Master Thesis in Computer Science
Jan 2020

Predicting User Mobility using Deep Learning Methods

Harihara Reddy Pamuluri

Dept. Computer Science & Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Department of Computer Science & Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full-time studies.

Contact Information:
Author:
Harihara Reddy Pamuluri
E-mail: paha17@student.bth.se

University advisor:
Hüseyin Kusetoğullari
Dept. Computer Science & Engineering

Dept. Computer Science & Engineering	Internet	:	www.bth.se/didd
Blekinge Institute of Technology	Phone	:	+46 455 38 50 00
SE–371 79 Karlskrona, Sweden	Fax	:	+46 455 38 50 57

Abstract

Context. The context of this thesis to predict user mobility using deep learning algorithms which can increase the quality of service for the users and reduce the cost of paging for telecom carriers.

Objectives. This study first investigates to find the suitable deep learning algorithms that can be used to predict user mobility and then an experiment is performed with the chosen algorithms as a global model and individual model then evaluate the performance of algorithms.

Methods.Firstly, a Literature review is used to find suitable deep learning algorithms and then based on the finding an experiment is performed to evaluate the chosen deep learning algorithms.

Results. Results from the literature review show that the RNN, LSTM, and variants of the LSTM are the suitable deep learning algorithms. The models are evaluated with metrics accuracy. The results from the experiment showed that the individual model gives better performance in predicting user mobility when compared to the global model.

Conclusions. From the results obtained from the experiment, it can be concluded that the individual model is the technique of choice in predicting user mobility.

Keywords: Deep learning, Mobility Prediction, Time series Forecasting, Long Short-Term Memory

Acknowledgments

I would first like to thank my industrial supervisors Mr. Jaeseong Jeong and Mrs. Mona Matti at Ericsson, for giving me this opportunity. They allowed this paper to be my own work and steered me in the right direction by providing constructive guidance consistently.

It is a pleasure for me to express my special thanks and gratitude to my supervisor Hüseyin Kusetoğullari, for his expertise, remarkable supervision, and continuous support. This thesis would not have been possible without his motivation and guidance.

I would also like to thank my family and friends for their unconditional love and support throughout the research.

List of Figures

List of Tables

List of Abbreviations

AI -Artificial Intelligence
ANN - Artificial Neural Network
LSTM - Long Short-Term Memory
GRU - Gated Recurrent Unit
ML - Machine Learning
DL - Deep Learning
RNN - Recurrent Neural Networks
TSF - Time-series Forecasting
UE - User Equipment
5G - Fifth Generation
IoT - Internet of Things
MM - Mobility Management
CDR - Call Details Record
GPS - Global Positioning System
IMSI - International Mobile Subscriber Identity
eNB - eNodeB or Evolved Node B
HMM - Hidden Markov Model
MMM - Mixed Markov Model
EBM - Event-Based Monitoring

Contents

Chapter 1

Introduction

There is a massive increase in the usage of smartphones and IoT devices, which are constantly connected to the networks through CDRs, data traffic. As these devices are always connected to the networks, the telecom industry has access to a huge amount of data, which is used to improve the quality of service for the users [1]. As there is no satisfaction with the quality of service, there is always an increase in demand of the users with mobile communication technology. User's movement patterns are used by the researchers to improve the conditions of communication, services to give better performance [2]. Large volumes of mobility data like Call Detail Records (CDRs), data traffic of mobile networks, tracking Global Positioning System (GPS), Wi-Fi data access points are generated by mobile phones (IMSI), which can be sensed by people movements. To study human dynamics and provide trajectories of people on a large scale, researchers found that data traffic from cellular networks 2G/3G/4G/5G is highly useful. The advantages like consumption of low energy, wide range a large number of people being covered, high-cost efficiency can be obtained by collecting human trajectories with smartphones using the internet [3]. In wireless networks, Mobility prediction is used in managing the bandwidth resources and efficient planning [4]. Mobility prediction is utilized by mobile and wireless networks, which focuses on performing effective resource management of networks and to predict the user's future location [5].

5G stands for fifth-generation cellular network technology, with lower latency, higher bit rates and is 100-1000 times faster than the networks that currently available. The telecom carriers or the network operators have already started the 5G network and services, but the individual and industries adopt this technology in 2020 due to the lack of supporting devices [6]. As per the research, 18 billion IoT devices are to be expected by 2022 [7].

1.1 Description of the problem

Due to the increase in the number of networks connecting devices the telecom carriers are facing a challenge in improving the quality of service for the users. It is very important for the telecom carriers to provide uninterrupted network access with a faster network to the users. The mobility of the user plays an important role in improving the service. Therefore, this thesis deals with neural networks to predict the mobility of the UE, which helps the carriers to increase or decrease the bandwidth of the base station depends on the users existing under that base station.

A base station has a limit of UEs handling at a time. If the UE count increases the handling capacity of the base station, then the service is interrupted (call dropping, slow data connectivity). In 5G, this issue can be solved remotely by increasing the handling capacity of the base station. In the same way, if the user count is lower than the handling capacity of the base station, the handling capacity can be decreased, which helps the telecom carries to reduce the cost of maintenance

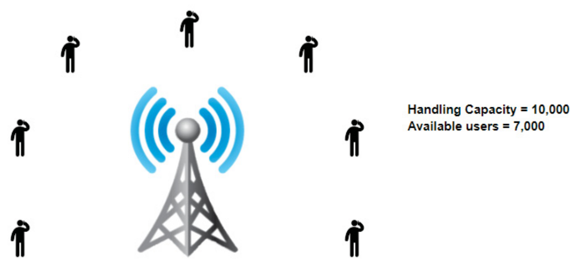


Figure 1.1: Handling capacity of base station is more than available users

From Figure 1.1 , the handling capacity of the base station at a time is 10000. In this case, there are only 7000 users who are active under the base station. There is no interruption in the service for the users.

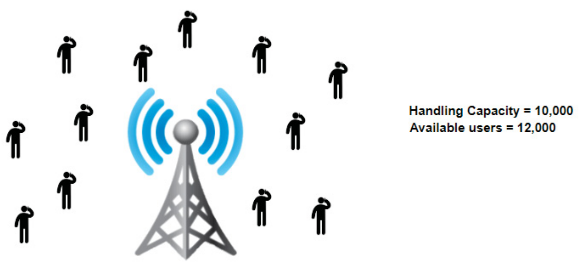


Figure 1.2: Handling capacity of base station is less than available users

From Figure 1.2 , the number of active users under the base station is more than the handling capacity of the base station. In this case, the quality of the service is compromised and users of the telecom carriers faces service interruption (call dropping, slow mobile data connectivity). If the telecom carries can predict the increase in user count upfront, the preconfiguration of the service is adapted and issue can be solved and quality of the service can be improved as shown in Figure 1.3.

Figure 1.3: Sloved by preconfiguring the base station

Paging:

An electromagnetic wave is sent to the base station from the antenna when UE1 makes a call to connect UE2. That electromagnetic wave is converted to high-frequency light pulses by the antennas sent from UE1 searches for UE2 by sending an alert signal from all the antenna/base stations is known as paging. If the telecom carriers can predict the current location of UE2, the light pulse is then sent directly to the destination antenna/base station which then converts to electromagnetic waves by the antenna and connects UE 1 call to UE 2. Predicting the location of the users helps the telecom carriers in reduces the cost of paging an improve the quality of service by making the connectivity faster between the users. The problem is explained in detail with the figures below.

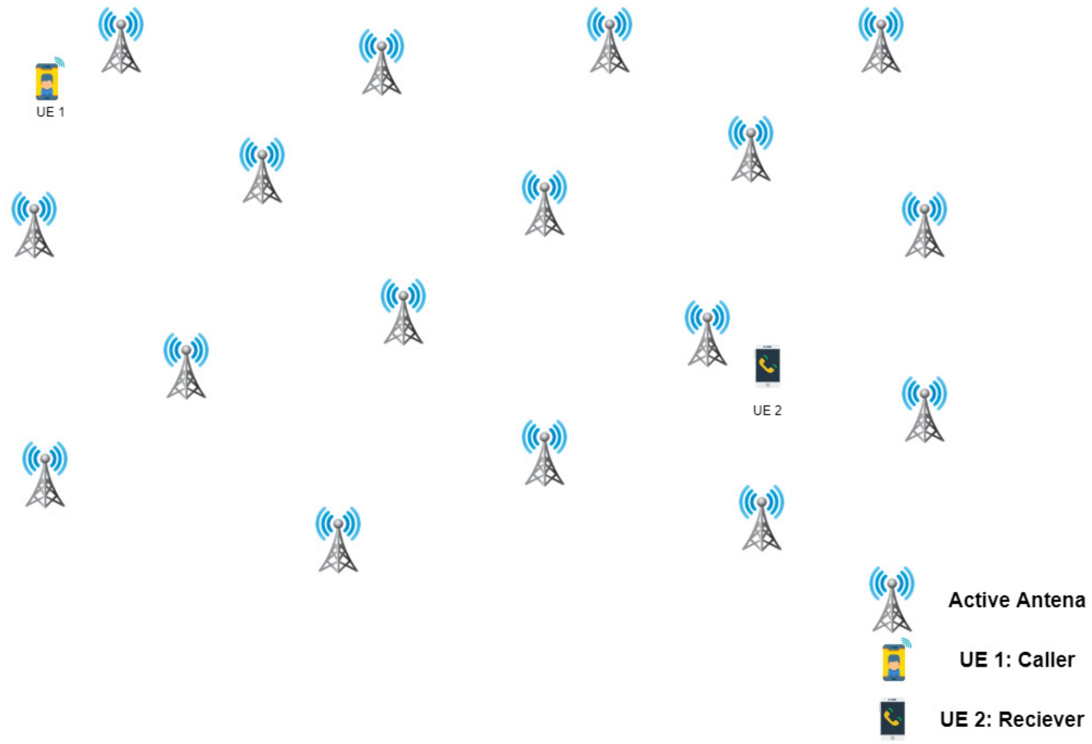


Figure 1.4: Paging Problem

From the above figure, UE1 is the caller and UE 2 is the receiver. When UE1 makes a call to UE2 an electromagnetic wave is sent from UE1 to the antenna located near that UE. If UE2 is active, then the signal is sent directly to the destination antenna where UE 2 is located. If UE2 is not active then the light pulse is to all the antenna where UE2 was active in the past and the tracking area (Tracking area is a group of antennas in a location) antennas of the past locations to find UE2, which costs the telecom carries. This issue can be solved by predicting the UE's mobility.

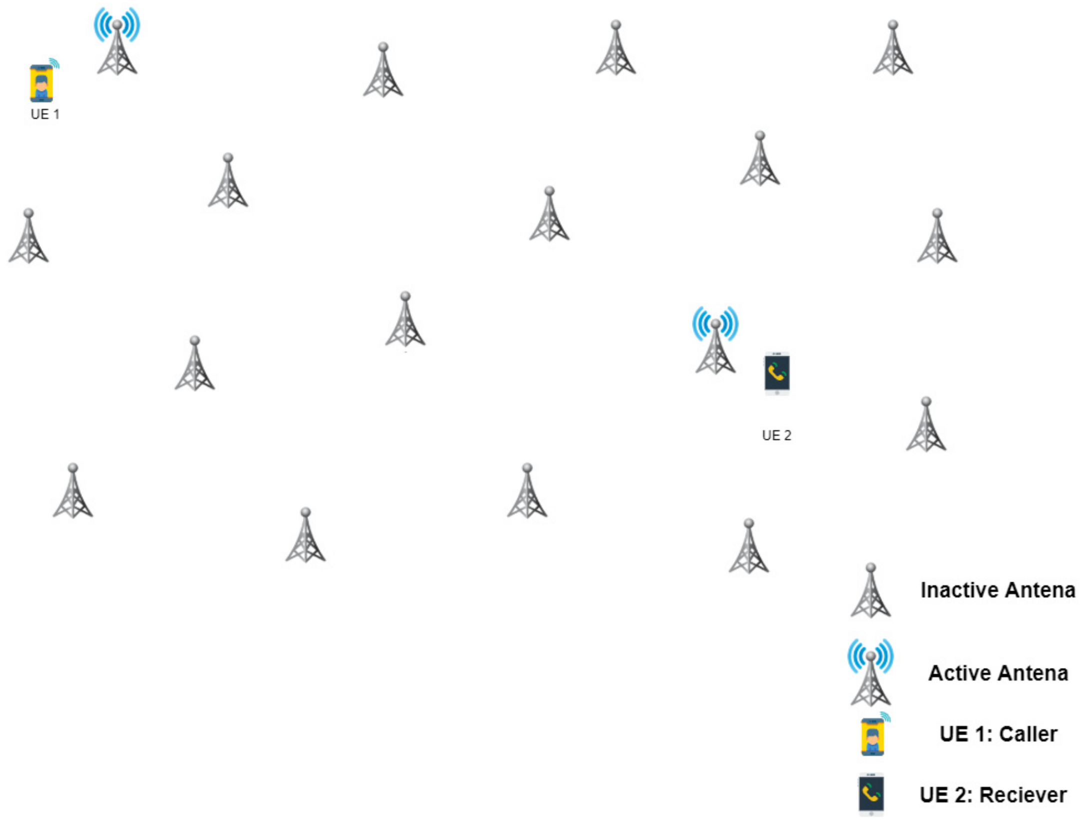


Figure 1.5: Solution for paging

From the above figure, it is shown that the electromagnetic wave sent from UE1 is received directly by UE2 from the antenna where UE2 is located. The UE2 located antenna is active and all other antennas are inactive for that call. This can reduce the cost of paging for the telecom carriers and improve the quality of service for the users.

1.2 Aim

The main aim of the thesis is to find suitable deep learning models to predict the next 96 eNB's based on the available historical data. And, to find which technique, either global model or individual model for each IMSI gives better prediction accuracy.

1.3 Objectives

The below objectives are formulated to achieve the aim.

1. Understanding deep learning models in time series forecasting.
2. Identifying the algorithms that are suitable.
3. Analyzing the results of the global model and the individual model.

1.4 Research Questions

Research Questions:

RQ 1: What are the deep learning algorithms that are suitable to forecast the time series data?
Motivation: The motivation behind formulating this research question is to find suitable Deep learning algorithms to predict the next n eNB's.

RQ 2: How is the performance of the deep learning models vary for the global model and individual model in predicting the next n eNB's?
Motivation: The motivation behind formulating this research question is to evaluate the performance of the selected deep learning algorithms as a global model and as an individual model for each IMSI.

1.5 Structure of the thesis

Chapter 1 discusses the introduction, description of problem, aim, objectives and research questions of the thesis.
Chapter 2 presents the background of this thesis and also discusses the related work and research done in time series forecasting, mobility prediction, and neural networks.
Chapter 3 explains the time series forecasting in detail.
Chapter 4 explains the research methods used in the research, followed by models used in the experimental method.
Chapter 5 presents the results obtained from the research methods.
Chapter 6 covers analyzing the results and discussing them.
Chapter 7 deals with the conclusion and future work of the thesis.

Chapter 2

Background & Related Work

2.1 Artificial Intelligence

2.1.1 Artificial Intelligence

In the early 1950s, A famous mathematician Alan Turing proposed a question "Can Machines Think?" in his research paper "Computing Machinery and Intelligence" and it became the basic goal foundation for Artificial Intelligence [8].

Artificial Intelligence is the area of computer science which deals with machine intelligence. As everyone knows that the computer solves the problems based on the instructions given by humans in the form of programs but the ability of the computer intelligence to learn itself to solve problems is called artificial intelligence [9]. In other words, the simulation of human intelligence processes by machines is called artificial intelligence [10].

2.1.2 Machine Learning

Arthur Lee Samuel is an American pioneer in the artificial intelligence computer gaming field. In the year 1959, In one of his research papers, he defines machine learning as "The field of science that gives computers the ability to learn without being explicitly programmed" [11]. In simple words, machine learning is the subset of artificial intelligence that gives the ability to learn improve automatically from the previous experience without programmed explicitly. It targets the computer algorithms that access the data to learn themselves [12].

Machine learning algorithms are classified into 3 categories. They are:

1. **Supervised Learning:** The system in which both input data, output data are provided and labeled for classification to give learning features to further data processing. The term supervised learning is derived from the idea of the algorithm is learning from the trained data [13].

   Supervised Learning is divided into 2 categories. They are:

   (a) **Classification:** It predicts when the output data belongs to the specific category. Example: Classifying whether the color is "RED" or "BLUE".

   (b) **Regression:** : It predicts when the output data belongs to real value. Example: predicting the stock market price.

2. **Unsupervised Learning:** The system in which only input data is provided but not the output labeled data is Unsupervised learning. The algorithm must find the best insights from the input data for further processing [13]. Unsupervised Learning is divided into 2 categories. They are:

   (a) **Clustering:** Finding inherent groupings in the data is called Clustering. Example: grouping customers based on the products they bought.

   (b) **Association:** Discovering rules that describe large portion of data is called association rule. Example: The people buy "Milk" tends to buy "Bread".

3. **Reinforcement Learning:** Reinforcement Learning target on taking the actions according to the situation to maximize the reward [14]. In supervised learning, the data consists of answer key pairs so that the model is trained with correct answers but in reinforcement learning answer key pairs are not provided and it performs the tasks by learning itself from the previous experience [14]. Example: we can consider chess game as an example the reinforcement algorithm decided which optimal move should take based on the previous moves to get maximum reward.

2.1.3 Deep Learning:

Deep Learning is a subset of machine learning that train computers like how human intelligence processed the data to find insights to make business decisions. It follows neural network architecture hence deep learning algorithms are also called deep neural networks [15]. The concept of deep learning is from 1980s, but nowadays deep learning is attracting and growing rapidly because of following reasons:

1. Deep Learning needs a distinctive amount of labeled data [16].

2. Higher accuracy has been observed when compared to human intelligence [16].

3. Although the training time has been reduced due to higher GPU's and Cloud Computing clusters yet demands higher computing power [16].

Examples of Deep Learning implemented areas are Autonomous Driving, Aerospace, etc.

Neural Network:

According to the Dr. Robert Hecht-Nielsen neural network is defined as a "computing system consisting of a number of highly interconnected processing elements, which can process the information by their dynamic state response to external inputs" [17].

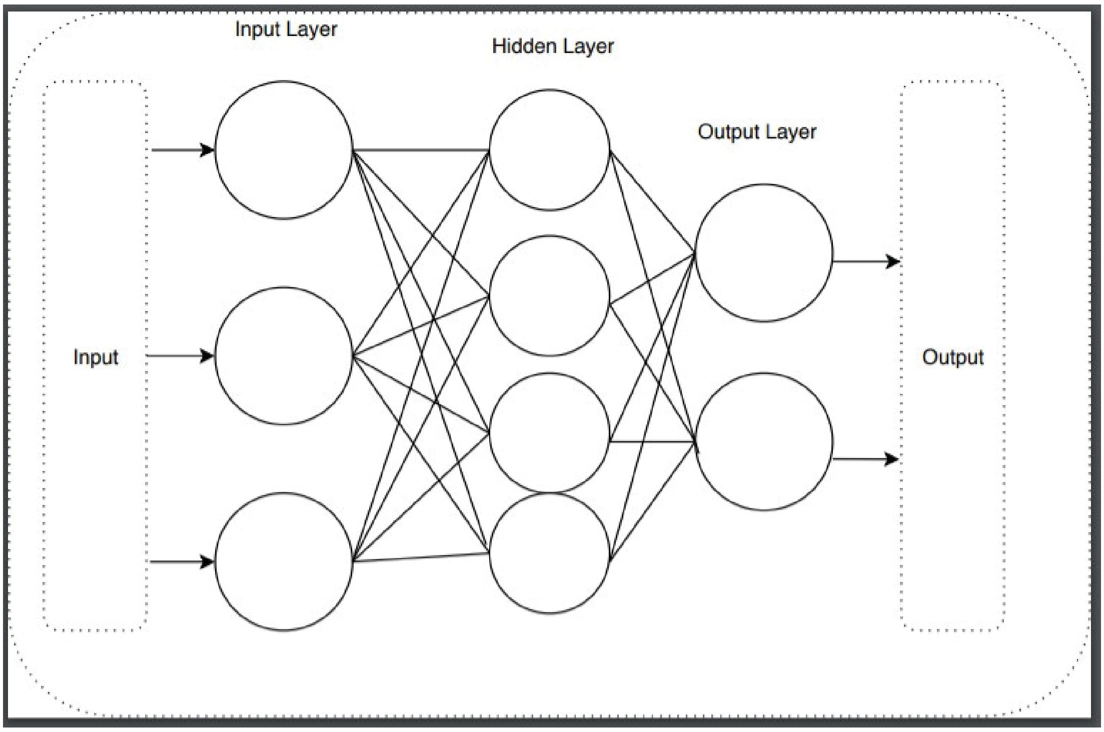


Figure 2.1: Architecture of Neural Network

Input Layer: It will transfer the input data to the next layers [18].
Hidden Layer: All the calculations are done in this layer and response is transferred to the following layers [18].
Output Layer: Activation function is used in this layer and it is responsible for convert the output response to appropriate format [18].

2.2 Related Work

Live location of people can easily be identified with a great rate of localization accuracy and by developing GPS and other technologies that are used for positioning we get greater sample rate. To identify locations, we can use the location history shared by social networking applications which are location-based. With the wide usage of location-based social networking applications throughout the globe, this research on prediction of the location has been moving forward with greater precision and more feasibility. Location of people can be predicted using mobility pattern mining from trajectories history, this prediction can be done by gathering data of time, location and the current state of the people [19].

The author in the article [20] mobility Markov chain model to predict the next location of the user based on the n previously visited locations. The author proposed an algorithm known as n-MMC which tracks the n locations of the user's previous location data.

There were several types of research and studies conducted by many researchers to minimize location prediction problems. Traditional approaches mainly focus on the Markov Model [21], which use one- order or multiple-order state-transition matrix to model the mobility pattern of people. Multi-order Markov model possesses problems, there are possibilities to not get any recalls when any new state is not present in the transition matrix.

The recently proposed model for location prediction is the Mixed Markov chain Model (MMM) [21]. This Model states that standard Markov Models (MM) and Hidden Markov Models (HMM) are non- specific enough for all types of moving objects. As we already know how MM works, bundles multiple individuals into a set of groups depending on their traces of mobility and generates a Markov model for each group of individuals. The test results of this already conducted experiment have proven to be accurate and give us a prediction rate of 74.1% for MMM [22].

Some researchers were using neural network approaches to identify the problems with location prediction. Authors in the article [23] presented an STF-RNN model that can predict the next location of individuals using a recurrent neural network. The proposed recurrent model includes time and space interval sequences this is used to discover long-term dependencies this is helpful in improving the efficiency of the given model.

Chapter 3

Time Series Forecasting

Time series forecasting is defined as forecasting future observations based on historical data. It is mostly used in decision-making applications [19].

Time series data: In simple terms, time-series data can be defined as a sequence of data points in a successive order of the same thing collected over a time period and are stored in time order. Time series data consists of two major components, that is the time units and a numerical value for respective time unit. Time series data can be collected from any variable that changes over time. The data points in time series data are collected at regular intervals of time. In this time-series data, each time unit possesses at most one value at the same time.[20]

The advantage with the time series data is, it provides the required information for the researcher by not having any minimum or maximum time limitation that must be considered while collecting the data. This lead-in widespread use of time series data across many organizations as it allows to predict the future values with the help of previously collected (past values) data.

There are two ways to collect time-series data. Stationary time series and Non-stationary time series.

Time series data is classified into two types:
  Univariate time-series data
  Multivariate time-series data.

Univariate Time-series data: As the name suggests it contains an only one-time dependent variable. Univariate time-series data that consists of a single observation collected in a sequence over equal time increments.

Time	Temperature
5:00 am	59 °F
6:00 am	59 °F
7:00 am	58 °F
8:00 am	58 °F
9:00 am	60 °F

Figure 3.1: Univariate time-series data

Multivariate Time series data: If a time series data has more than one-time dependent variable it is called multivariate time series data. The variables are a multivariate time series data not only depends on its past values but also depends on other variables present in the data.

Time	Temperature	cloud cover	dew point	humidity	wind
5:00 am	59 °F	97%	51 °F	74%	8 mph SSE
6:00 am	59 °F	89%	51 °F	75%	8 mph SSE
7:00 am	58 °F	79%	51 °F	76%	7 mph SSE
8:00 am	58 °F	74%	51 °F	77%	7 mph S
9:00 am	60 °F	74%	51 °F	74%	7 mph S
10:00 am	62 °F	74%	52 °F	70%	8 mph S

Figure 3.2: Multivariate time-series data

Time series forecasting is classified into two types:
One-step ahead forecasting
Multi-step ahead forecasting.

One step ahead forecasting: Predict the next immediate future data from the historical data given by the predictor.

Multi-step ahead forecasting: Predicting not only one step but multiple steps ahead from the current data point based on the historical data.

Chapter 4

Methods

The following methods have been used to answer the research question formulated in this thesis. Initially, a literature review is conducted to study the existing algorithms for the time series forecasting. The selected algorithms are used in the experiment to evaluate the performance of the models.

4.1 Literature review

A literature review is selected as the research method to answer the first research question. The main aim of this method is to find the relevant and popular deep learning models that are used in time series forecasting. The results obtained from the literature review are further discussed in section 5.1:

The search string that is used to select the articles are:

1. Time series forecasting using Deep learning
2. Time series forecasting using LSTM
3. Predicting user mobility using Deep learning

The articles that are obtained from the above search strings are further filtered using inclusion and exclusion criteria.

- Articles that are in published English language.
- Articles that are published between 2010-Present (Related to the models selected for time series forecasting.)
- Articles that implement deep learning models on time-series data.
- Articles that have some key concepts definitions are selected even they are published before 2010

4.2 Experiment

4.2.1 Software Environment:

Python is a high level, general-purpose, scripting programming language created by Guido van Rossum in 1991 [24]. Python is considered as the most preferred programming language for machine learning as the syntaxes are simple and easy to learn [25]. Open source libraries like TensorFlow and Pandas help in developing machine learning applications and providing easier data analysis in Python respectively. Fast Experimentations in nueral networks is facilitated by Keras, an open source, high-level deep learning models API written in Python. Jupyter Notebook is an open source web application, which is used by the developers or researchers to create, share documents that have code, text and much more.

4.2.2 Hardware Environment:

The deep learning models were developed in HP EliteBook 840 G5 system consisting of Intel(R) Core (TM) i7-8650 processor @ 1.90GHz 2.1 GHz and 32GB RAM. The system is running on Windows 10 Enterprise 64-bit operating system.

4.2.3 Data-sets:

We conducted an experiment on two data sets. They are:

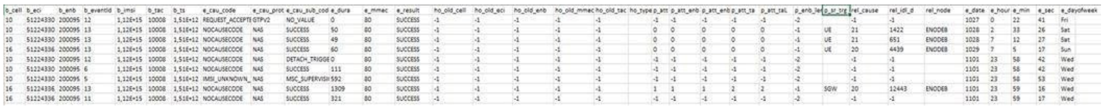
The datasets contain eNB id where the IMSI is located at a particular time. The raw data of the data-set contains 32 columns of data gathered from the telecom carriers through CDR that deals with multiple service-related issues like handover procedure, paging.

EBM1 data set:
Data set contains of 201910 IMSIs data from 2017-10-26 09:00:00 to 2017-11-03 06:00:00. The data set contains 1371 unique eNB's.

EBM2 data set:
Data set contains of 335959 IMSIs data from 2015-12-04 13:00:00 to 2015-12-10 23:45:00. The data set contains 8851 unique eNB's.

4.2.4 Data Preprocessing:

The raw data contains 32 columns of data. Once the raw data is extracted from the server, it is to be analyzed and the features that are required for the time series are to be selected.



b_cell	b_eci	b_enb	b_eventid	b_imsi	b_tac	b_ts	e_cau_code	e_cau_prot	e_cau_sub_cod	e_dura	e_mmec	e_result	ho_old_cell	ho_old_eci	ho_old_enb	ho_old_mmec	ho_old_tac	ho_type	p_att	p_att_enb	p_att_enb	p_att_ta	p_att_tal	p_enb_lei	p_sr_trg	rel_cause	rel_idl_d	rel_node	e_date	e_hour	e_min	e_sec	e_dayofweek
10	51224330	200095	12	1,12E+15	10008	1,51E+12	REQUEST_ACCEPTE	GTPv2	NO_VALUE	0	80	SUCCESS	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-2		-1	-1		1027	0	22	41	Fri
10	51224330	200095	13	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	50	80	SUCCESS	-1	-1	-1	-1	-1		0	0	0	0	0	-1	UE	21	1422	ENODEB	1028	2	33	26	Sat
10	51224330	200095	13	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	49	80	SUCCESS	-1	-1	-1	-1	-1		0	0	0	0	0	-1	UE	21	651	ENODEB	1028	7	12	27	Sat
16	51224336	200095	13	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	60	80	SUCCESS	-1	-1	-1	-1	-1		0	0	0	0	0	-1	UE	20	4439	ENODEB	1029	7	5	17	Sun
10	51224330	200095	12	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	DETACH_TRIGGE	0	80	SUCCESS	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-2		-1	-1		1101	23	58	42	Wed
10	51224330	200095	6	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	111	80	SUCCESS	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-2		-1	-1		1101	23	58	42	Wed
10	51224330	200095	5	1,12E+15	10008	1,51E+12	IMSI_UNKNOWN_	NAS	MSC_SUPERVISI	592	80	SUCCESS	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-2		-1	-1		1101	23	58	53	Wed
16	51224336	200095	13	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	1309	80	SUCCESS	-1	-1	-1	-1	-1		1	1	1	2	2	-1	SGW	20	12443	ENODEB	1101	23	59	16	Wed
16	51224336	200095	11	1,12E+15	10008	1,51E+12	NOCAUSECODE	NAS	SUCCESS	321	80	SUCCESS	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-2		-1	-1		1101	23	59	17	Wed

Figure 4.1: Raw data

From the above data, the features that are selected are:

1. **b_IMSI:** IMSI or International Mobile Subscriber Identity is a unique number that identifies the user equipment. In simple words, it is the id of the user equipment(UE) that uses the cellular network.
2. **b_enb:** eNB or eNodeB is known as Evolved Node B or E-UTRAN Node B. eNB is the hardware that connects the cellular networks which communicate with UE's wireless.
3. **e_hour:** Hour of the day (24 hours)
4. **e_date:** Date
5. **e_min:** Minutes
6. **e_sec:** Seconds

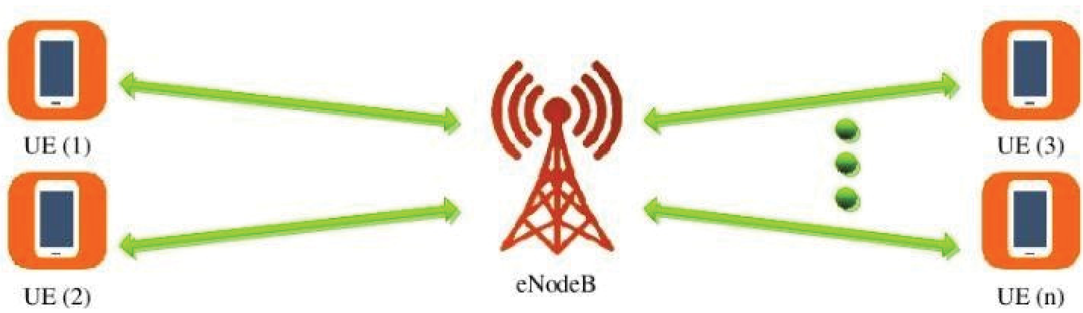


Figure 4.2: Picture representation of eNodeB and UE

4.2.5 Training and testing data-sets:

EBM1 Data-set:
The data obtained from the data sets are divided into training and testing data sets to train and test the model's performance. From EBM 1 data set, it contains a total of 755 time slots from 2017-10-26 09:00:00 to 2017-11-03 06:00:00 out of which 96 time slots from 2017-11-02 06:00:00 to 2017-11-03 06:00:00 are taken for testing data set and the remaining data set is taken for training the model.

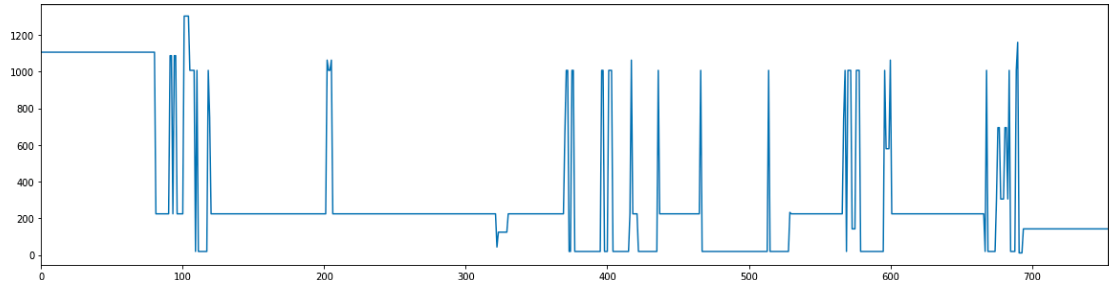


Figure 4.3: Example of 1 IMSI (274) from EBM1 data-set

In Figure 4.3, the plotted graph an example of user mobility. The X-axis is the total no of time slots and Y-axis is eNB id. The example IMSIs are taken for the EBM1 data set.

EBM2 Data-set:
The data obtained from the data sets are divided into training and testing data sets to train and test the model's performance. From EBM 2 data set, it contains a total of 621 time slots from 2015-12-04 13:00:00 to 2015-12-10 23:45:00 out of which 96 time slots from 2015-12-10 23:45:00 to 2015-12-09 23:45:00 are taken for testing data set and the remaining data set is taken for training the model.

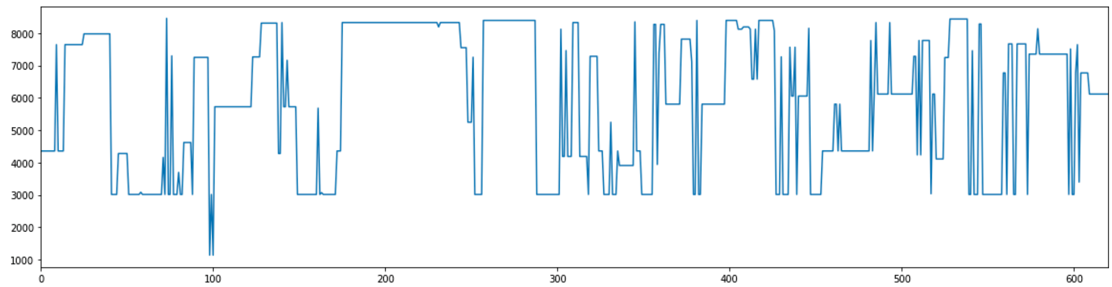


Figure 4.4: Example of 1 IMSI (0) from EBM2 data-set

In Figure 4.4, the plotted graph an example of user mobility. The X-axis is the total no of time slots and Y-axis is eNB id. The example IMSIs are taken for the EBM2 data set. The data with selected features is again processed and converted to the data with 15 min timestamp (96-time slots for each day for each IMSI). This data contains a lot of missing data points as the IMSI may not be connected to the network always. The data interpolation is done considering that the IMSI is staying at the same eNB as it is in the previous time slot till the next filled slot. The maximum number of time slots for the IMSI is 755 in ebm1 and 621 in ebm2.

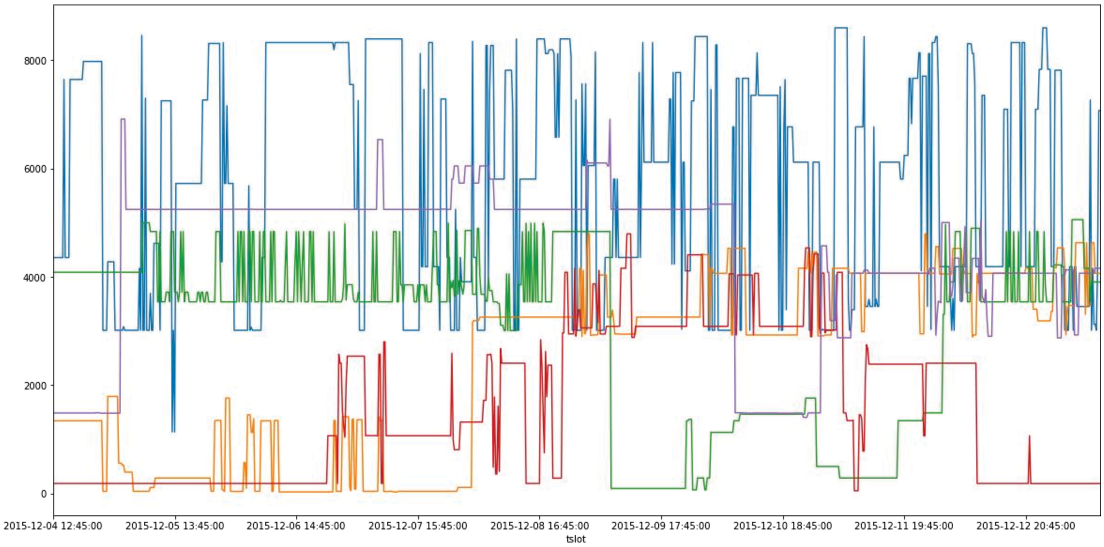


Figure 4.5: Visual representation of 5 IMSIs

In Figure 4.5, The mobility of 5 random users from data set EBM1 is plotted. The X-axis represents the time and Y-axis represents the eNB id of the IMSIs.

4.3 Models Implemented:

4.3.1 RNN:

Recurrent Neural Network (RNN) can be considered as an extension to the conventional feed-forward neural network, that can handle a variable-length sequence input. The RNN can handle variable-length sequences, by having a recurrent hidden state whose activation at each time is dependent on that of the previous time [26]. The output of a generative RNN is the probability distribution of the next element over a sequence, given its current state. RNN is a natural generalization of feed-forward neural networks.

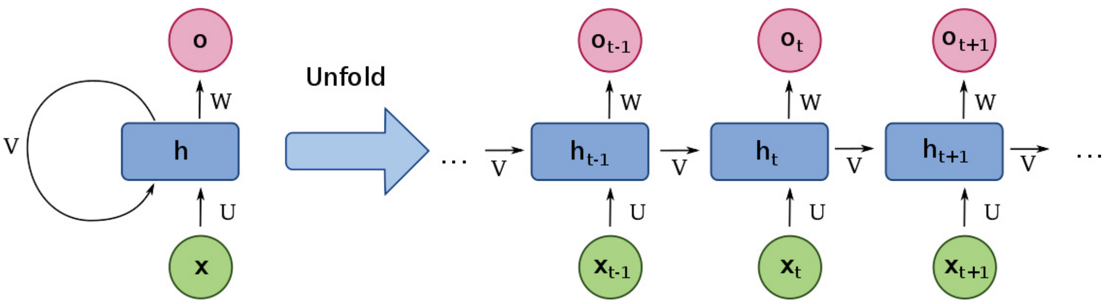


Figure 4.6: RNN [27]

Generally, the recurrent hidden state is implemented as

$$\mathbf{h}_\tau = \sigma_\mathbf{h}\left(\mathbf{W_h x_\tau} + \mathbf{U_h h_{t-1}} + \mathbf{b_h}\right) \tag{4.1}$$

$$\mathbf{y_t} = \sigma_\mathbf{y}\left(\mathbf{W_y h_t} + \mathbf{b_y}\right) \tag{4.2}$$

x_t : input vector
h_t : hidden layer vector
y_t : Output vector
W, U, b : Parameter matrics and vector
σ_x, σ_y : Activation Functions [28]

They are widely used in the areas of speech recognition and time-series data [29, 30].

4.3.2 LSTM:

In the mid-'90s as a solution to vanishing gradient problem, a new approach was proposed by German researchers (Sepp Hochreiter and Juergen Schmidhuber) called LSTM [31]. LSTM stands for Long Short-Term Memory which is an advanced Recurrent Neural Network that can learn long-term dependencies [32]. It has been proven stable and powerful for modelling long-range dependencies in many previous studies [33, 34, 35, 36]. The major innovation of LSTM is its memory cell(neuron) which essentially acts as an accumulator of the state information [37].

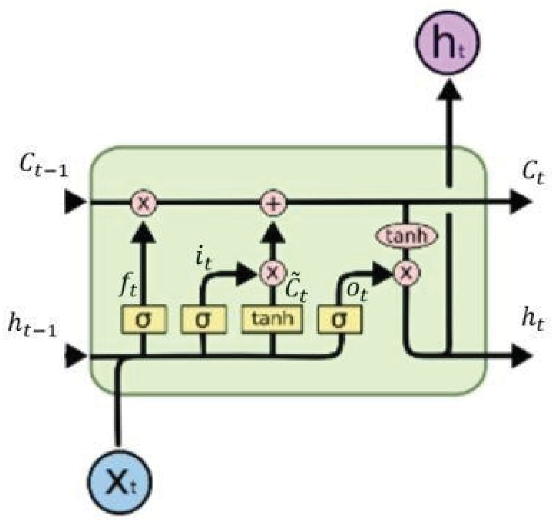


Figure 4.7: Standard LSTM

$$\begin{aligned} \text{C} &= \text{Cell state} \\ \text{f} &= \text{Forget gate} \\ i &= \text{Input gate} \\ \text{o} &= \text{Output gate} \\ \text{h} &= \text{Hidden state} \\ \sigma &= \text{Sigmoid function} \\ \tanh &= \text{Hyperbolic Tangent function} \end{aligned}$$

The LSTM contains a cell state(C), a Forget gate, an Input gate, and an output gate.

Cell state [38]: Cell state handles the long-term memory which allows storing the data related to previous cells in the LSTM cell. Forget gate which is below the cell state modifies it whereas the input modulation gate adjusts the cell state. From the equation, forget gate multiplies with the previous cell state and adds with the input gate multiplied with the current cell state.

$$\mathbf{C_t} = \mathbf{f_t} * \mathbf{C_{t-1}} + \mathbf{i_t} * \widetilde{\mathbf{C}}_\mathbf{t} \tag{4.3}$$

Forget gate [38]: Forget gate forgets the output when multiplied by 0 and stores the information in the cell state is multiplied by 1 to a position in the matrix. From the equation, the sigmoid function has pertained to the weighted function of previous hidden state and input.

$$\mathbf{f_t} = \sigma\left(\mathbf{w_f} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_f}\right) \tag{4.4}$$

Input gate [38]: Activation functions are important parts for every gate. This gate handles the type of information to enter the cell state. Summation of the previous cell state is the equation of the cell state, where the sigmoid function cannot forget the memory and only can add memory within the range of [0,1]. If a float number is added with the range [0,1] it can never be zero/forget, so the tanh activation function is applied to the weighted input with the range [-1,1] which permits the cell state to forget the memory.

$$\mathbf{i_t} = \sigma\left(\mathbf{W_i} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_i}\right) \tag{4.5}$$

$$\tilde{\mathbf{C}}_\mathbf{t} = \tanh\left(\mathbf{W_c} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_c}\right) \tag{4.6}$$

Output gate [38]: This gate handles the information about which value is to be moved from the matrix to the next hidden state from all the possible values.

$$\mathbf{o_t} = \left(\sigma\left(\mathbf{w_o} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_o}\right)\right) \tag{4.7}$$

$$\mathbf{h_t} = \mathbf{o_t} * \tanh(\mathbf{C_t}) \tag{4.8}$$

LSTM has been used in advanced machine learning practices such as time-series analysis, speech recognition, handwriting recognition, etc.

4.3.3 Stacked LSTM:

As LSTMs deal with sequence data addition of layers will increase the abstraction level for the input observations. This results in chunking of input observations and portraying problems at different time scales.

Stacked LSTMs acts as a reliable technique in solving sequence prediction problems. A Stacked LSTM architecture is an LSTM model that consists of multiple LSTM layers where the layers produce sequence output instead of single value output to the next LSTM layer [39].

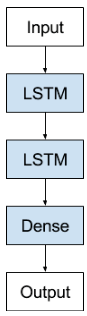


Figure 4.8: Stacked LSTM

4.3.4 Bi-directional LSTM:

Bi-directional LSTM is a combination of Bi-Directional RNN with LSTM [40]. Bi-directional LSTM can access long-range context with input in both directions. If the Bi-directional LSTM is used for hidden layers, we get the Deep Bi-directional LSTM [41]. The structure of Bi-directional RNN and Bi-directional LSTM are shown below.

Bi-directional LSTM duplicates the first recurrent layer in the network making two layers next to each other. The input to the first layer is regular while a reversed copy of the input is sent to the second layer.

The use of bi-directional LSTMs was justified in the domain of speech recognition due to the interpretation of the whole context utterance said rather than a linear interpretation [40].

Bidirectional LSTMs are not appropriate for all sequence prediction problems, but they offer better results in some of the domains.

In bidirectional LSTM, the time-steps of the input sequence are processed one at a time, but the network steps in the input sequence are run in both directions at the same time.

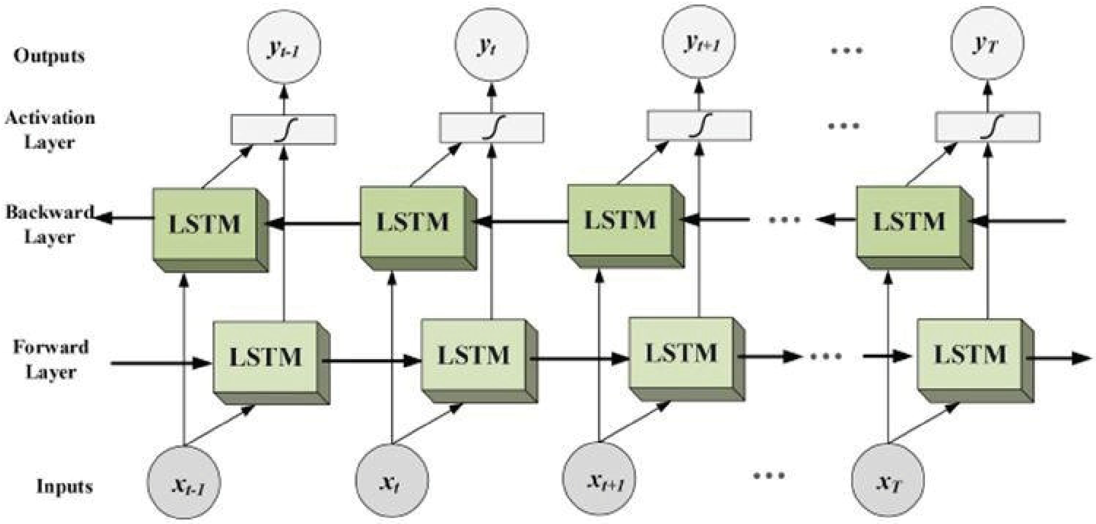


Figure 4.9: Bi-directional LSTM

4.3.5 CNN-LSTM:

CNN-LSTM is a combination of convolutional neural network which is good at one-dimensional data and Long Short-Term Memory which is good at extracting and learning the long-term dependencies. CNN-LSTM consists of Convolutional Layer, Max pooling layer, Sequential layer, and Linear Decoder [42]. The mentioned layers are shown in the architecture diagram below:

The CNN-LSTM model outperforms regression and convolutional neural network models [42]. As it is a hybrid approach, it is used in predictions that are interrelated and have longer dependencies.

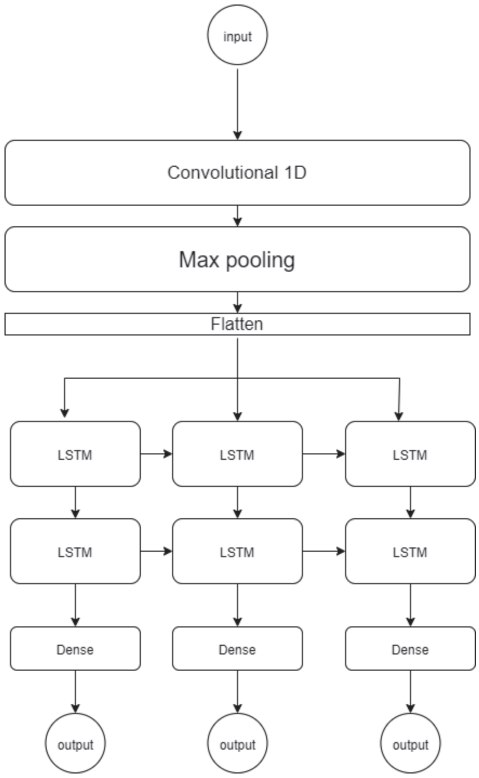


Figure 4.10: CNN-LSTM

4.3.6 GRU:

Gated Recurrent Unit GRU) was proposed by Chung, Junyoung in [34, 43]. It is the most popular variant of LSTM. In this variant the forget gate and the input gates combined into a single gate called "Update gate," and also made some other changes like merging the cell state and hidden state. This model is simpler than the standard LSTM [38]. It was developed to make recurrent units adaptively capture the dependencies of different time series. The GRU has gating units that modulate the flow of information inside the unit, but they do not have separate memory cells [26].

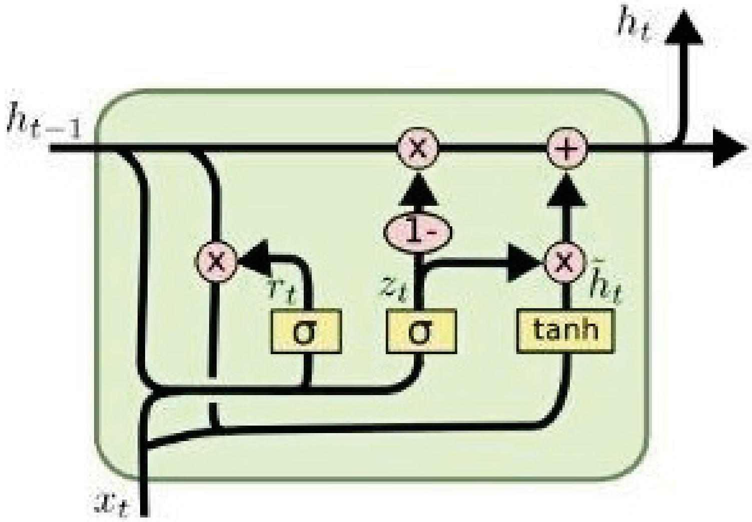


Figure 4.11: GRU

$$\begin{gathered}
x_t = \text{input vector} \\
z_t = \text{update gate vector} \\
r_t = \text{reset gate vector} \\
h_t = \text{output vector} \\
\sigma = \text{sigmoid function} \\
\tanh = \text{Hyperbolic Tangent function}
\end{gathered}$$

$$\mathbf{z_t} = \sigma\left(\mathbf{W_z} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}]\right) \tag{4.9}$$

$$\mathbf{r_t} = \sigma\left(\mathbf{W_r} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}]\right) \tag{4.10}$$

$$\tilde{\mathbf{h}}_\mathbf{t} = \tanh\left(\mathbf{W} \cdot [\mathbf{r_t} \cdot \mathbf{h_{t-1}}, \mathbf{x_t}]\right) \tag{4.11}$$

$$\mathbf{h_t} = (\mathbf{1} - \mathbf{z_t}) * \mathbf{h_{t-1}} + \mathbf{z_t} * \tilde{\mathbf{h}}_\mathbf{t} \tag{4.12}$$

The information from the previous steps that are required to be passed on to the future is determined by the model with the help of update gate $\mathbf{z_t}$. Similarly, the model decides the amount of past information to forget with the help of the reset gate $\mathbf{r_t}$.

4.3.7 Metrics:

Accuracy:
The metrics accuracy tells whether the model is being trained well and calculates the performance of the model. In this thesis accuracy is the metrics that evaluate

the performance of the model/technique in predicting the next 96 eNB's based on the test data. The formula to calculate the accuracy of the model is:

$$\text{Accuracy} = \frac{\text{no of } (\mathbf{T} = \mathbf{P})}{\mathbf{t}} * \mathbf{100} \tag{4.13}$$

$$\begin{aligned} \text{T} =& \ \text{True eNB's} \\ \text{P} =& \ \text{Predicted eNB's} \\ \text{t} =& \ \text{length of True eNB's} \end{aligned}$$

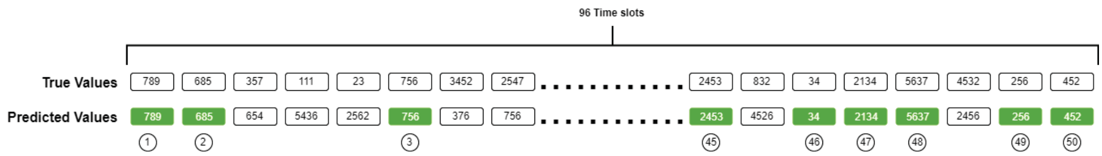


Figure 4.12: Metrics Accuracy

From the above figure, the model predicts 50 time-slots among 96 time-slots. The accuracy will be $\frac{50}{96} * 100 = 52.08$. The accuracy of the selected IMSI is 52.08%

4.4 Model Implementation

Global Model: In this technique, the whole data set is split to training and testing data-set and all the IMSIs are on a single model.

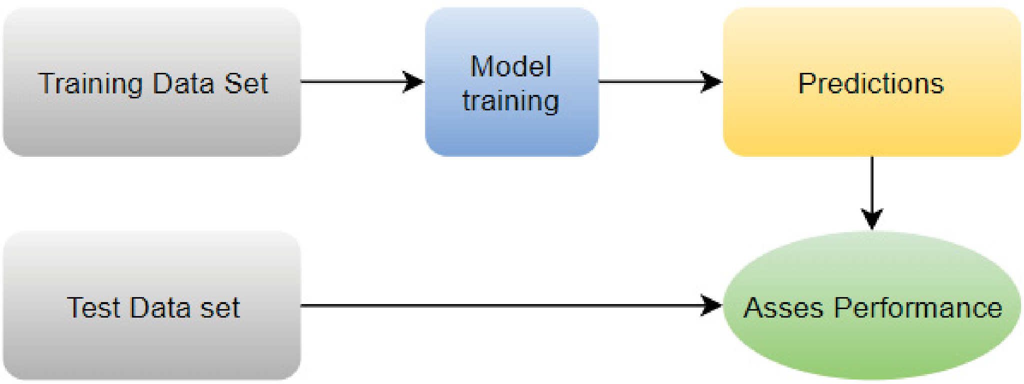


Figure 4.13: Global Model

Individual Model: In this technique, each IMSI is assumed as an individual data set and is split to train and test data and trained on its own model. In simple words each IMSI has its own model.

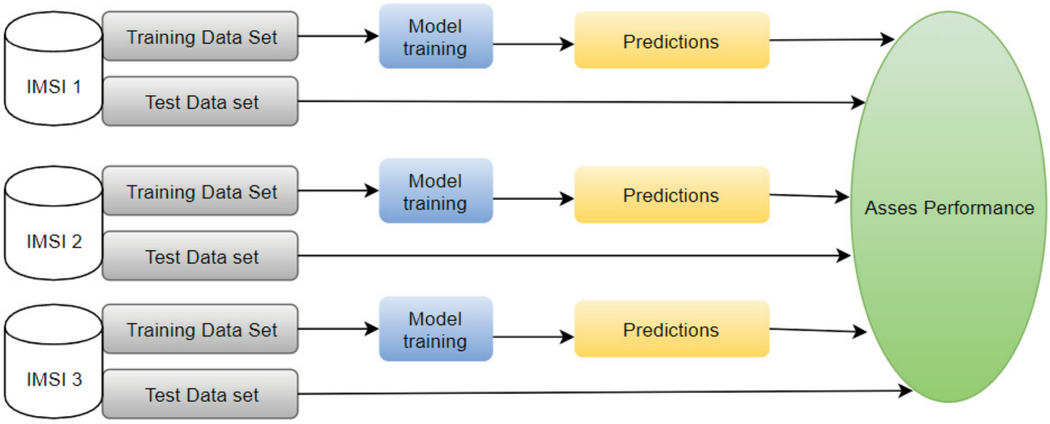


Figure 4.14: Individual Model

Chapter 5

Results

5.1 Literature review results:

The finding of the literature reviews is tabulated below:

Comment	**Answer**
LSTM fully convolutional networks for time series classification [44].	From the experimental results, the author concluded that the LSTM based deep neural network shows improvement over the current state of the art time series classification.
Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model [45].	LSTM models were used to detect faulty problems in industries that showed positive results.
Deep learning for time-series analysis [46].	Combining multiple deep learning models (Stacked LSTM and CNN-LSTM) help us to get better results and be more effective for time series analysis.
A novel bidirectional mechanism based on a time series model for wind power forecasting [47]	The bidirectional mechanism was used in improving the performance of the deep learning models for forecasting.
Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition [48].	LSTM was combined with CNN it helped in better video action recognition which explains to us how well the deep neural networks work when they are combined network.
Recurrent neural networks for multivariate time series with missing values [49]	RNN and GRU models are used to predict the multi variate-time series data with missing labels.
Time series prediction using RNN in multi-dimension embedding phase space [50].	The prediction power and value of RNN in chaotic time series was proved to be efficient and showed some impact

Conditional time series forecasting with convolutional neural networks [51]	CNN was found to be the simple, efficient and easily interpretable network that can act as a strong baseline for forecasting conditional time series
Time-series extreme event forecasting with neural networks at uber [52]	Neural networks were used in forecasting time-series events for uber which tells us how effective, efficient and accurate the deep neural networks are.
Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction [53]	A deep-stacked bidirectional and unidirectional LSTM neural network is proposed in this paper for network-wide traffic speed prediction.
Lstm Network: a deep learning approach for short-term traffic forecast [54]	After comparing the LSTM network with other representative forecast models, the author confirms that the LSTM network achieves better performance.
Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks [55]	In this paper, the author proposed a combination of deep convolutional neural network and long short-term memory which outperforms DCNN, LSTM, and SVM

Table 5.1: Literature review results

5.2 Experimental Results:

This chapter presents the results of the experiment.

An experiment is conducted on 2 datasets (EBM 1 and EBM 2) using deep learning models to assess the performance of the global model and individual model.

EBM 1 results:

Model	Global Model	Individual Model
RNN	31.6278	51.2647
LSTM	34.6254	56.6371
Stacked LSTM	39.5318	58.2491
CNN-LSTM	32.5148	54.1143
GRU	39.5361	56.6381
Bi-directional LSTM	39.7384	56.6276

Table 5.2: EBM 1 data-set results

The prediction accuracy of the deep learning models (RNN, LSTM, Stacked LSTM, CNN-LSTM, GRU, Bi-directional LSTM) for the global model was $31 < X < 40$ and the accuracy for an individual model using the same models were 52 < X < 59. From the above results, the individual model for each IMSI performs better than the global model.

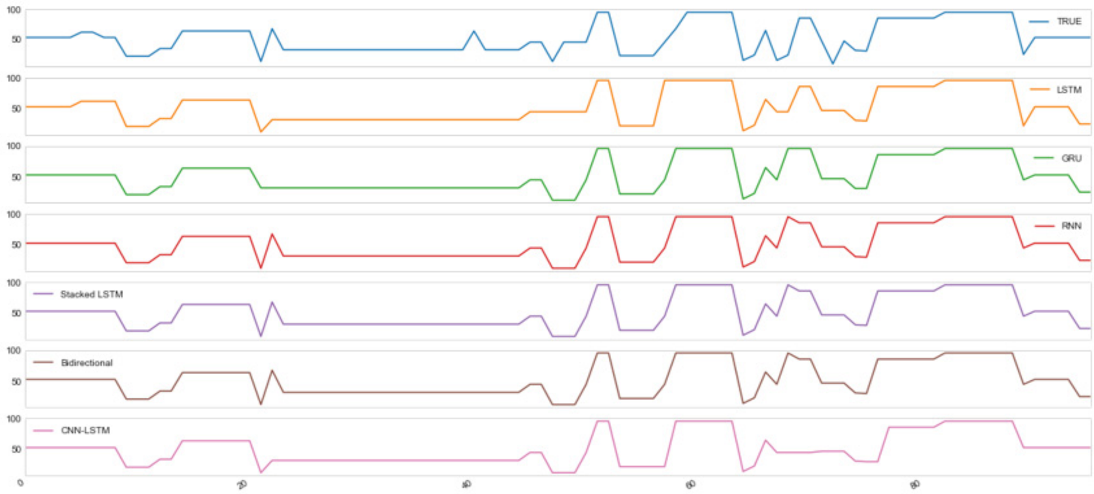


Figure 5.1: Example of forecasting result of 1 IMSI from EBM 1 data-set

The above figure represents the true values and predicted values of 1 IMSI using all deep learning models from data set EBM1. The X-axis represents 96 time-slots and Y-axis represents eNB id.

EBM 2 results:

Model	Global Model	Individual Model
RNN	39.1573	50.6351
LSTM	41.8680	55.3271
Stacked LSTM	34.4791	53.8531
CNN-LSTM	36.5712	52.3218
GRU	40.1743	55.1771
Bi-directional	38.8333	55.7927

Table 5.3: EBM 2 data-set results

The prediction accuracy of the deep learning models (RNN, LSTM, Stacked LSTM, CNN-LSTM, GRU, Bi-directional LSTM) for the global model was 34 < X < 42 and the accuracy for the individual model using the same models were 50 < X < 56. From the above results, the individual model for each IMSI performs better than the global model.

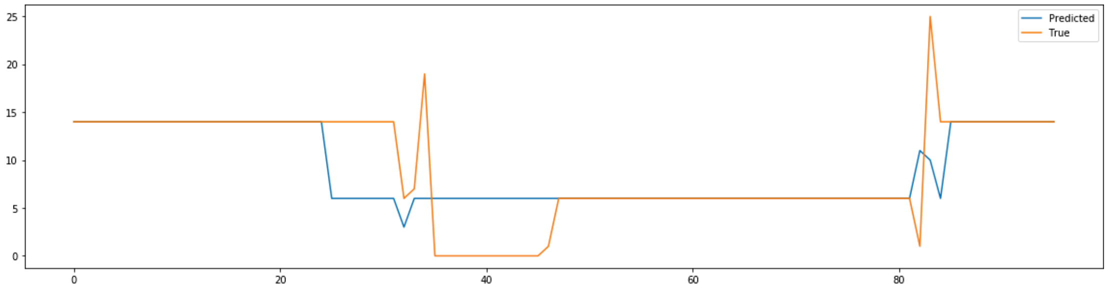


Figure 5.2: Example of forecasting result of 1 IMSI from EBM 2 data-set

The above figure represents the true values and predicted values of 1 IMSI using LSTM from data set EBM2. The X-axis represents the time-slots and Y-axis represents eNB id.

Chapter 6

Analysis and Discussion

The main aim of this thesis was accomplished by evaluating the performance of the deep learning-Global models and deep learning-individual models with the accuracy of the metrics. The performance of the models is tabulated in section 4. From the results obtained by conducting the experiment are compared and the best technique is identified. The objectives of this thesis are completed by answering the research questions.
After carefully analyzing the results obtained from the literature review, it is identified that LSTM is the best performing deep learning algorithm for time series forecasting. Additionally, it has also been found that LSTM has several variants such as Bidirectional LSTM, Stacked LSTM, and GRU. Papers' results of the tests performed by other researches showed that LSTM and its variants have outperformed other popular machine learning techniques to forecast time series data. Therefore, using algorithms LSTM, Bidirectional LSTM, Stacked LSTM and GRU an experiment is conducted.

6.1 Answering Research Questions:

RQ1) What are the deep learning algorithms that are suitable to forecast the time series data?

Suitable deep learning models are identified using the literature review method. From this research, the deep learning algorithms that are identified for time series forecasting are RNN, LSTM, Bidirectional LSTM. Stacked LSTM, GRU. These algorithms are variants of the recurrent neural networks. The detail description of these models is explained in Chapter 4. Using these algorithms mentioned above the experiment is conducted to answer the second research question.

RQ2) How is the performance of the deep learning models vary for the global model and individual model in predicting the next n eNB's?

To answer this research question experiment is conducted using the algorithms obtained from RQ1. The experiment is conducted using 2 data sets (Details of the data set are mentioned above). RNN, LSTM, Bi-directional LSTM, Stacked LSTM, CNN-LSTM, GRU are the deep learning models that are trained to obtain the accuracy metrics. Initially, the whole data-set was split into training and testing data set and each of the selected models is trained. The metrics accuracy is tabulated for the global model. Now each IMSI in the data-set is divided into training and test data-set and is trained on its own model. In this case, the number of unique IMSIs in the data-set is equal to the number of models. The metrics accuracy is tabulated as an individual model. The same process is repeated for the second data-set. The results that are obtained after conducting the experiment are tabulated in chapter 5. Comparing these results, we can conclude that the individual model gives better accuracy than the global model.

6.2 Limitation of this work:

The following limitations are identified for this work:

1. We can only do experiments on datasets given by Ericsson (EBM1, EBM2). The data contains a lot of empty data points, which are should be cleaned. The data for the user who is staying in the same eNB's for the whole period is not considered. The users with less than 7 days of data are not considered.

2. The performance of the neural networks can be influenced by the parameters taken and computational resources.

3. Holiday effects, which are a critical factor in forecasting user mobility will be not in the cope of this thesis.

Chapter 7

Conclusion and Future Work

From the above research, selected deep learning algorithms (RNN, LSTM, Stacked-LSTM, CNN-LSTM, Bi-directional LSTM, GRU) are identified as suitable algorithms. The selected models are trained with the data sets (EBM1 and EBM2) to predict the next 96 eNB's (eNB's of the next day/24 hours) based on the historical data. The model is evaluated using the test data. After evaluating the results, the individual model gives better prediction accuracy than the global model. From the results, there is not much difference between the deep learning models. By improving the prediction accuracy of the users, the telecom carriers can improve their quality of service and can reduce the cost of paging.

The future work for this thesis can be predicting multiple eNB's at a single time-slot. Taking holiday effects into consideration in predicting user mobility can improve the prediction accuracy. The time-series data reflects the event or observation with respect to time. If any unusual event (Festival, Sports events, etc.) it affects the results in time series forecasting.

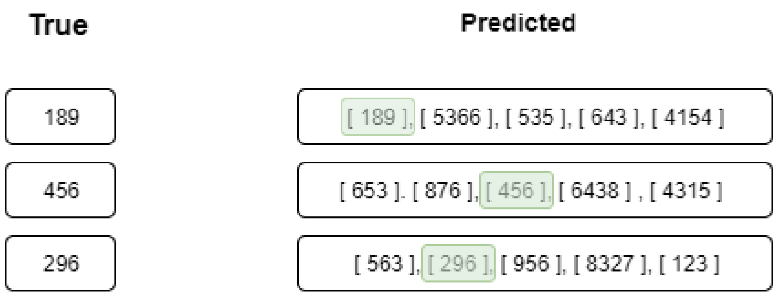


Figure 7.1: Predicting multiple eNB's at a single time-slot

Bibliography

[1] *Telcos: The untapped promise of big data.* (Visited on).

[2] H. Si et al. "Mobility Prediction in Cellular Network Using Hidden Markov Model". In: *2010 7th IEEE Consumer Communications and Networking Conference*. Jan. 2010, pp. 1–5. DOI: 10.1109/CCNC.2010.5421684.

[3] Yuanyuan Qiao et al. "A Hybrid Markov-based Model for Human Mobility Prediction". In: *Neurocomputing* 278 (Sept. 2017). DOI: 10.1016/j.neucom.2017.05.101.

[4] U. Palani, K. C. Suresh, and Alamelu Nachiappan. "Mobility prediction in mobile ad hoc networks using eye of coverage approach". In: *Cluster Computing* 22 (2018), pp. 14991–14998.

[5] David Stynes, Kenneth Brown, and Cormac Sreenan. "A probabilistic approach to user mobility prediction for wireless services". In: Sept. 2016, pp. 120–125. DOI: 10.1109/IWCMC.2016.7577044.

[6] *Enterprise Networks - 5G technology needs edge computing architecture.* URL: https://www.cisco.com/c/en/us/solutions/enterprise-networks/edge-computing-architecture-5g.html.

[7] *Internet of Things forecast – Ericsson Mobility Report.* URL: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast.

[8] INTELLIGENCE BY AM TURING. "Computing machinery and intelligence-AM Turing". In: *Mind* 59.236 (1950), p. 433.

[9] *What is Artificial Intelligence Exactly?* URL: https://www.badfive.com/what-is-artificial-intelligence-exactly/.

[10] *What is AI (artificial intelligence)? - Definition from WhatIs.com.".* URL: https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence..

[11] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (July 1959), pp. 210–229. ISSN: 0018-8646. DOI: 10.1147/rd.33.0210.

[12] *What is Machine Learning? A definition - Expert System.* URL: https://www.expertsystem.com/machine-learning-definition/.

[13] *Supervised and Unsupervised Machine Learning Algorithms.* URL: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/.

[14] *Reinforcement learning - GeeksforGeeks.* URL: https://www.geeksforgeeks.org/what-is-reinforcement-learning/.

[15] *Deep Learning Definition.* URL: https://www.investopedia.com/terms/d/deep-learning.asp.

[16] *Deep Learning Artificial Neural Networks.* URL: https://machinelearningmastery.com/what-is-deep-learning/.

[17] Maureen Caudill. "Neural Networks Primer, Part I". In: *AI Expert* 2.12 (Dec. 1987), pp. 46–52. ISSN: 0888-3785.

[18] David Fumo. *A Gentle Introduction To Neural Networks Series — Part 1.* URL: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc.

[19] Chenyang Xu and Changqing Xu. "Predicting Personal Transitional Location Based on Modified-SVM". In: Dec. 2017, pp. 340–344. DOI: 10.1109/CSCI.2017.57.

[20] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. "Next Place Prediction Using Mobility Markov Chains". In: *Proceedings of the First Workshop on Measurement, Privacy, and Mobility.* MPM '12. Bern, Switzerland: Association for Computing Machinery, 2012. ISBN: 9781450311632. DOI: 10.1145/2181196.2181199. URL: https://doi-org.miman.bib.bth.se/10.1145/2181196.2181199.

[21] Akinori Asahara et al. "Pedestrian-Movement Prediction Based on Mixed Markov-Chain Model". In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* GIS '11. Chicago, Illinois: Association for Computing Machinery, 2011, pp. 25–33. ISBN: 9781450310314. DOI: 10.1145/2093973.2093979. URL: https://doi.org/10.1145/2093973.2093979.

[22] X. Fan et al. "A Deep Learning Approach for Next Location Prediction". In: *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)).* May 2018, pp. 69–74. DOI: 10.1109/CSCWD.2018.8465289.

[23] Abdulrahman Al-Molegi, Mohammed Jabreel, and Baraq Ghaleb. "STF-RNN: Space-Time Features-based Recurrent Neural Network for Predicting People's Next Location". In: Dec. 2016. DOI: 10.1109/SSCI.2016.7849919.

[24] *Python (programming language).* URL: https://en.wikipedia.org/wiki/Python_(programming_language).

[25] *Top 5 best Programming Languages for Artificial Intelligence field*. URL: https://www.geeksforgeeks.org/top-5-best-programming-languages-for-artificial-intelligence-field/.

[26] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[27] *RECURRENT NEURAL NETWORKS 101: UNDERSTANDING THE BASICS OF RNNS AND LSTM*. URL: https://builtin.com/data-science/recurrent-neural-networks-and-lstm.

[28] *Recurrent neural network*. URL: https://en.wikipedia.org/wiki/Recurrent_neural_network.

[29] David Salinas, Valentin Flunkert, and Jan Gasthaus. *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*. 2017. arXiv: 1704.04110 [cs.AI].

[30] Réal Carbonneau, Kevin Laframboise, and Rustam Vahidov. "Application of machine learning techniques for supply chain demand forecasting". In: *European Journal of Operational Research* 184 (Feb. 2008), pp. 1140–1154. DOI: 10.1016/j.ejor.2006.12.004.

[31] Caryn Elisabeth. *A Beginner's Guide to Recurrent Neural Networks*. URL: https://medium.com/@carynmccarthy15/a-beginners-guide-to-recurrent-neural-networks-bfacb27bddb6.

[32] Wentao Zhu et al. *Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks*. 2016. arXiv: 1603.07772 [cs.CV].

[33] Alex Graves. *Generating Sequences With Recurrent Neural Networks*. 2013. arXiv: 1308.0850 [cs.NE].

[34] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[35] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. *On the difficulty of training Recurrent Neural Networks*. 2012. arXiv: 1211.5063 [cs.LG].

[36] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[37] Xingjian Shi et al. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. 2015. arXiv: 1506.04214 [cs.CV].

[38] Eugine Kang. *Long Short-Term Memory (LSTM): Concept*. URL: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359.

[39] Jason Brownlee. *Stacked Long Short-Term Memory Networks*. URL: https://machinelearningmastery.com/stacked-long-short-term-memory-networks/.

[40] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM networks". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 4. July 2005, 2047–2052 vol. 4. DOI: 10.1109/IJCNN.2005.1556215.

[41] A. Graves, N. Jaitly, and A. Mohamed. "Hybrid speech recognition with Deep Bidirectional LSTM". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. Dec. 2013, pp. 273–278. DOI: 10.1109/ASRU.2013.6707742.

[42] Jin Wang et al. "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model". In: Jan. 2016, pp. 225–230. DOI: 10.18653/v1/P16-2037.

[43] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: 1409.1259 [cs.CL].

[44] F. Karim et al. "LSTM Fully Convolutional Networks for Time Series Classification". In: *IEEE Access* 6 (2018), pp. 1662–1669. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2779939.

[45] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. *Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model*. 2016. arXiv: 1612.06676 [cs.LG].

[46] John Cristian Borges Gamboa. *Deep Learning for Time-Series Analysis*. 2017. arXiv: 1701.01887 [cs.LG].

[47] Yongning Zhao et al. "A novel bidirectional mechanism based on time series model for wind power forecasting". In: *Applied Energy* 177 (Sept. 2016), pp. 793–803. DOI: 10.1016/j.apenergy.2016.03.096.

[48] Xuanhan Wang et al. "Beyond Frame-level CNN: Saliency-aware 3D CNN with LSTM for Video Action Recognition". In: *IEEE Signal Processing Letters* PP (Sept. 2016), pp. 1–1. DOI: 10.1109/LSP.2016.2611485.

[49] Zhengping Che et al. "Recurrent Neural Networks for Multivariate Time Series with Missing Values". In: *Scientific Reports* 8 (June 2016). DOI: 10.1038/s41598-018-24271-9.

[50] Jun Zhang and K. F. Man. "Time series prediction using RNN in multi-dimension embedding phase space". In: *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*. Vol. 2. Oct. 1998, 1868–1873 vol.2. DOI: 10.1109/ICSMC.1998.728168.

[51] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. *Conditional Time Series Forecasting with Convolutional Neural Networks*. 2017. arXiv: 1703.04691 [stat.ML].

[52] Nikolay Laptev et al. "Time-series extreme event forecasting with neural networks at uber". In: *International Conference on Machine Learning*. Vol. 34. 2017, pp. 1–5.

[53] Zhiyong Cui et al. *Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction*. 2018. arXiv: 1801.02143 [cs.LG].

[54] Z. Zhao et al. "LSTM network: a deep learning approach for short-term traffic forecast". In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75. ISSN: 1751-9578. DOI: 10.1049/iet-its.2016.0208.

[55] Haiyang Yu et al. *Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks*. 2017. arXiv: 1705.02699 [cs.LG].

Faculty of , Blekinge Institute of Technology, 371 79 Karlskrona, Sweden