the solution (the network behavior is forgotten). On the other hand, the Levenberg-Marquardt algorithm [49] which is derived from Hessian based algorithms [59] can investigate data more in detail and provide even more accurate and subtle features. The Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field. It solves problems with the local curvature of a function of many variables. The Hessian matrix is closely connected with the Jacobian matrix. 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. 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 Levenberg-Marquardt algorithm is basically a Hessian-based algorithm for nonlinear least square optimization [8] and that is why we take the advantage of this algorithm in creating the neural network. Generally, for neural networks the most important function is the error function as follows:

$$e = \sum_{k=1}^{P} \frac{1}{2} (t_k - y_k)^2 \qquad (9)$$

where y_k is the output for the *k-th* pattern and t_k is a desired output, P is the total number of training patterns. To understand better the process of this algorithm, a step-by-step description of this training method in neural networks is given as follows in pseudo code [58].

```
InitializeWeights;
while not StopCriterion do
        calculate e(z) for each pattern;
                $e1 = \sum_{p=1}^{P} e^P(z)^T e^P(z)$ ; derived from (9)
        calculate J(z) for each pattern;
        repeat
                calculate $\Delta z$ (10);
                calculate e2;
                        $e2 = \sum_{p=1}^{P} e^P(z + \Delta z)^T e^P(z + \Delta z)$ ; derived from (9)
                if (e1 <= e2) then
                $\mu := \mu * \beta$;
                endif;
        until (e2 < e1);
        $\mu := \mu/\beta$;
        $w := z + \Delta z$;
endwhile;
```

where the *J(z)* parameter presents the Jacobian matrix, the *e(z)* parameter denotes the error of the network for pattern *p*. The *β* parameter a factor that increase or decrease the μ parameter which is the most variable parameter called the learning parameter. This parameter varies over the time with iteration process. If it is 0, then the algorithm changes into Gauss-Newton method. On the other hand if this parameter is very large, then it can turn into steepest decent or the error back-propagation algorithm.

The convergence of the algorithm is given by either reaching some predetermined values of the gradient or the error decrease under the predetermined error limit.

The vector Δz is calculated as follows:

$$\Delta \mathrm{z} = [J^T(z)J(z) + \mu I]^{-1} J^T(z)E \tag{10}$$

where E is the vector with the length of P as follows:

$$E = \left[t_1 - y_1\; t_2 - y_2\; \dots t_p - y_p\right]^T \tag{11}$$

The Hessian matrix is given by $J^T(z)J(z)$. Vector I refers to the identity matrix. The Jacobian matrix $J(z)$ is also calculated in each iteration step along with Hessian matrix $J^T(z)J(z)$ that brings about slowing down of the algorithm.

6.2 Evaluation framework and scenario

In this section, there is a description of the overall environment for the network which has to be set in order to carry out mobile users movement prediction simulation. In the first part, this section focuses on data collection, data preprocessing, network creation, network configuration, network training and network optimization. Then, performance matrices are given. Finally, results, showing the mobile users movement prediction, are discussed in the last part.

6.2.1 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 preprocessing should be carried out as described in chapter 7.1.2. The dataset was obtained in .csv format in Microsoft Excel. It consists of 5 columns (areaID.cellID, areaID, cellID, Latitude, Longitude) and 12975 rows. 12575 rows were used for training, validation and testing process and 400 rows (user's steps) were predicted. AreaID.cellID is derived from the Reality Mining dataset from The Massachusetts Institute of Technology and is completed by a set of measured data in the remaining fields [51]. CellID is a unique identifier of each BTS within a Location area code displayed as areaID [52]. Only one of geolocation coordinates is computed by the network in order to find the best performance results. The areaID,cellID column is not considered as this information is redundant in terms of prediction. Latitude and longitude are measured in meters. This prepared data is used to predict a mobile users movement by the geolocation coordinates as there are enough correlations among the data. Data is time variant, in other words, there is a long-term dependence as explained in chapter 5.3.3. That makes it a suitable set of data for prediction investigation with NARX neural network.

6.2.2 Data preprocessing

Generally, data can be processed immediately without using any preprocessing methods and in some cases it is even more convenient 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 nonlinear trends (rising or decreasing values) or suffer from seasonality (patterns caused by a factor that is repeated over and over again) [46]. Very important part of data selection is to find mutual relationships among data called correlations. The idea is to preprocess 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. At that case, there are other neural networks (see chapter 5.1.3) like SOM neural networks or its modifications, that can solve this problem but it is not the objective for this work. Data correlations can be calculated as follows:

$$R = \frac{\sum_{t=1}^{N}(y(t)-\bar{y})(y(t+k)-\bar{y})}{\sum_{t=1}^{N}(y(t)-\bar{y})^2} \qquad (12)$$

where *y(t)* is an output vector and *k* stands for the time delay. To calculate the mean value of *y* the following equation is given:

$$\bar{y} = \sum_{t=1}^{N}\frac{y_t}{N} \qquad (13)$$

Sometimes the outliers (values that differ a lot from the mean) 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 [-1, 1]. This normalization step can significantly speed up the training and produce more accurate results.

6.2.3 Network creation

When knowing the environment, data preprocessing, architecture and topology of the network the first step is to load a prepared set of data commonly in .csv format. The loaded vector has to be then converted into standard neural network cell array form that can be processed by neural network code. The basic parameters have to be set as follows

- Training function (algorithm)
- Input delays
- Feedback delays
- Number of Hidden layers
- Open or closed loop of the network (this is discussed more in detail in the chapter "Network training")

Finally, the data set is preprocessed by MATLAB functions and ready for training. At the beginning the trial-and-error method is applied while observing changing results. For this work, the network is set to the architecture 4-5-1 as seen in Fig. 22. The first four input layers indicate four columns (variables) in a given data set as described in 6.2.1. The five hidden layers are set by the neural network by default and the number of hidden layers is adjusted according to results obtained from network training and optimization described in chapters 6.2.5 and 6.1.2. The last layer denotes to the predicted mobile user movement vector.

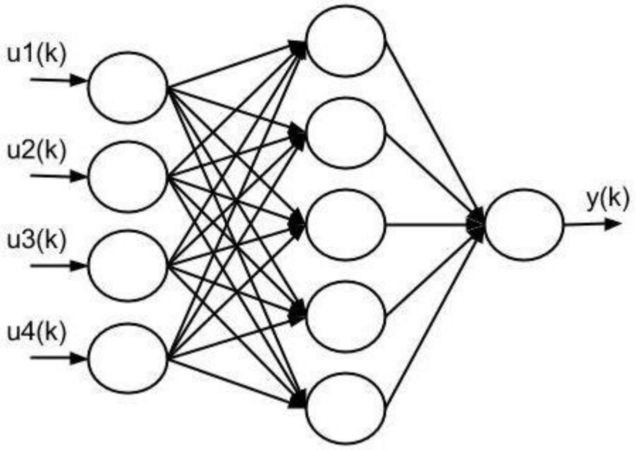

Fig. 22: The 4-5-1 architecture

6.2.4 Network configuration

The dataset is divided into 3 parts: training, validation and testing. Typically, data is divided in a ratio of 70 % for training, 15% for validation and 15% for testing by default. This division is applied because of error, generalization and performance measurement. These three parameters are used for evaluating of the network. Training data is responsible for calculating errors during the training process. Validation set is used for having the generalization under control and stopping the training process when it reaches a predetermined value. Testing has no connection to the training process at all and provide an independent check of performance during and after training.

All the network parameters, data division, number of layers and delays were set to default values and are shown in Tab. 3. This configuration is used for the first result in section 6.4.

Training function	traingda		
Architecture parameters		**Data distribution**	
Input delays	1:2	Training data	70 %
Feedback delays	1:2	Testing data	15 %
Number of hidden layers	5	Validation data	15 %
Training parameters			
Maximum Epochs	1000	Learning rate	0.01
Maximum Training Time	Inf	Learning rate increase	1.05
Performance Goal	0	Learning rate decrease	0.7
Minimum Gradient	1e-7	Maximum Performance increase	1.04
Maximum Validation Checks	6		

Tab. 3: Configuration for Gradient descent with adaptive learning algorithm

Tab 3. shows that the Gradient descent with adaptive learning backpropagation algorithm (in MATLAB denoted as traingda) is used, input and recurrent connections are delayed in a ratio of 1:2 which creates the network memory to solve the vanishing gradient problem described in chapter 5.3.3 and provide more efficient and accurate prediction result because of the weights adjustment over the training time. Five hidden layers is set by default and the number of hidden layers is adjusted according to results obtained from network training and optimization described in chapters 6.2.5 and 6.1.2. In the training parameters part, Maximum amount of training cycles is 1000 and there is no limitation in terms of Maximum Training Time. The network Performance Goal should incline to 0 to obtain the most accurate results and the Minimum Gradient should incline to 1e-7, this value is sufficient in most cases as the human senses are not capable of capturing such subtle parameter deviations. The Validation Checks describes how many equal states of the training stops the network, in other words, if the Maximum Validation Check reaches the given limits, set to 6 in this case, then the training process stops. The Learning rate defines the velocity of the training process. The higher increase of the Learning rate the faster learning and vice versa and the right setting depends on the data set character. The Maximum Performance increase determines how big performance steps can be done to obtain the required performance close to 0. The higher value of the Maximum Performance increase the faster training process and very often the worse accuracy.

6.2.5 Network training

As it has already been mentioned above, the NARX network can work in 2 modes (see chapter 5.3.1). The first, the open-loop mode is generally used for the network training part when the output data is known. 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. 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 has time steps. Normally, the network gives *y(t-1)* result at the same time when the network reads *y(t-1)* input data. 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 timeline one step to the left so that the minimum delay starts at 0 value instead of 1.

There are two very problems that should be taken into account and reduced or eliminated before training. First, static neural networks can be frozen in local minima [53]. We do not need to be worried about that as the NARX network 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. 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 [46] as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{14}$$

$$MSW = \frac{1}{n}\sum_{j=1}^{n}(w_j)^2 \tag{15}$$

$$MSEreg = \xi MSE + (1-\xi)MSW \tag{16}$$

where the number of values in the error vector is given by the *N* parameter for the MSE and the *n* parameter for the MSW, t_i presents the target vector and y_i indicates the predicted vector. The w_j parameter presents weighs while the ξ parameter is the performance ratio.

The NARX network can be stopped on 3 main conditions. The first one is the performance or the MSE (mean squared error) that calculates errors between output and target vectors. The next one is the gradient limit that can be considered the learning rate. The last one is the validation check that stops learning when a predetermined number of validation steps is reached or increase.

6.3 PERFORMANCE MATRICES

- **Nonlinear regression function [54]**

  Generally, nonlinear regression defines a nonlinear regressive model of function *f*.

$$X_R = f(X_i, \beta) + \varepsilon \qquad (17)$$

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

- **Autocorrelation function [55]**

  Generally, autocorrelation function indicates an autocorrelation model, where vector $\boldsymbol{y}$ is shifted in time and compared with itself. It is a function for finding repeating patterns.

$$R_{yy}(k) = \frac{1}{T-1} \sum_{t=1}^{T-k} (y_t - \bar{y})(y_{t+k} - \bar{y}) \qquad (18)$$

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

  For a perfect prediction model, there should only be one non-zero value of the autocorrelation function and it should occur at zero lag. The formula for the autocorrelation for lag *k* is as follows:

$$r_k = \frac{R_{yy}(k)}{c_0} \qquad (19)$$

  where c_0 is the sample variance of the time series.

- **Mean Square Error [46]**

  Generally, the MSE of an estimator measures the average of the squares of the “errors”. It means the difference between what is estimated and the estimator.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (t_i - y_i)^2 \qquad (20)$$

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

6.4 RESULTS

This section shows the final results of mobile user movement prediction using the NARX network. Three types of algorithms, used in the NARX network, are presented in order to show differences between them: the Gradient descent with adaptive learning algorithm, the Optimized gradient descent with adaptive algorithm and the Optimized Levenberg-Marquardt algorithm. The final chapter sums up the obtained results.

6.4.1 Gradient descent with adaptive learning algorithm

The results in this chapter are calculated with default values as defined in chapter 6.2.4 for the Gradient descent with adaptive learning algorithm described in 6.1.1.

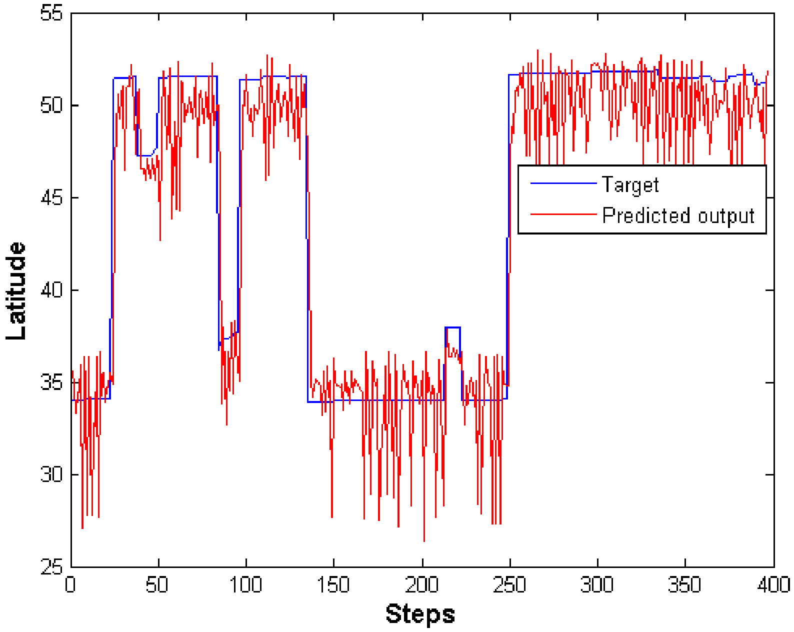

Fig. 23: Mobile users movement prediction in geolocation coordinates

Fig. 23 depicts 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 but the predicted line suffers from significant jitter which would bring about significant problems with mobile user movement prediction. Firstly, this is caused by the vanishing gradient (2), (3) during the learning process and secondly by the fact that this algorithm is unable to predict accurately, although it generally used for prediction, in terms of this prediction task.

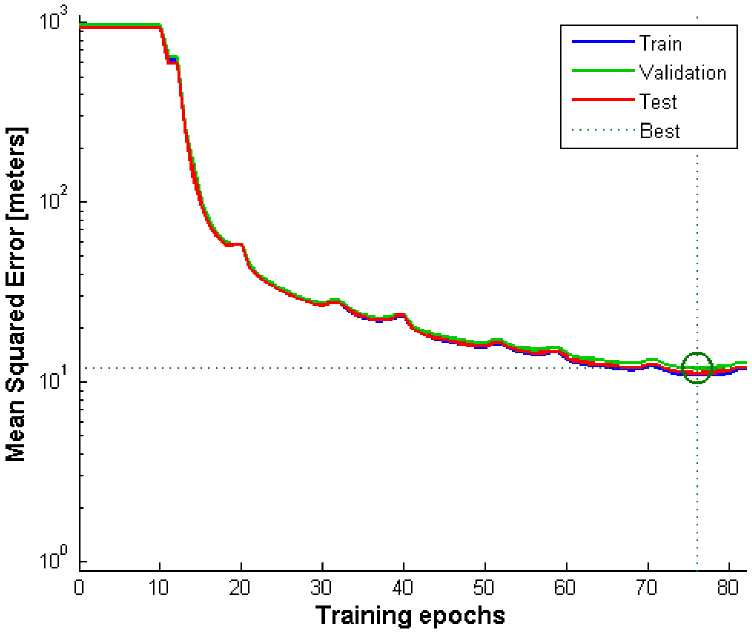

Fig. 24: MSE for best train, validation and test performances

Fig. 24 depicts the MSE during the training process (blue line), the MSE of the validation process (green line), and the MSE for the test process (red line). The dotted line only graphically displays the minimum MSE and the number of epochs required to reach the minimum validation MSE. Fig. 24 shows how many epochs (cycles of the training) are needed to get the minimum validation MSE (20) of 11.8746 meters which is not sufficient result from the performance point of view, the closer to the zero value the more accurate result. Also can be seen that the network training takes a lot of time, 76 epochs, to reach desired results according to the curves that slightly decrease during the training. Less epochs could be used to get results faster but at the cost of even worse results.

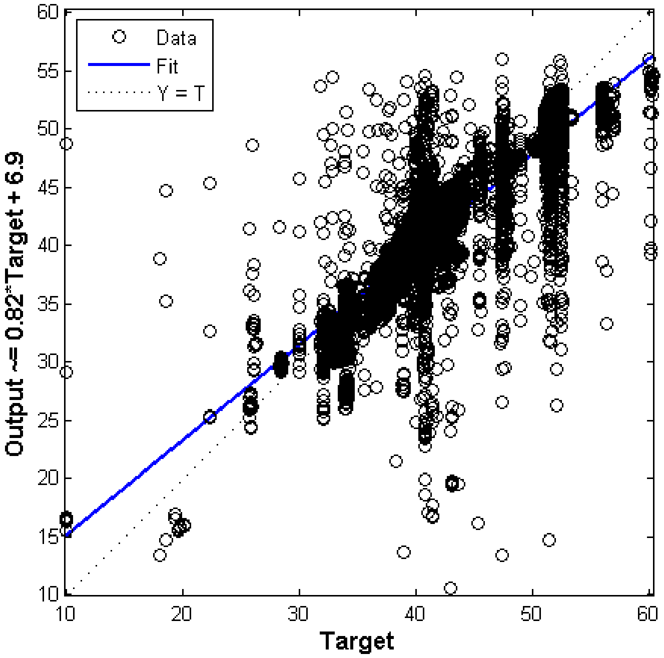

Fig. 25: Regression performance

Fig. 25 depicts the regression (17) toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (blue line). Fig. 25 also shows 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 prediction case, the blue line should copy the dotted line and the X_R parameter would have to be one. The result of the regression is 0.8704 which equals to approximately 87 % similarity between the target and the predicted output. Such similarity is not sufficient for movement prediction.

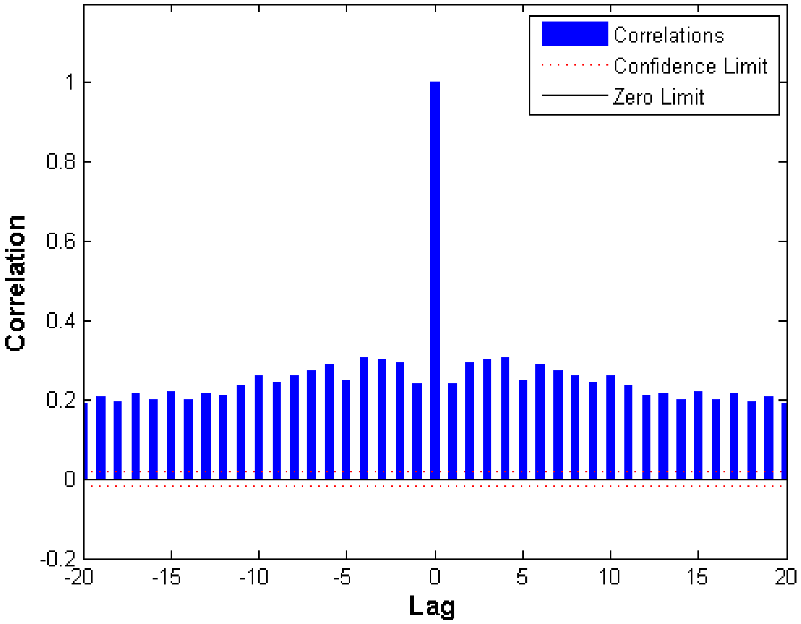

Fig. 26: Autocorrelation of errors

Fig. 26 depicts the correlated errors in time steps (blue bar lines) and the Confidence Limit line (dotted lines) showing limits inside which the prediction is almost perfect. Fig. 26 shows how the predicted errors of the calculated output and the target are related in time (18). The value 1 at zero lag indicates that the errors are the same if there is no time delay. Other non-zero lags evince some correlations that indicate some inaccuracies during the training process according to the non-zero blue bar lines leading to inaccurate prediction.

6.4.2 Optimized gradient descent with adaptive learning algorithm

This chapter provides results based on the optimized Gradient descent with adaptive learning algorithm. The optimization is done using iteration process which is able to find the best possible combination of the network parameters as described in 6.1.2. The optimized parameters are highlighted in bold in Tab 4. These parameters in bold are selected because they affect the network results the most when they are changed.

Training function	traingda		
Architecture parameters		**Data distribution**	
Input delays	1:2	Training data	70 %
Feedback delays	1:2	Testing data	15 %
Number of hidden layers	**3**	Validation data	15 %
Training parameters			
Maximum Epochs	**150**	Learning rate	**0.001**
Maximum Training Time	Inf	Learning rate increase	**1.1**
Performance Goal	0	Learning rate decrease	**0.5**
Minimum Gradient	1e-7	Maximum Performance increase	1.04
Maximum Validation Checks	**9**		

Tab. 4: Configuration for optimized Gradient descent with adaptive learning algorithm

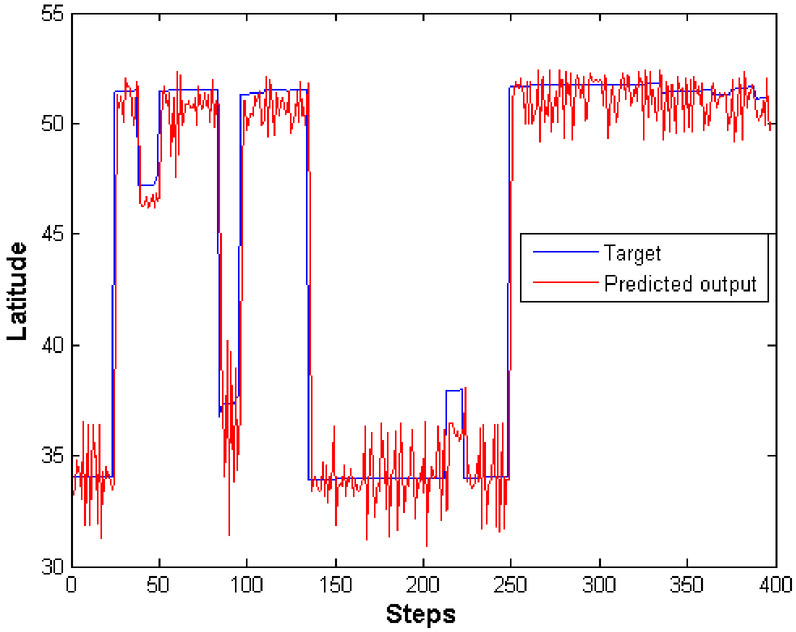

Fig. 27: Mobile users movement prediction in geolocation coordinates

Fig. 27 depicts that the trend of the predicted line movement follows the target 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 cases. Firstly, it is again caused by the vanishing gradient (2), (3) during the learning process and secondly by the fact that this algorithm is unable to predict accurately in terms of this prediction task even if there is the optimization applied.

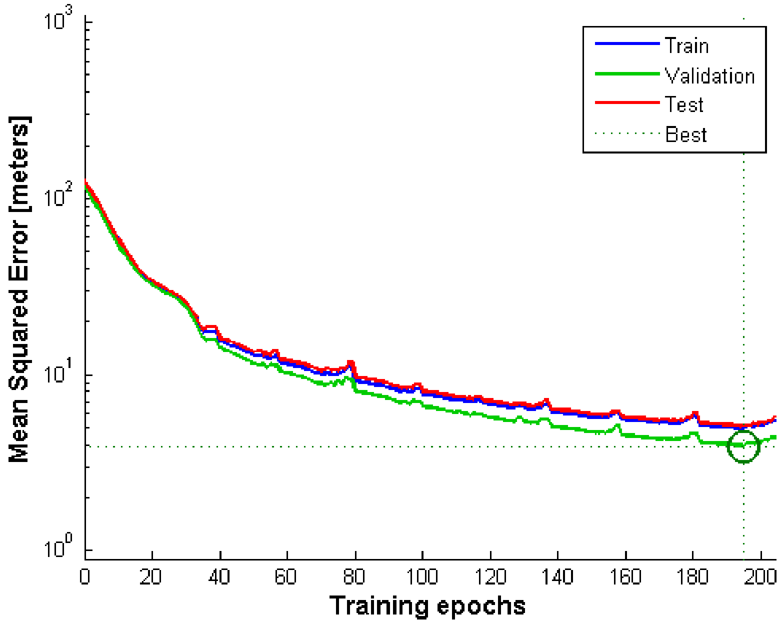

Fig. 28: MSE for best train, validation and test performances

Fig. 28 shows how many epochs (cycles of the training) are needed to get the minimum validation MSE (20) of 6.1507 meters which is significantly lower than the validation MSE value 11.8746 obtained by the previous not optimized algorithm. The network does not evince any overfitting, described in 6.2.5, according to almost no difference between the train and the validation performance because they are close to each other. The more are the train and the validation lines far away from each other the more the network tends to be overfitted. Also can be seen that in this case the network training takes more time than the previous training process, 195 epochs, but on the other hand, more accurate results are obtained. The optimization method has its pros and cons. Again less epochs could be used to get results faster but at the cost of even worse results.

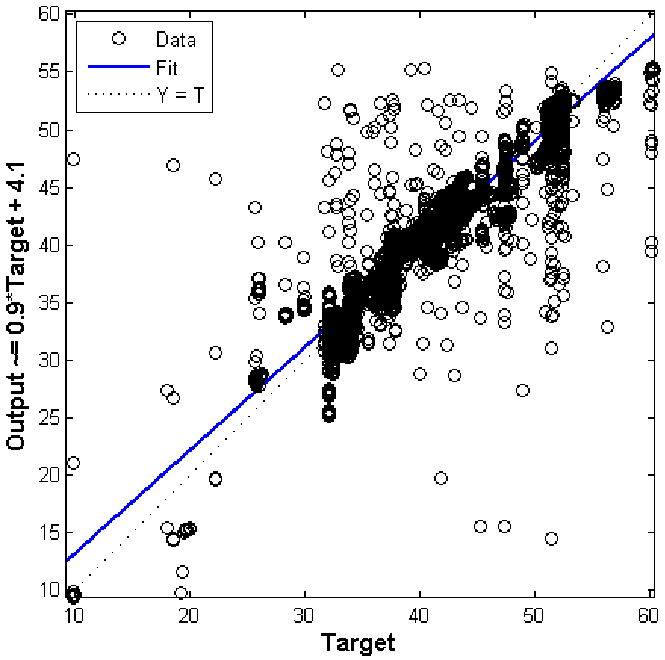

Fig. 29: Regression performance

Fig. 29 depicts the regression (17) toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (blue line). Fig. 29 also shows scattering of predicted real movement which can play a very significant role in terms of mobile users movement prediction use cases. In this case, the result is not enough to be used in real industry cases. For ideal prediction case, the blue line should copy the dotted line and the X_R parameter would have to be one. The result of the regression is 0.9413 which equals to approximately 94.1 % similarity between the target and the predicted output. Such similarity is more sufficient for movement prediction than in previous case.

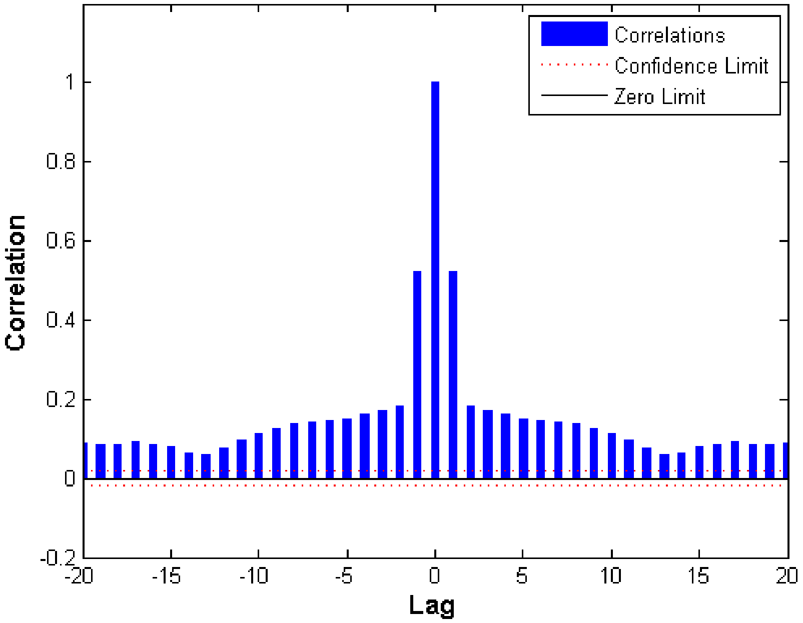

Fig. 30: Autocorrelation of errors

Fig. 30 describes how the predicted errors of the calculated output and the target are related in time (18). The value of 1 at zero lag indicates that the errors are the same if there is no time delay. Other non-zero lags evince some correlations that indicate some inaccuracies during the training process according to the non-zero blue bar lines leading to inaccurate prediction. In this case, the blue lagged bar lines are closer to the Confidence Limit that shows an improvement in usage of the optimization method described in 6.1.2.

6.4.3 Optimized Levenberg-Marquardt algorithm

In this chapter, the Levenberg-Marquardt algorithm (in MATLAB denoted as trainlm), described in 6.1.3, is evaluated. Again, the optimization method, described in 6.1.2, is applied in order to return as accurate results as possible. The Levenberg-Marquardt algorithm uses momentum learning instead of the learning rate. This method is predominantly useful for obtaining optimized results in larger data sets. The optimized parameters are highlighted in Tab. 5. These parameters in bold are selected because they affect the network results the most when they are changed.

Training function	**trainlm**		
Architecture parameters		**Data distribution**	
Input delays	1:2	Training data	70 %
Feedback delays	1:2	Testing data	15 %
Number of hidden layers	**10**	Validation data	15 %
Training parameters			
Maximum Epochs	**250**	Mu	**0.001**
Maximum Training Time	Inf	Mu Decrease Ratio	**0.1**
Performance Goal	0	Mu Increase Ratio	**5**
Minimum Gradient	1e-7	Maximum Mu	**1000**
Maximum Validation Checks	**6**		

Tab. 5: Configuration for Levenberg-Marquardt algorithm

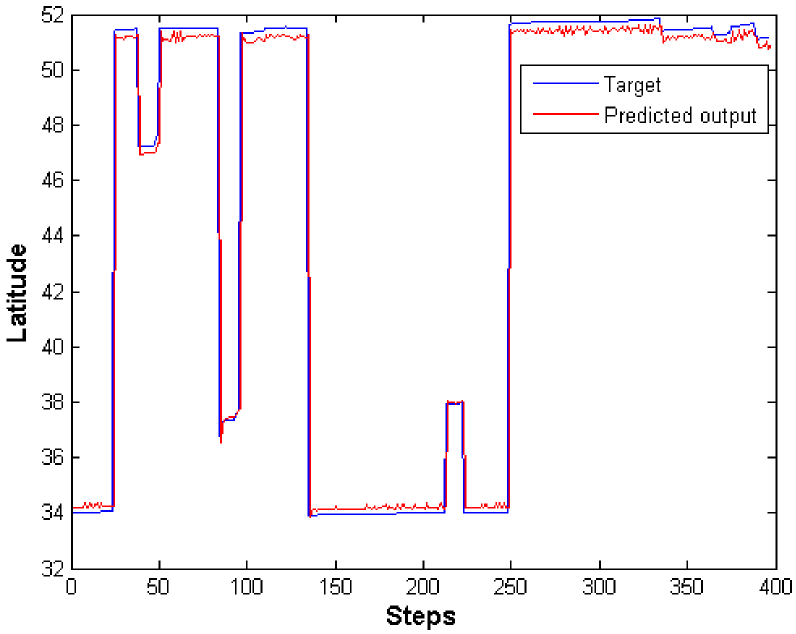

Fig. 31: Mobile users movement prediction in geolocation coordinates

Fig. 31 depicts 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 which would cause minimum problems with mobile user movement prediction. This result is sufficient enough to be used for real industry cases. The Levenberg-Marquardt algorithm is obviously able to manage the gradient vanishing problem (2), (3) and is suitable for this type of prediction tasks. In other words, the mobile users movement prediction is very accurately predicted.

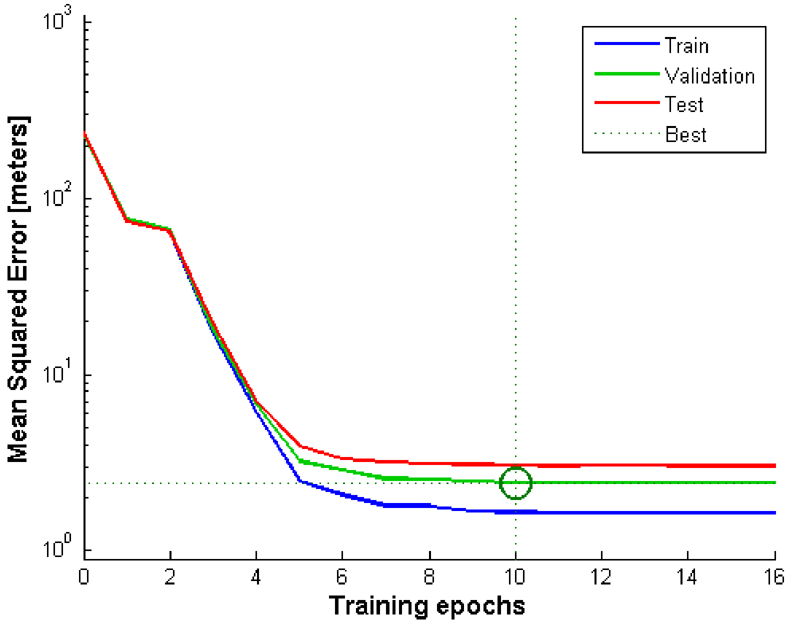

Fig. 32: MSE for best train, validation and test performances

Fig. 32 shows how many epochs (cycles of the training) are needed to get the minimum validation MSE (20) of 1.8306 meters. The network does not evince any overfitting, described in 6.2.5, according to almost no difference between the train and the validation performance because they are close to each other. The more are the train and the validation lines far away from each other the more the network tends to be overfitted. It is very close to the ideal zero performance value. This algorithm obviously outperforms the Gradient descent with adaptive learning algorithm as seen in final discussion in 6.4.4. Also can be seen that, in this case, the network training does not take much time for the training process, only 10 epochs, and what is more, the obtained results are very accurate even for real industry cases. Again less epochs could be used to get results faster but at the cost of even worse results and it could be counterproductive in this case.

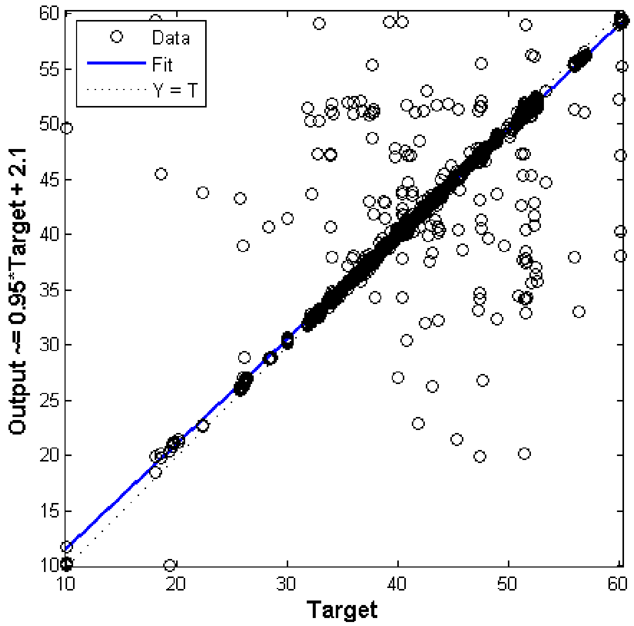

Fig. 33: Regression performance

Fig. 33 depicts the regression (17) toward the mean of the real movement (dotted line), the predicted values (black points) and the mean of the predicted values (blue line). Fig. 33 also shows scattering of predicted real movement which can play a very significant role in terms of mobile users movement prediction use cases. In this case, the result shows only few deviations from the mean line which indicates the suitability of this algorithm to be used in real industry cases. For ideal prediction case, the blue line should copy the dotted line and the X_R parameter would have to be one. The result of the regression is 0.9771 that means 97.7% similarity between the target and the predicted output. That is, without any doubt, considered a perfect result and in combination with conventional statistical methods the results accuracy could reach almost 100%.

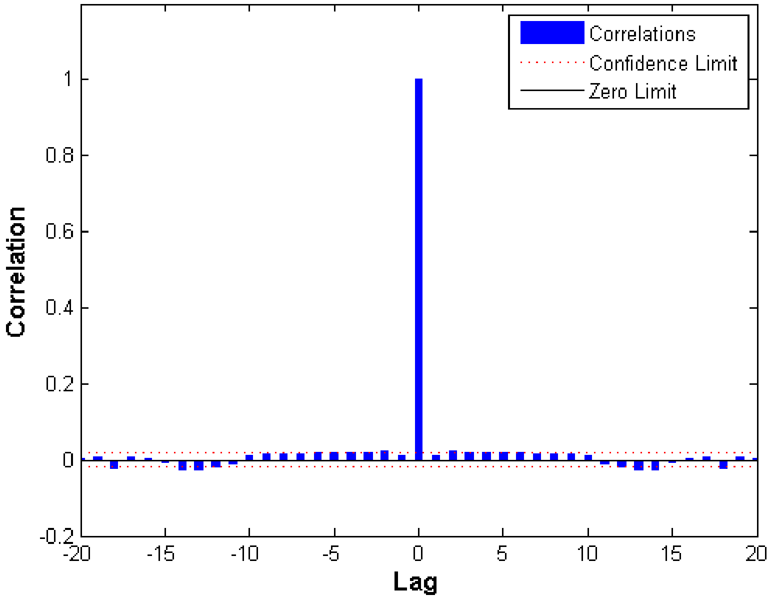

Fig. 34: Autocorrelation of errors

Fig. 34 depicts the correlated errors in time steps (blue bar lines) and the Confidence Limit line (dotted lines) showing limits inside which the prediction is almost perfect. Fig. 34 describes how the predicted errors of the calculated output and the target are related in time (18). The value 1 at zero lag indicates that the errors are the same if there is no time delay. Other non-zero lags evince some correlations that indicate very small inaccuracies during the training process according to the non-zero blue bar lines. In this case, the blue lagged bar lines are almost all inside the Confidence Limit that shows a perfect prediction in terms of usage of this algorithm.

6.4.4 Discussion

Three algorithms are used in this thesis. The Gradient descent with adaptive learning algorithm (GDA) in two variants (not optimized and optimized) and the Levenberg-Marquardt algorithm (LM) (optimized). The first algorithm does not evince sufficient result for mobile users movement prediction at all. The second algorithm can predict with the accuracy of around 94% which is enough but only for some types of use cases. Use cases that need accuracy of prediction around 1-2 meters or less require results with smaller scattering of the predicted values. The third one, the Levenberg-Marquardt algorithm, is dominant for mobile users movement prediction in telecommunications network in all aspects and even for wider type of prediction tasks [56]. To sum up all the results, see Tab. 6.

	Results discussion		
	GDA not optimized	GDA optimized	LM optimized
Train MSE	10.8812	5.2873	**1.9919**
Validation MSE	11.8746	6.1507	**1.8306**
Test MSE	11.1659	6.0036	**2.0430**
Regression	0.8704	0.9413	**0.9771**
Number of epochs	76	195	**10**

Tab. 6: Results discussion

Tab. 6 shows differences between the used algorithms. Performance results displayed in the table show the efficiency of the train, validation and test process respectively in meters. The GDA not optimized algorithm has the worst results because the algorithm itself is not the best choice for mobile users movement prediction even though it is generally used for prediction tasks as it has shown up during the mobile user movement prediction simulation. The training takes 76 epochs to obtain the results, the regression is around 87% and the overall performance of the MSE is not sufficient for industry usage. The GDA optimized algorithm outperforms the not-optimized GDA in terms of the overall performance of MSE and the regression about 94% thanks to the optimization method for finding the best possible combination of the network parameters. On the other hand, after the optimization the optimized GDA network needs more time to get trained, 195 epochs. Finally, the optimized LM algorithm outperforms both previous algorithms in terms of MSE (around 2) and regression (more than 97%), the values highlighted in bold, which indicates the best results and even in the shortest time. Only 10 epochs are needed to obtain results. This solution could be used for real industry cases.

7 CONCLUSIONS

In this thesis, an overview of big data is introduced in connection with the telecommunications industry. This work also shows a software and hardware concept used for big data that obtains data from telecommunications probes. Big data solution offers deeper insight to customer’s behavior in telecommunications network and quality of service through analytics tools. The work also gives a look at customer needs of the current market and comprehensive use cases summary for telecommunications industry.

In this thesis, common prediction using neural NARX neural network and its optimization for mobility prediction has been also analyzed. In the first step the Gradient descent with adaptive learning algorithm has been investigated in order to analyze its suitability to predict user mobile movement. The results show that this type of algorithm is not serviceable even if the final regression is around 87% similarity between the target and the calculated output. It could lead to considerable problems in built-up areas and prediction would be inaccurate. The system average train, validation and test MSE equals to 11.3 which supports the assumption of overall poor performance.

Further, optimization of the Gradient descent with adaptive learning algorithm has been proposed. The optimization consist of iterative selection of parameters for the neural network. This approach improves MSE to 5.8, which is more acceptable overall system performance for real 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 by optimized algorithms grows up to almost 94% that would be enough for some specific use cases.

The final algorithm, the Optimized Levenberg-Marquardt algorithm, is perfectly convenient for mobile users movement prediction. This algorithm outperforms the previous solutions in all aspects. The system average MSE equals to 1.95 which supports the fact that the high accuracy performance of the prediction reaches almost 98%. In addition, this accuracy is obtained very fast, the network needs only around 8 epochs to be completely trained. These results are based on the second order derivation character and the momentum learning process. It assures high quality optimization and in terms of the Levenberg-Marquardt algorithm there is not a lot of space for improving the whole solution with this data set.

In the future, more algorithms for prediction experiment and the results could be combined with classical conventional 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.

8 REFERENCES

[1] A. R. Simon, "Chapter 1 - What's in a Data Warehouse?".*Data Warehousing for Dummies*. John Wiley & Sons, © 1997.Books24x7. Web. Feb. 12, 2014.

[2] C. Eaton, D. Deroos, T. Deutsch, G. Lapis and P.C. Zikopoulos, Understanding Big data: Analytics for Enterprise Class Hadoop and Streaming Data, Mc Graw-Hill Companies, 978-0-07-179053-6, 2012.

[3] R. Weiss and L.J. Zgorski, "Obama Administration Unveils “Big data” Initiative: Announces $200 Million in new R&D Investments", Office of Science and Technology Policy Executive Office of the President, March 2012.

[4] P. Warden, “Big data Glossary – A Guide to the New Generation of Data Tools”, O’Reilly Media, Inc., 2011.

[5] Y. Li and S. Manoharan, “A performance comparison of SQL and NoSQL Databases”, Department of Computer Science University of Auckland New Zealand, IEEE, 2013.

[6] S. Sagiroglu and D. Sinanc, “Big Data: A Review”, Gazi University, Department of Computer Engineering, Ankara, Turkey, IEEE, 2013.

[7] R. Smolan, J. Erwitt, “The Human Face of Big data”, Hardcore, November 20, 2012.

[8] V. Borkar, M.J. Carey and C. Li, "Inside “Big data Management”: Ogres, Onions, or Parfaits?”, EDBT/ICDT 2012 Joint Conference Berlin Germany, 2012.

[9] S. Singh and N. Singh, "Big data Analytics", 2012 International Conference on Communication, Information & Computing Technology Mumbai India, IEEE, October 2011.

[10] S. Rosenbush and M. Totty (2013, March 10). The Evolution of Big Data in U.S. Corporations [online]. Available: http://whatsthebigdata.com/2013/03/10/the-evolution-of-big-data-in-u-s-corporations/.

[11] A. Katal, M. Wazid, R. H. Goudar, “Big Data: Issues, Challenges, Tools and Good Practices”, IEEE, 2013.

[12] S. Singh and N. Singh, "Big Data Analytics", 2012 International Conference on Communication, Information & Computing Technology Mumbai India, IEEE, October 2011.

[13] R.D. Schneider, “Hadoop for Dummies Special Edition”, John Wiley&Sons Canada, 2012.

[14] A.B. Patel, M. Birla, U. Nair, “Addressing big data problem using Hadoop and Map Reduce”, Engineering (NUiCONE), Nirma University International Conference, ISBN: 978-1-4673-1720-7, IEEE, December 2012.

[15] H. C. Yang, A. Dasdan, R. L. Hsiao, and D. S. Parker from Yahoo and UCLA, "Map-Reduce-Merge: Simplified Data Processing on Large Clusters", paper published in Proc. of ACM SIGMOD, pp. 1029–1040, 2007.

[16] D. Sindol (2014, June 20). Big Data Basics - Part 6 - Related Apache Projects in Hadoop Ecosystem [online]. Available: http://www.mssqltips.com/sqlservertip/3262/big-data-basics--part-6--related-apache-projects-in-hadoop-ecosystem/.

[17] Webpage image [online]. Available: http://ubolonton.blogspot.co.uk/2010/05/my-understanding-of-mapreduce.html.

[18] IBM sources, confidential information.

[19] Teradata White paper 02.13 EB 6705, “Teradata White paper 02.13 EB 6705”, Produced in USA, 2013.

[20] Teradata Data Warehousing 02.14 EB 3025, “Teradata solution technical overview”, printed in USA, 2014.

[21] Teradata. Next Generation Analytics with Teradata Aster Discovery Platform [online]. Available: http://www.teradata.com/Resources/Videos/Next-Generation-Analytics-with-Teradata-Aster-Discovery-Platform/?LangType=1033&LangSelect=true.

[22] Oracle, “In search of insight and foresight, Getting more out of big data”, A report from the Economist Intelligence Unit Limited, 2013.

[23] J. P. Dijcks, “Oracle: Big Data for the Enterprise”, USA, September 2014.

[24] SAP. Big Data Solution [online]. Available: http://www.sap.com/cz/solution/big-data.html.

[25] SAP. More Effective Cancer Therapies – with SAP HANA [online]. Available: http://www.sap-tv.com/video/#/29423.

[26] S. Hauser (2014, March 20). Data-driven insights [online]. Available: http://www.microsoft.com/enterprise/it-trends/big-data/articles/data-driven-insights.aspx#fbid=4OPb3tNdAZL.

[27] R. Lievano (2014, June 10). From Data Exhaust to Data Insight—the Journey ahead for Telcos [online]. Available: http://www.microsoft.com/enterprise/it-trends/big-data/articles/from-data-exhaust-to-data-insights.aspx#fbid=4OPb3tNdAZL.

[28] M. J. Foley (2013, October 28). Microsoft makes available its Azure-based Hadoop service [online]. Available: http://www.zdnet.com/article/microsoft-makes-available-its-azure-based-hadoop-service/.

[29] Microsoft. Reporting Services (SSRS) [online]. Available: http://msdn.microsoft.com/en-us/library/ms159106.aspx.

[30] Microsoft. Analysis Services [online]. Available: http://msdn.microsoft.com/en-us/library/bb522607.aspx.

[31] IBM sources, confidential information.

[32] IBM sources, confidential information.

[33] IBM sources, confidential information.

[34] IBM sources, confidential information.

[35] IDC company (2015). Unlocking Big Data, Business Values, Drivers, and Challenges [online]. Available: https://www.unlockingbigdata.com/industries.cfm.

[36] T. S. Lim, W. Y. Loh, Y. S. Shih, “A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms”, Kluwer Academic Publishers, Boston, Machine Learning, pp. 203–229, 2000.

[37] Doc. RNDr. I. Mrázová, CSc., “Dobývání znalostí”, Katedra teoretické informatiky, Matematicko-fyzikální fakulta, Univerzity Karlovy v Praze. Available: ksvi.mff.cuni.cz/~mraz/datamining/lecture/Dobyvani_Znalosti_Prednaska_Uvod.pdf.

[38] prof. Ing. J. Tučková, CSc. Algoritmy a struktury neuropočítačů [online]. Available: http://amber.feld.cvut.cz/ssc/ssc-p/asnP1_12.ppt.

[39] prof. Ing. J. Tučková, CSc. Algoritmy a struktury neuropočítačů [online]. Available: http://amber.feld.cvut.cz/ssc/ssc-p/asnP2_12.ppt.

[40] R. Rojas. Neural Networks. Springer, 1996.

[41] S. Haykin, “Neural_Networks_-_A_Comprehensive_Foundation”, McMaster University, Ontario Canada, Pearson Education Singapure Pte. Ltd., 1999.

[42] B. Karlik and A. V. Olgac, “Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks”, International Journal of Artificial Intelligence And Expert Systems (IJAE), vol. 1, issue 4.

[43] MathWorks, Neural Networks Overview [online]. Available: http://www.mathworks.com/help/nnet/gs/neural-networks-overview.html.

[44] Z. Li, X. Li, N. V. Narasimhan, A. Nayak, I. Stojmenovic, “Autoregression Models for Trust Management in Wireless Ad Hoc Networks”, ISBN: 978-1-4244-9266-4, pp. 1-5, Global Telecommunications Conference (GLOBECOM 2011), IEEE, 2011.

[45] MathWorks. How dynamic neural networks work [online]. Available: http://www.mathworks.com/help/nnet/ug/how-dynamic-neural-networks-work.html.

[46] E. Diaconescu, PhD. “The use of NARX Neural Networks to predict Chaotic Time Series”, University of Pitesti, ISSN: 1991-8755 191, vol. 3, issue 3, March 2008.

[47] T. Lin, C. L. Giles, B. G. Horne, S.Y. Kung, “A Delay Damage Model Selection Algorithm for NARX Neural Networks”, *IEEE Transactions on Signal Processing, “Special Issue on Neural Networks”*, vol. 45, no. 11, pp. 2719-2730, 1997.

[48] 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.

[49] D. Mishra, A. Yadav, S. Ray, and P. K. Kalra, “Levenberg-Marquardt Learning Algorithm for Integrate-and-Fire Neuron Model”, Indian Institute of Technology, Kanpur, India, vol.9, no.2, November 2005.

[50] S. Akoush, A. Sameh, “Movement Prediction Using Bayesian Learning for Neural Networks”, ISBN: 0-7695-2938-0, Systems and Networks Communications. Second International Conference, IEEE, 2007.

[51] M. Ficek, L. Kencl, "Spatial extension of the Reality Mining Dataset," Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference, pp.666-673, 8-12 Nov. 2010 doi: 10.1109/MASS.2010.5663788.

[52] F. Meneses, A. Moreira, “Using GSM CellID Positioning for Place Discovering”, ISBN: 1-4244-1085-1, Pervasive Health Conference and Workshops, IEEE, 2006.

[53] M. T. Hagan, O. D. Jesus, R. Schultz, “Training Recurrent Networks for Filtering and Control”, in (editors) L.R. Medsker, L.C. Jain, CRC Press, 2001.

[54] MathWorks. Nonlinear Regression [online]. Available: http://www.mathworks.com/discovery/nonlinear-regression.html.

[55] MathWorks. Autocorrelation function [online]. Available: http://www.mathworks.com/help/econ/autocorr.html#btzjb3t

[56] A. Tsakonas, N. Ampazis and G. Dounias, “Towards a Comprehensible and Accurate Credit Management Model: Application of Four Computational Intelligence Methodologies”, University of the Aegean

[57] M. T. Koné, PhD. The future of human evolution [online]. Available: http://futurehumanevolution.com/artificial-intelligence-future-human-evolution/artificial-neural-networks.

[58] E. Alba and J. F. Chicano, “Training Neural Networks with GA Hybrid Algorithms”, ISBN: 978-3-540-24854-5, ISSN: 0302-9743, Genetic and Evolutionary Computation – GECCO, pp 852-863, 2004.

[59] S. Srihari, “The Hessian Matrix”, *Machine Learning* [online]. Available: http://www.cedar.buffalo.edu/~srihari/CSE574/Chap5/Chap5.4-Hessian.pdf