FEASIBILITY ANALYSIS OF LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK IN TIME SERIES CRIME PREDICTION:
A CASE OF BOLE SUB CITY POLICE DEPARTMENT

TSION ESHETU

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters in Information System Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

Oct,2019

FEASIBILITY ANALYSIS OF LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK IN TIME SERIES CRIME PREDICTION:
A CASE OF BOLE SUB CITY POLICE DEPARTMENT

TSION ESHETU

Advisor: DR. BAHIRU SHIFAW

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters in Information System Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

Oct,2019

DECLARATION

I hereby declare that this MSc thesis is my original work, has not been presented for a degree in any other university, and all source of materials used for this thesis have been properly acknowledged.

Name: Tsion Eshetu

Signature _____

This MSc thesis has been submitted for examination with my approval as thesis advisor.

Name: Dr. Bahiru Shifaw

Signature_____

Date of Submission: Sep, 2019

ACKNOWLEDGEMENTS

I am very grateful to GOD. Without His grace and blessings, this study would not have been possible.

I would like to thank my university advisor Dr. Bahiru Shifaw for his guidance, insights and willingness to talk in depth about the subject field.

Of course, I would like to thank Bole sub city police commission for their sincere help in gathering crime data I need for my research.

I am also thankful to my family and friends for always being there in case of any difficulty in my research.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Crime influences people in many ways. Prior studies have shown the relationship between time and crime incidence behavior. This thesis attempts to determine and examine the relationship between time, crime incidences types and locations by using one of the neural network models for time series data that is, long short-term memory network. Six thousand thirty-three (6033) records of five years (2014-2018) data were collected from bole sub city police department in a paper form which has been converted to electronic format. After pre-processing of raw data, it has been analyzed and tested using long short-term memory recurrent neural network model.

R-square score is also used to test the accuracy. The result for r-square score on crime type prediction on monthly, daily and hourly basis is 0.879, 0.925 and 0.959 respectively whereas the results on location prediction in monthly, daily and hourly manner are 0.930, 0.989 and 0.968. The study results show that applying long short-term memory recurrent neural network (LSTM RNN) enables to come up with more accurate prediction about crime incidence occurrence with respect to time. Predicting crimes accurately helps to improve crime prevention and decision and advance the justice system.

Key words: Crime prediction, time-series prediction, predictive policing, LSTM, RNN

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

OSAC	State Overseas Security Advisory Council
UNODC	United nations office on drugs and crime
RNN	Recurrent neural networks
LSTM	Long Short-Term Memory
ANN	Artificial Neural Networks
AI	Artificial Intelligence
ML	Machine Learning
ARMA	Autoregressive moving Average
AR	Auto-regressive
MA	moving average
VARMA	vector auto-regressive moving average
ARIMA	Autoregressive integrated moving Average
SARIMA	Seasonal Autoregressive integrated moving Average
ESN	Echo state networks
CNN	convolutional neural network
MLP	multilayer perception
NLP	natural language processing
SVM	support vector machines
FIS	fuzzy inference system

FL	fuzzy logic
ANFIS	neuro-fuzzy inference system
URL	Uniform resource locator
WEKA	Waikato Environment for knowledge Analysis
ID3	Iterative Dichotomiser 3
KNN	K-Nearest Neighbor
PacRNN	pairwise-ranking based collaborative recurrent neural networks
BPR	Bayesian Personalized Ranking
EHR	Electronic Health Record
GUI	graphical user interface
HTML	Hypertext mark-up language
MIME	Multipurpose Internet Mail
CSV	Comma Separated Values
SQL	Structured query language
HDF5	Hierarchical data format 5
API	Application Program Interface
GPU	Graphics Processing Unit
CPU	Central Processing Unit
MATLAB	Matrix Laboratory
TPU	Tensor Processing Unit
GO	Golang
CNTK	Cognitive toolkit
iOS	Internetwork operating system

ADAgrad	Adaptive gradient algorithm
RMSprop	Root Mean Squared Propagation
MAE	Mean Absolute Error
DDGF	Data-Driven Green's Function
EM	Expectation Maximization
PHM	Prospective Hotspot Maps
HADOOP	High Availability Distributed Object-Oriented Platform
HDFS	Hadoop Distributed File System
YARN	Yet Another Resource Negotiator

Chapter 1

INTRODUCTION

1.1 Background of the study

Crime is a major issue within society; so too is fear of crime [1]. Human beings need to live and work in a place where they are safe. They want to ensure that there is a concerned body that protects their lives as well as their properties from potential hazards. In fact, one of the main functions of any government is to ensure that law and order for the security of its citizens are put in place. Crime prevention polices have been shown to be beneficial in reducing not only crime, but also fear of crime. In the current era when a large volume of crimes occur in society, prevention of crime has become one of the most imperative global issues, along with the great concern of strengthening public security [2]. Areas in which crime prevention strategies have not been employed have been seen to have higher fear of crime, decreasing property values and in turn, lower quality of life.

The scale of crime represents a considerable challenge to law enforcement agencies in Africa [3]. Ethiopia is one of the developing countries that have been influenced by crime through threatening the quality of life, intimidating human rights and fundamental freedom, and causing a severe challenge to the society and so on.

In the annual United States Department of State Overseas Security Advisory Council (OSAC) 2016 Crime and Safety Report of Ethiopia rated Ethiopia's crime rate as high. The same holds true in the 2015 report, the crime was high, while in 2014 it was considered as mere opportunistic and thus not rated at all [4].

United nations office on drugs and crime (UNODC)and the Federal Government of Ethiopia started the development of a National Crime Prevention Strategy on 1 July 2016 with the goal of tackling the conditions allowing crime to flourish, supporting groups that are at-risk of committing crimes and victims of crime alike, as well as promoting the reintegration of offenders so as to keep them from reoffending.

In the case of Ethiopia, including Addis Ababa police commission, until recent years there were no modern tools and techniques employed to facilitate the handling and processing

of records [5]. Some of the generalized statistical records that associate with crime are kept in a computerized way but the detailed data we need for our research is stored manually.

The criminal records, which is used for this study, contains information about the type of crime committed by each criminal, specific date and time and also specific place. It consists of about 10,000 records of five years of data. Criminal records from six sub-stations are sent manually to the commission by filling the criminals profile form. Only the crimes that has given heavy weight will be recorded in the commission's criminal record list.

The crime prevention strategy are being taken through community policing which focuses on building ties and working closely with the community. These methods include encouraging the community to help prevent crime by providing advice, creating teams of officers to carry out community policing in designated neighbour, clear communication between the communities and the police about their objectives and strategies and the likes.

Community policing is related to problem-oriented policing.Nevertheless, the crime control effort does not depend on the previous records and can't tell what comes in the future. To make the police work more truthful, on top of community policing, predictive policing is a powerful tool to predict events and also to reveal the hidden causality.

Police districts in most of developed countries "prediction" has become the new watchword for innovative policing. Using predictive analytics, high-powered computers, and good oldfashioned intuition, police are adopting predictive policing strategies that promise the holy grail of to stop crime before it happens.

The fundamental notion underlying the theory and practice of predictive policing is that we can make probabilistic inferences about future criminal activity based on existing data [6]. In short, we can use data from a wide variety of sources to compute estimates about phenomena such as where gun violence is likely to occur [7], where a serial burglar is likely to commit his next violation, and which individuals a suspect is likely to contact for helpPast data from both conventional and unconventional sources combine to yield estimates, with a specified degree of certainty, about what will happen in the future. Police organizations can then make decisions accordingly [8].

Data mining turns out to be a stepping stone for predictive policing. By using data and data mining technologies, predictive policing has the potential to provide the best evaluation of what will happen. Researches have shown that data mining can greatly improve crime analysis and aid in reducing and preventing crime.

1.2 SWOT Analysis

Analysis of strengths, weaknesses, opportunities and threats is recommended to identify the issues that would provide critical information. Prior identification of weaknesses and threats helps to identify appropriate approaches for internal improvement and justification of factors that may result in adverse impacts beyond the control of the crime prevention. Recognition of strengths and opportunities enables gaining the maximum benefits from internal and external environments toward achieving the goals and targets set.

key strengths, weaknesses, opportunities and threats of crime prevention in Ethiopia.

Strengths	**Weakness**
➢ Overall increasing level of awareness on understanding the extent and impact of crime ➢ At least on paper, every criminal data is available. ➢ Generalized form of crime reports exists in a computerized form ➢ Enforcement networks with community exist, some well established	➢ The importance of data and information for upcoming crime incidences is not well understood by enforcement authorities ➢ Data on many aspects of impacts are often lacking ➢ Databases for each crime report is not established. ➢ Too few skills of data analysis
Opportunities	**Threats**
➢ Developments crime prediction strategy will improve efficiency and ability to prevent crime	➢ Reductions in public budgets threaten data gathering, investment in information systems, etc. ➢ Lacking sufficient data on enforcement practice

➢ Introduce databases to prosecutors to increase the effectiveness of data supplement. ➢ To promote a wider concept of crime prevention technologies	

Table 1.2-1: SWOT analysis

1.3 Statements of the problem

One of the main social problems in modern society is crime which is basically defined as an offence against public law. Since it is a main problem, crime has major effects on victims, the society and social institutions. Crime is a multi-faceted social problem because it involves personal responsibility as well as social, cultural and political aspects that contribute to it.

Numerous studies have shown that different factors such as time of the day, weather, environmental characteristics, and past incidence of crime will make an influence in the future criminal activity.

According to the current U.S. Department of State Travel Advisory at the date of this report's publication, Ethiopia has been assessed as level two exercise increased caution. Petty crimes occur at random in Addis Ababa. Criminal Violence in Addis Ababa and in southwestern and southeastern Ethiopia has resulted in numerous injuries and deaths. Moreover, location of police resources can be viewed as another problem. It involves the total amount of police time required to perform tasks. Demanding services on irregular time is one of the problem of police manpower assignment. [9].

This research incorporates a set of closely related crime events from both the immediate and longer-term past, with more recent crimes given a heavier weight in order to develop predictive model using LSTM recurrent neural network.

1.4 Research Questions

The research questions this study seek to answer are the following.

RQ1. How can we predict crime types and their time of occurrence in relation to time of incidence?

RQ2. How well will LSTM RNN perform when implemented in crime prediction problems?

1.5 Objectives

1.5.1 General objective

The objective of this research is to apply long short term memory network and verify the feasibility and accuracy of a model in crime prediction

1.5.2 Specific objective

- To identify and examine research and other literature that addresses the future of crime
- To collect data from different sources
- To pre-process the collected data
- To split data into training set and test set
- To develop LSTM model by analyzing training set
- To predict crime type as well as location
- To apply test set and optimize rule's parameters
- To interpret and analyse results of the model
- Finally, to forward recommendations based on findings

1.6 Significance of the study

This research would be helpful for police resources allocation before the crime happenes. It can help in the distribution of police at most likely crime places for any given time, to grant an efficient usage of police resources.

1.7 Research Methodologies

1.7.1 Data Collection

The data we used in our research was obtained from bole sub city police department. They were all in a paper form. We feed the data to computer using excel then convert it to a CSV format. The reason behind using CSV format is that the library we used for analysis provides a module to handle this type of format and can easily read and process.

1.7.2 Data pre-processing

After the data is collected and converted to a soft copy, pre-processing techniques like missing value detection, converting categorical data to numeric, data normalization, transformation and splitting into train and test set are done in order to convert the data to a model understandable form.

1.7.3 Modelling

LSTM RNN is applied to train the dataset and evaluate the performance of the test data to analyse the model. A remarkably effective regularizer called dropout regularizer is used to reduce over fitting and improve errors.

1.7.4 Evaluation

Coefficient of determination also called R-square is used to evaluate the model. It is the most important and popular method to evaluate how well the predicted line approximates the real data points. And MAE is used to know the difference between the predicted and the actual value. It is easy to understand and can handle outliers better than the other metrics.

1.8 Scope and limitations

This research is concerned on crime types that befall in time and place. Furthermore, the research is based on data obtained about crimes occurred in Addis Ababa at bole sub city. While conducting this research, certain problems were facing as constraints. Shortage of time is one of the main constraints as data used for this study was not appropriately maintained using computers by the organizations. Besides, financial problem is another limitation which was challenging for the successful completion of this research on time.

1.9 Operational definitions

Community policing: community-oriented policing that focuses on building ties and working with communities.

Time series prediction: Prediction of a variable value at equal time intervals into the future.

Predictive policing:is the application of quantitative analytical techniques to find expected targets for police involvement and prevent crime or answer for previous crimes by creating statistical predictions [10].

1.10 Thesis organization

The remaining part of the thesis is prepared as follows. Chapter two presents the review of literature on the domain of data mining technology and crime prevention respectively and also related works about it.In chapter three the methodologies that are used to accomplish this thesis are discussed briefly including data collection, preprocessing and prediction methods, materials, and tools which have been used. Chapter four designs and models each phase. It comprises training and building the models. Chapter five reports the results of the designed model. Results of the experiment are also analyzed and interpreted. Conclusion is presented in chapter six. The last chapter presents recommendation and future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

2.2 What is crime

Everyone knows what crime is. There are many definations of crime. But the popular definitions approach the law as a given, sociological definitions approach the issue in a more social way drawing attention not only to the act itself but the law itself and whose interests it seeks to protect. It makes a distinction between private offences (such as arguments or personal disputes) and public offences that offend a broader set of social norms or values. One of the most popular defination is the Oxford Dictionary of Sociology defines crime. It defines crime in a more complex way: 'an offence which goes beyond the personal and into the public sphere, breaking prohibitory rules or laws, to which legitimate punishments are attached, which needs the involvement of a public authority [11].'

Numbeo which is the website for statistical data, lists Crime index of Ethiopia as 46.74%(forthy six point seven four). It lists crime increasing in the past three years is 68.79%(sixty eight point seven nine) which is getting high.

2.3 Time series Forecasting

Forecasting is used in almost every field for various reasons. It is the act of predicting the future based on the history. Past data or information of a time series are processed withforecasting methods in order to forecast the future outcomes of these data or time series [12]. A time series is a sequential set of data points, measured typically over successive times. It is mathematically defined as a set of vectors $x(t), t = 0,1,2\ldots$ where t represents the time elapsed. The variable x(t) is treated as a random variable. The measurements taken during an event in a time series are arranged in a proper chronological order [13].

For time series data, time series forecasting is used which is, one of prediction technique used to predict events through a sequence of time. It has many social utilities including weather forecasting, Earthquake prediction, Astronomy, Crime prediction, pattern

recognition, signal processing. Time serious forecasting is done using many computer technologies including machine learning, fuzzy logic, gaussian processes and hidden markov models.

Machine learning (ML) is a category of algorithm that allows software applications to become more accurate in predicting outcomes without being programmed [14]. Companies are using machine learning algorithms to improve business decisions, increase productivity, detect disease, forecast weather, and do many more things. We need to understand our data and prepare for the future data with the help of technology. The concept of machine learning is changing data into information. Machine learning is an intersection of statistics, engineering, computer science, and also appears in other disciplines. It can be useful to numerous fields from politics to geosciences. It can be applied for many problems. Any field that needs to understand and turn on data can profit from machine learning techniques.

we can say that machine learning is using algorithms for obtaining structural metaphors from data examples. A computer learns something about the structures that represent the information in the raw data. Structural descriptions are another term for the models we build to contain the information extracted from the raw data, and we can use those structures or models to predict unknown data. Structural descriptions (or models) can take many forms. Each model type has a different way of applying rules to known data to predict unknown data [15].

2.4 Statistical forecasting models

2.4.1 Autoregressive moving Average (ARMA) model

The ARMA model is one of the most widely used models since it combines the advantages of the auto-regressive AR(p) and the moving average MA(q) models [16]. The ARMA model was originally proposed in 1951 by Peter Whittle in his thesis “Hypothesis testing in time series analysis” and was adapted by George E. P. Box and Gwilym Jenkins in 1971 [11]. An ARMA (p, q) model of order (p, q) isdefined by:

$$y_t = \mathrm{c} + e_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i}$$

where yt is the original series and εt is a series of unknown random errors which are assumed to followed the normal probability distribution.

The multivariate version of the ARMA model is called vector auto-regressive moving average (VARMA) which is given by

$$y_t = \mathrm{v} + A_1 y_{t-1} + \cdots + A_p y_{t-p} + u_t - M_1 u_{t-1} + \cdots + M_q u_{t-q}$$

where y_t is the output, y_{t-p} and u_{t-q} are respectively the past outputs variables and the past inputs exogenous variables and A_p and M_q are matrices of parameters.

2.4.2 Autoregressive integrated moving Average (ARIMA) model

The models defined previously as AR, MA, and ARMA are used in stationary time series analysis. A time series is said to be stationary, if the mean of the series and the covariance among its observations do not change over time and do not follow any trend. In practice, most time series are non-stationary, so in order to fit stationary models, it is indispensable to get rid of the non-stationary sources of variation. One solution to this, introduced by Box and Jenkins, is the ARIMA model which generally overcomes this limitation by introducing a differencing process which effectively transforms the non-stationary data into a stationary one. This is done by deducting the observation in the present period from the prior one. The ARIMA model is called "Integrated" ARMA since the stationary model that is fitted to the differenced data that has to be summed in order to deliver a model for the original non stationary data [17]. The general form of the ARIMA (p, d, q) process is described as

$$y_t = \mu + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q}$$

where parameters p, d, and q are non-negative integers that refer to the order of the autoregressive part, the degree of first differencing involved, and the order of moving average part respectively. This capacity to deal with non-stationary process has made the

ARIMA model one of the most popular and widely used approaches in time series forecasting.

2.4.3 Seasonal ARIMA (SARIMA) model

ARIMA is the most widely used forecasting methods for univariate time series data forecasting. Although the method can handle data with a trend, it does not support time serious with seasonal component. An extension to ARIMA that supports the direct modelling of seasonal component of the series is called SARIMA. It adds three new hyper parameters to specify the Auto regression (AR), differencing (I) and Moving average(MA) for the seasonal component of the series, as well as an additional parameter for the period of seasonability. This model is generally termed as the SARIMA (p, d, q)×(P,D,Q)s model. Where p, d and q are trend elements specifically, p is trend auto regressive order, d is trend difference order and q is trend moving average order. On the other hand P, D and Q are seasonal elements and specifically, p is seasonal auto regressive order, D is seasonal difference order, Q is seasonal moving average order and S is the number of time steps for a single seasonal period.

The mathematical formulation of a SARIMA (p, d, q)×(P,D,Q)s model in terms of lag polynomials is given:

$$y_t = \phi_p(B^s)\emptyset_p(B)\Theta_s^D\Theta^D z_t = \theta_q(B)\Theta_Q(B^s)a_t$$

Here,z_t is the seasonally differenced series.

2.5 Neural Network Forecasting Models

Neural networks have been around for 50 years and above. They are also a form of model for machine learning. [18]. The fundamental unit of a neural network is a node, which is loosely based on the biological neuron in the mammalian brain. The connections between neurons are also modeled on biological brains, as is the way these connections develop over time (with “training”). Deep learning arisen from that decade’s explosive computational development as a serious contender in the field, winning many important machine learning rivalries [19]. The interest has not cooled as of 2017; today, we see deep learning mentioned in every corner of machine learning.

There are 3 widely used architectures for time series forecasting and are dealing with long term dependencies namely feed forward neural networks, recurrent neural networks, LSTM and ESNs

2.5.1 Feed Forward Neural Network

These networks are the foundations of most deep learning models like convolutional neural network (CNN) and recurrent neural networks (RNN). They are also known as multilayer perception (MLP). We use these models mostly for supervised machine learning tasks where we already know our target values. They are arranged in layers where taking in input is the first layer and the output is the last layer. The middle layers are hidden layers that has no connection with the external world. extremely important for basis of many commercial applications, such as stock market prediction. Feed forward network can be calculated by substituting the outputs from the first layer into the second layer which is given by:

$$y_l = p_l(w_l p_{1-1}(w_{l-1} x_{l-1} + b_{l-1}) + b_t$$

Where,

p_l, is a vector of activation functions

w_l, is weights from each neuron in a layer

x_l, is the input signal from *l-1* and,

b_l, is an arbitrary offset vector

2.5.2 Recurrent Neural Network

Recurrent neural networks (RNN), a class of nets that can predict the future. They can observe time series data like stock prices, and tell you when to buy or sell. They can anticipate car trajectories and support avoiding accidents in autonomous driving systems. [20]. They can also take sentences, documents, or audio samples as input, making them extremely useful for natural language processing (NLP) systems such as automatic translation, speech-to-text, or sentiment analysis (e.g., reading movie reviews and extracting the rater's feeling about the movie). Furthermore, recurrent neural networks'

skill to anticipate also makes them capable of doing surprising creativity. You can ask to forecast which are the most probably following notes in a melody, then arbitrarily choose one of these notes and play it. Then ask the network for the next most similar notes, play it, and recurrence the process again and again. Before you know it, your net will compose a melody such as the one produced by Google's Magenta project. Similarly, RNNs can produce image captions, sentences, and much more. They can hold the long-term time dependencies in the input data. Recurrent Neural Networks accomplish this because their hidden state captures information from an arbitrarily long context window and does not have the limitation of the other techniques. Moreover, the number of states they can model is represented by the hidden layer of nodes, and these states grow exponentially with the number of nodes in the layer. This makes Recurrent Neural Networks exceptional at capturing a lot of time-dimension relevant information across many input vectors.

A Recurrent Neural Network contains a feedback loop that is used to learn from sequential data, including sequences of fluctuating lengths [21]. Recurrent Neural Networks contain an extra parameter matrix for the connections between time-steps, which are used/trained to capture the temporal relationships in the data. Recurrent Neural Networks are trained to produce sequential output at each time-step depending on both the present input and the input at all previous time steps.

2.5.3 Long short-term memory (LSTM) model

Long short-term memory networks help in preventing errors that comes with backpropagation through time and layers. They allow recurrent neural networks to learn over manty steps by maintaining the constant errors and open a channel to link reasons and effects remotely. This is one of the fundamental challengesto AI and machine learning, since algorithms are often challenged by environments where signals are sparse and delayed.

LSTMs holds information separately from the normal flow of the RNN in a gated cell. Information can be kept in, read or written from a cell, like data in a computer's memory. The cell makes judgements via gates that open and close about,when to allow reads and writes, what to store , and removals. Yet, these gates are analog, applied with element-wise

duplication by sigmoids, unlike the digital storage on computers, which are all in the range of 0-1. Analog has the benefit over digital of being differentiable, and is suitable for backpropagation [22].

Those gates act on the signals they receive, and similar to the neural network's nodes, they block or pass on information based on its strength and import, which they filter with their own sets of weights. Those weights are attuned via the recurrent networks learning process like the weights that control input and hidden states. Then, the cells can learn when to leave,delete, orallow data to enter through the iterative procedure of making deductions, backpropagating error, and regulating weights via gradient descent.

2.5.4 Echo state networks (ESN) model

Echo state networks (ESN) presents the architecture and supervised learning principles for recurrent neural networks (RNNs). The central idea is to drive a random, large, fixed recurrent neural network with the input signal, through inducing in each neuron within this "reservoir" network a nonlinear response signal and join the desired output signal by a trainable linear combination of all of these response signals.

Echo state networks are able to be set up with or without direct trainable input-to-output connections, with or without output-to-reservoir feedback, with different neuron types, different reservoir-internal connectivity patterns, etc. Moreover, the output weights can be computed with any of the available offline or online algorithms for linear regression. In Margin-maximization criteria identified from training support vector machines(SVM) have been used to determine output weights in addition to least-mean-square error solutions (i.e., linear regression weights) [23].

The primary discrete-time, sigmoid-unit ESN with N reservoir units, K inputs, and L outputs are governed by the state update equation

$$x(n+1) = f(Wx(n) + Winu(n+1) + Wfby(n))$$

where, x(n) is the N-dimensional reservoir state, f is a sigmoid function W is the N×N reservoir weight matrix, Win is the N×K input weight matrix, u(n) is the Kdimensional input signal, Wfb is the N×L output feedback matrix, and y(n) is the L-dimensional output

signal. In tasks where no output feedback is required, Wfb is nulled. The extended system state z(n)=[x(n);u(n)] at time n is the concatenation of the reservoir and input states. The output is acquired from the extended system state by

y(n)=g(Woutz(n)), where g is an output activation function (typically the identity or a sigmoid) and Wout is a L×(K+N)-dimensional matrix of output weights.

In the state harvesting stage of the training, the ESN is driven by an input sequence u(1),…,u(nmax) , which yields a sequence z(1),…,z(nmax) of extended system states. The system equations (1), (2) are used here. If the model includes output feedback (i.e., nonzero Wfb), then during the generation of the system states, the correct outputs d (n) (part of the training data) are written into the output units ("teacher forcing"). The obtained extended system states are filed row-wise into a state collection matrix S of size nmax × (N+K) . Usually some initial portion of the states thus collected are discarded to accommodate for a washout of the arbitrary (random or zero) initial reservoir state needed at time 1. Likewise, the desired outputs d(n) are sorted row-wise into a teacher output collection matrix D of size nmax×L .

The desired output weights Wout are the linear regression weights of the desired outputs d (n) on the harvested extended states z(n). A mathematically straightforward way to compute Wout is to invoke the pseudoinverse (denoted by ·†) of S : (3) Wout=(S†D)′ , which is associate degree offline rule (the prime denotes matrix transpose).Linear signal processingmethods like online adaptive methods, can also be used to calculate output weights.

2.6 Hybrid Models

Using artificial neural networks for time series forecasting has unlocked the door to new hybrid models. Hybrid models includes a combination of diverse methods from machine learning and typical statistical models [24]. In the literature, diverse combinations of techniques have been proposed in order to overcome the limitations of single models. The idea behind the mixture of different models in forecasting is to build a strong architecture that will use the unique features of each model in order to capture different patterns in the data. A good example is the combination of ARIMA and ANNs where an ARIMA process

combines three different processes comprising an AR function regressed on past values of the process, MA function is a part to make the data series stationary by transforming them in order to deal with the non-stationary linear section while the neural network model deals with nonlinearity [25].

In recent years, several models combining fuzzy inference system (FIS) with neural networks and others methods have been proposed. A fuzzy inference system is the process of mapping a given input to an output using fuzzy logic (FL). FL provides a easy way to reach at a definite decision based upon ambiguous, loose, missing input or noisyinformation and it is a problem-solving control system methodology. It can be designed from expert knowledge or from data. Castillo and Merlin described the application of several neural network architectures to the problem of simulating and predicting the dynamic behavior of complex economic time series. Damousis and Dokopoulos presented a fuzzy expert system that forecasts the wind speed at a wind energy conversion system and implement two genetic algorithms for comparing the fuzzy expert system. Potters and Negnevitsky also describe in their paper an adaptive neuro-fuzzy inference system (ANFIS) for short term wind forecasting in Tasmania. Kasabov et al. developed online evolving system for dynamic time series prediction.

2.7 Related Work

As an inevitable and important social problem of our societies, criminal activities have been traditionally studied in different fields. The recent movements of open data have made large repositories of crime activity data become publicly accessible, which provides a unique opportunity to start studying crime in a more systematic manner. Along with the recent advances in Big Data analytic techniques, we are able to extensively analyze and deeply understand crime patterns. In the current literature, researchers have devoted their attention on analyzing criminal activities mainly from place-centric perspectives.

Alexander Stec and Diego Klabjan (2018) take advantage of deep neural networks, including variations that are suited to the spatial and temporal aspects of the crime prediction problem. In order to address the temporal aspects of crime prediction, a

Recurrent Neural Network (RNN) is used. For the spatial aspects of crime prediction, a Convolution Neural Network (CNN) is fitted [26].

Lawrence McClendon and Natarajan Meghanathan (2015) used WEKA to conduct a comparative study between the violent crime patterns from the Communities and Crime Unnormalized Dataset. They implemented Linear Regression, Additive Regression, and Decision Stump algorithms using the same finite set of features, on the Communities and Crime Dataset. Overall, the linear regression algorithm performed the best among the three selected algorithms [27].

Getahun Tesemma(2017) focused on finding the nexus between urban poverty and property crimes which were committed in Addis Ababa city. Explanatorycross-sectional study was carried out in the ten sub-cities of Addis Ababa. Analysis was done using SPSS version 24. The findings characterize the offender respondents as poor and commission of property crime was their coping strategy for survival [28].

Adewale Opeoluwa Ogunde1 et al. (2017) applied classification rule mining method to develop a system for detecting crimes in universities. WEKA mining software along with the Iterative Dichotomiser 3 (ID3) decision tree algorithm is used to analyze and train the data. The model acquired was used to create a system that showed the unseen relationships among the crime-related data, in the form of decision trees. The developed system could effectively predict a list of possible suspects by simply analyzing data retrieved from the crime scene with already existing data in the database [29].

Ying-Lung Lin et al.(2018) proposed a model that establishes a range of spatial-temporal features by integrating the notation of a criminal environment in grid-based crime prediction modeling. The Deep Neural Networks model is used, which outperforms the popular Random Decision Forest, Support Vector Machine, and K-Near Neighbor algorithms. The F1-score increases about 7% on 100-by-100 grids. Experiments demonstrate the need for the geographic feature design for increasing performance and descriptive ability. Furthermore, testing for crime dislocation also shows that the model design outperforms the baseline [30].

Mona Mowafy et al. (2018) proposed methods and techniques that can predict the crime type from unstructured text. The paper, focuses on how practically building an unstructured crime data prediction model using the scikit-learn Python Toolkit. The main goal of the structured model is to predict the crime type, based on the unstructured data of the crime occurrence reports for improving the policeman decisions and lessen their investigation energies [31].

Dr.M.Sreedevi et al. (2018) focused mainly on crime factors. By Using the concept of Data Mining, they extract previously unknown useful information from an unstructured data. They approach between criminal justice and computer science to create a data mining practice that can assist solve crimes quicker. For classification purpose Naive Bayes Algorithm which is supervised learning method is used. The algorithm categorizes a news article into crime type to which it fits the best. Apriori Algorithm is used to identify trends and patterns in crime. And prediction is done using the decision tree concept [32].

Aarathi Srinivas Nadathur et al. (2018) offers a extensive overview of crime occurrences and their relevance in literature by combining approaches. At first, the paper analyses and finds features of crime occurrences proposing a combinatorial incident description schema. The proposed plan aims to increase a chance to systematically join different elements, or crime characteristics. Additionally, a comprehensive list of crime-related offences is put forward. This enables a thorough understanding of the repeating and underlying criminal activities. This paper intends to be of use to specialists and law enforcement officers in discovering patterns and trends for making forecasts, finding relationship and possible explanations [33].

Ms. Vrushali Pednekar et al. (2018) classify clustered crimes based on occurrence frequency during different years. Data mining is used in the investigation, analysis and detection of patterns for incidence of different crimes. Various clustering approaches of data mining are used to study the crime data. The K-Nearest Neighbor (KNN) classification is used for crime prediction. The proposed system focuses on finding the most criminal hotspots. It focuses on crime factors of each day instead of focusing on causes of crime incidence likepolitical enmity etcit or criminal background of offender [34].

Prajakta Yerpude and Vaishnavi Gudur (2017) used data mining techniques that are applied to crime data for predicting features that affect the high crime rate. Unsupervised learning splits an unstructured,unreliable data into classes or clusters while supervised learning gets the anticipated output by using datasets to train, test on them. Naïve Bayes,Decision treesand Regression are some of the supervised learning methods in data mining and machine learning on previously collected data and thus used for predicting the features responsible for causing crime in a region or locality. Based on the rankings of the features, the Crimes Record Bureau and Police Department can take actions to decrease the probability of occurrence of the crime [35].

Trung T. Nguyen et al. Sung described a crime predicting method which forecasts the types of crimes that will occur based on location and time. In the proposed method forecasting is done for the jurisdiction of Portland Police Bureau (PPB). Data from various public sources used to forecast the crime type in a particular location over a period of time using different machine learning algorithms including Support Vector Machine (SVM), Random Forest, Gradient Boosting Machines, and Neural Networks. With their first dataset in which demographic was used as it is, SVM does not seem to be a suitable model for this task because of the bad classification accuracy when compared to other used methods. Random Forest or Gradient Boosting turned out to be the two best models. With the second dataset that used demographic data as reference to generate missing features, SVM and Neural Network show the best accuracy models when compared to other two ensemble methods [36].

Hyeon-Woo Kang and Hang-Bong Kang proposed a feature-level data fusion method with environmental context based on a deep neural network (DNN). Their dataset consists of data collected from various online databases of crime statistics, demographic and meteorological data, and images in Chicago, Illinois. They select crime-related data by conducting statistical analyses. They train DNN, which consists of spatial, temporal, environmental context, and joint feature representation layers. Coupled with crucial data extracted from various domains, fusion DNN is a product of an efficient decision-making process that statistically analyzes data redundancy. Experimental performance results show

that DNN model is more accurate in predicting crime occurrence than other prediction models [37].

Suhong Kim et al. proposed machine-learning-based crime prediction. Vancouver crime data for the last 15 years is analyzed using two different data-processing approaches. Machine-Learning predictive models, K-nearest-neighbour and boosted decision tree, are implemented and a crime prediction accuracy between 39% to 44% is obtained when predicting crime in Vancouver [38].

Mami Kajita and Seiji Kajita(2018) develop an algorithm that finds a causality of events only by learning a spatio-temporal dataset, employing a Green's function scheme. The data is taken from open data of burglaries in Chicago, and it has a good prediction accuracy higher to or similar to standard methods of crime prediction. They find a cascade influence of crimes that has a long-time, logarithmic tail; the result is reliable with an earlier study on burglaries. The current method is a powerful tool not only to predict events but also to reveal their hidden connection [39].

Somayeh Shojaee et al. conducted an experiment to obtain better supervised classification learning algorithms to predict crime status by using two different feature selection methods tested on real dataset. Comparisons in terms of Area Under Curve (AUC), that Naïve Bayesian (0.898), k-Nearest Neighbor (k-NN) (0.895) and Neural Networks (MultilayerPerceptron) (0.892) are better classifiers against Decision Tree (J48) (0.727), and Support Vector Machine (SVM) (0.678). Furthermore, the performance of mining results is improved by using Chi-square feature selection technique [40].

Jonathan Albert U. Tumulak and Kurt Junshean P. Espinosa presents a simple approach to predict crime in a geographic space using grid thematic mapping and neural networks. The study particularly focuses on the possibility of using historical crime data in predicting future crime incidents. The dataset is divided into monthly, weekly, and daily data called as a snapshot for training the model. The study area, Cebu City, is divided into a square grid of various sizes and crime incidents, from each cell in the grid of each snapshot, will then be recorded. This data is then fed into the neural network to predict possible areas where crime would happen for the next time interval. Initial results have shown that the model is accurate enough to predict crime areas when the data snapshot is divided monthly

and weekly and grid size is set to a large size of about 1000 by 1000 meters and 750 by 750 meters. The best model was able to yield an F1 score of 0.95. Although the created model is simple, this can be a stepping stone to further crime modelling and prediction studies. This tool can be used to aid decisions and policy-making in the deployment of police resources throughout the city [41].

No.	Research Group	Objective	Detection	LSTMRNN Methods	Remark
1	Alexander Stec& Diego Klabjan	Address the temporal and spatial aspects of crime prediction	RNN & CNN	NO	They took the advantage of deep neutral network to address the temporal aspects and spatial aspects by using RNN & CNN respectively.
2	Lawrence McClendon and Natarajan Meghanathan	Conduct a comparative study	Linear Regression, Additive Regression, and Decision Stump	NO	The linear regression algorithm performed the best among the three selected algorithms
3	GetahunTesemma	Finding nexus between urban poverty and property crimes	Descriptive cross-sectional study & SPSS version 24	NO	Descriptive cross-sectional study was used to study 10 sub-cities of A.A and analysis was done using SPSS version 24.
4	AdewaleOpeoluwaOgunde et.al.	Develop system for detecting crime in universities	Classification rule mining method & Iterative dichotomiesr 3(ID3)	NO	Applied classification rule mining method to develop a system for detecting crimes and ID3

					decision tree algorism to analyze and train data. The system could effectively predict a list of possible suspects.
5	Ying-Lang Lin et.al	Incorporate the concept of a criminal environment & stablish spatial temporal features	Deep Neural Network	NO	DNN is used, which out performs popular random decision forest, Support vector machine and k-near neighbor algorithms.
6	Mona Mowafy et.al	Predict the crime type from unstructured text	Scikit-learn phyton toolkit	NO	Building an unstructured crime data prediction model using scikit-learn toolkit.
7	Dr.M.sreedevi et.al.	Solve crimes easly	Naïve data algorithm &Apriori algorism	NO	Focused mainly on crime factors. For classification purpose Naïve Bayes Algorithm was used and Apriori Algorithm is used to identify trends and patterns in crime.
8	Aarathi Srinivas Nadathur et al.	Systematically combine different elements or crime characteristics	HADOOP & HDFS & YARN	NO	used to specialists in discovering patterns for making forecasts, finding relationship and

					possible explanations
9	Ms. VrushaliPednekar et.al	Classify clustered crimes	Data mininig, k-Nearest Neighbor (KNN)	NO	This system can predict regions w/c have high probability for crime rate and forecast crime pron areas. KNN is used for crime prediction.
10	PrajaktaYepude&VaishnaviGudur	Predict features that affect high crime rate	Decision trees, naïve bayes, & regrstion	NO	Data mining techniques are applied to crime data for predicting features that affect the high crime rate
11	Trung T. Nguyen	Describe a crime predicting method which forecasts the types of crimes that will occur based on location and time.	SVM, Random Forest, Gradient Boosting Machines, and Neural Networks	NO	After generating missing features, SVM and Neural Network show the best accuracy models than others.
12	Hyeon-Woo Kang and Hang-Bong Kang	proposed a feature-level data fusion method with environmental context	Deep neural network (DNN)	NO	Results show that DNN model is more accurate in predicting crime occurrence.
13	Suhong Kim et al.	Investigates machine-learning-based crime prediction.	KNN and boosted decision tree	NO	Obtain crime prediction accuracy between 39% to 44%.
14	Mami Kajita &Seiji Kajita	Find a causality of events to open data of burglaries in Chicago	Data-Driven Green's Function (DDGF), Expectation Maximization	NO	Find a causality of events using DDGF, EM & PHM and choose DDHF

			(EM) & Prospective Hotspot Maps (PHM)		
15	Somayeh Shojaee et al.	Obtain better supervised classification learning algorithms to predict crime status	Naïve Bayesian, k-NN, MLP, Decision Tree, and SVM.	NO	Conclude that Naïve Bayesian, k-NN and MLP are better classifiers against Decision Tree, and SVM.
16	Jonathan Albert U. Tumulak and Kurt Junshean P. Espinosa	presents a simple approach to predict crime in a geographic space using grid thematic mapping and neural networks.	NN	NO	Neural network model is proposed to predict crime and yield an F1 score of 0.95

Table 2.7-1: Summary of related works

Chapter 3

RESEARCH METHODOLOGY

3.1 Introduction

The objective of this research is to apply LSTM RNN model on crime data to predict crime and test model's accuracy by comparing it with actual data. Here in this chapter, methodologies followed are described comprising the study area, source data, methods of data collection, and preprocessing works. It also explains about the model and how it has been chosen. We have reviewed literatures and organizations that developed crime prediction.

A general procedure for crime prediction using LSTM RNN can be divided into four steps, which will be presented here, and discussed in detail in the following sections.

1. Data collection: collecting information from all the relevant sources to find answers to the research problem
2. Data pre-processing: transforming raw data in a useful and efficient format
3. Split data: splitting the data into training set and test set
4. Develop prediction model: LSTM model by analyzing training set

3.2 Description of study area

This study is conducted in Addis Ababa specifically Bole sub city, which is the seat for many international and local organizations. It holds 26 embassies, 57 hotels, 64 public parks which are administered under religious authority and 63 religious institutions are found in this Sub-City. It has fourteen woreda in it. According to its profile prepared by Addis Ababa city government, Bole sub city lies in 122.08 (one hundred twenty-two and 8/100) square kilometer of area. Its population is 328,900 (three hundred twenty-eight thousand nine hundred) where 154,542(one hundred fifty-four thousand five hundred forty-two) of it is male and the rest 174,358 (one hundred seventy-four thousand three hundred fifty-eight) is female. Its population density per square meter is 2694.1(two thousand six hundred ninety-four and 10/100). Bole sub city has a higher amount of international and local organizations like embassies and airport compared to other sub

cities. And it should be more secured as they are most significant instruments of the government in political, social and economic perspective.

3.3 Description of study data

The crime data used in this study comprises cheque fraud, violence, drug and theft crime cases that occurred in five years of time starting from January 1, 2014 till the end of 2018 GC. The data is supported by bole subcity police departement. Each case record contains several attributes including date time of a crime, type of crime and locations where the crime happens, which is originally recorded by police officers.

3.4 Data collection

Large amount of data is collected from bole subcity police departement. Originally informations of the corresponding crime is recorded when the individual is caught by the police. In fact, all police men are provided with a centralized form or record format that should be filled when crime is commited or when the offender is caught. This record format holds crime records that are stored manually which are waiting to be entered in to automated system. Eventhough, large amount of data is needed to train a neural network algorithm, due to time constraint to enter the data, top 6033(six thousand thirty three) records are taken for this research. We choose five crime types(assault, cheque fraud, violence, drug and theft), that happened more frequently than other crime types from all crime types.

3.5 Data Preprocessing

One of the critical problems of training machine-learning models is the data itself. The preprocessing holds a series of steps with the goal of compressing the input data and transform it in to representation suitable for applying in the specified model. It’s a process that combines data cleaning, data integration and data transformation. It aims to reduce some incomplete and inconsistent data. The selected machine-learning model then uses the results of preprocessed data.

3.5.1 Missing value

The first problem that we have to deal with is missing values. We cannot erase a row with missing values because it might hold necessary information in it. The idea behind handling

missing value is to use the mean of a column. Imputer, which is a class in sklearn.preprocessing is used to do this task.

3.5.2 Encoding categorical data

Before we run in to a model, we need to change our data in to model understandable format of numerical data. Label encoder, which is also a class in sklearn.preprocessing, is used to do this task. It encodes categorical data in to numerical variables.

3.5.3 Split the data

In order to train our machine-learning model we need to create a training set and in order to test our model we need to create test set. For this we used the library from sklearn.cross_validation and the class train_test_split.

3.5.4 Normalization

Making every data point to have the same scale is the goal of normalization. This helps for each feature to be equally vital. We used MinMax normalization technique which is one of the most common ways to normalize data. Where the minimum value gets transformed into 0 and maximum value gets transformed in to 1. The formula of minmax normalization is as follows:

$$x_n = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Where, $X_n$is normalized value of a variable X

$X_i$is current value of variable X

$X_{Min}$is minimum value of variable X and,

$X_{Max}$is the maximum value of variable X

3.6 Prediction Model

This section describes the structure and learning method of our model. We employed RNN using LSTM for prediction.

RNN model prediction with LSTM

We decided to use LSTM model cause our data is time series and LSTM is very good at holding long term memories. The prediction in a sequence of test samples can be influenced by an input by an input that was given many time steps before. This network can store or release memory in the go through the gating mechanism.

RNN are networks with loops in them, which allows information to persist. In a traditional neural network, we assume all inputs and outputs are independent of each other. LSTM is an RNN that remembers values over arbitrary intervals. It is well suited to classify and predict time series data of unknown duration.

The structure of RNN is similar to hidden marcov model. The difference is how parameters are calculated and constructed.

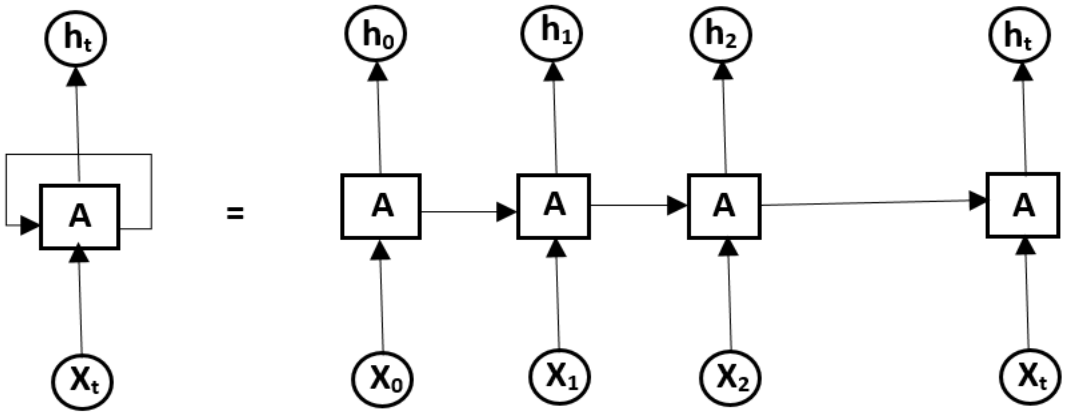


Figure 3.6-1:Recurrent Neural Network (RNN) [42].

RNN cell takes two inputs that are the outputs from the last hidden state and observation at time t. in the hidden state.

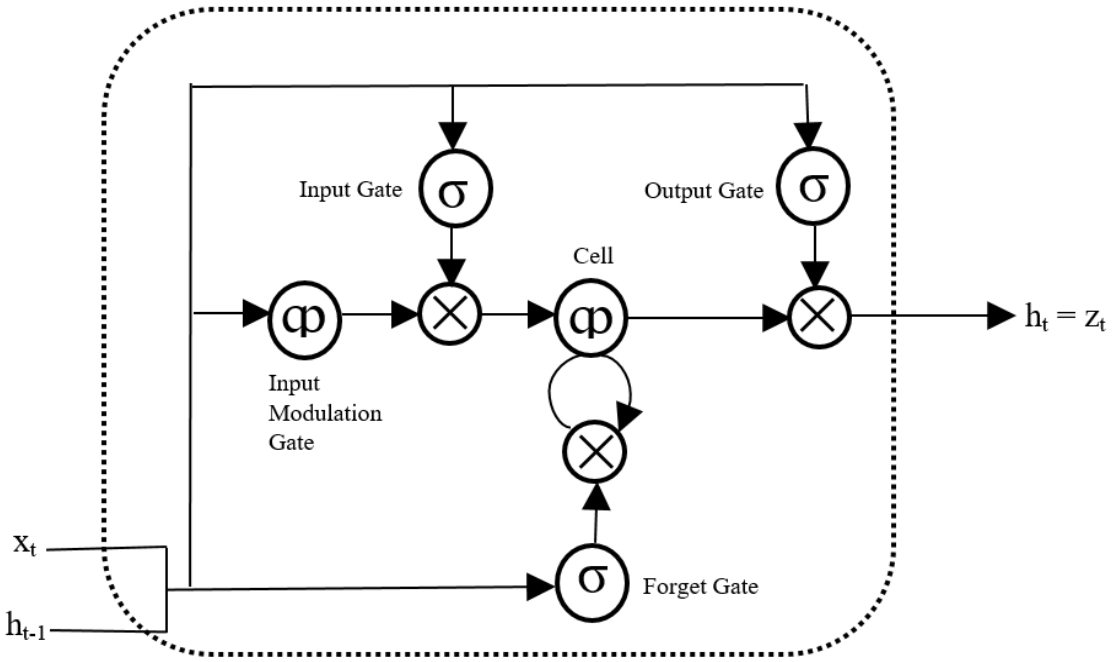


Figure 3.6-2:Long Short-Term Memory Network (LSTM) [43].

Cell state is usually called long-term memory. The looping arrows that allows information from previous to be stored in LSTM cell indicate the recursive nature of the cell. The cell state equation is as follows:

$$C_t = f_t * C_{t-1} + i_t * \hat{c}_t$$

The forget gate modifies the cell state and adjust by the input modulation gate. It realizes that there might be change in the context after encounter its first loop. Its other function is it compares with the current input X_t.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_t)$$

The input gates determine which information should enter the cell state. The input gate has a range of [0, 1] and is a sigmoid function. The sigmoid function will only add memory and not to be able to remove/forget memory.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$\hat{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

Output gate that is usually called focus vector highlights which information should go to the next hidden state.

$$O_t = \sigma(w_0 * [h_{t-1}, x_t] + b_0)$$

$$h_t = O_t * \tanh(C_t)$$

3.7 Tools

3.7.1 Anaconda

Anaconda is a distribution of the python and R programming languages for data science, machine learning, predictive analytics etc… It comes with more than 1500 packages as well as virtual environment manager and conda package. It aims to simplify package management and deployment. Anaconda provides tools needed to easily:

- Collect data from databases, files, and data lakes
- Manage environments conda
- Deploy projects with a single click
- Share, collaborate on and reproduce projects

Anaconda has a desktop graphical user interface (GUI) called Anaconda Navigator. This navigator allows users to launch applications and manage conda packages and so on.

Applications that are available by default navigator are:

- Jupiter Lab
- Jupiter Notebook
- QtConsole
- Spyder
- Gluviz
- Orange
- Rstudio
- Visual Studio Code

3.7.2 Jupyter Notebook

Exists to develop open-source software, open-standards, and services that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It is used to data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. Some of Jupyter Notebook features are:

- supports more than 40 programming languages, including R, Python, Julia, and Scala.
- Notebooks can be shared with others using email, Dropbox, GitHub and the JupyterNotebook Viewer.
- Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.
- Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, TensorFlow.

3.7.3 Python

Is a high-level, object-oriented programming language with energetic semantics. The purpose behind using Python is that Python is:

- Easy to learn quickly, so everyone can start programming.
- Rich, and easy to understand program code.
- More like a readable, human language than like a low-level language. This gives you the ability to program at a faster rate than a low-level language will allow you.
- allows you to create data structures that can be re-used, which reduces the amount of repetitive work that you'll need to do
- open-source and free.
- runs on all operating systems like Microsoft Windows, Linux, and Mac OS X.
- has support community with many web sites, mailing lists, and netnews groups that attract a large number of well-informed and cooperative contributes.
- doesn't have pointers like other C-based languages, making it much more reliable [44].

3.7.4 Pandas

Is an open source library that offers easy access and high-performance data structures and analysis tools for python programming language. Here are some of pandas library highlights [45]:

- Fast and well-organized Dataframe object for data management with unified indexing;
- Tools for reading and writing data of different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format;
- Intellectual data arrangement and integrated treatment of missing data: increase in automatic label-based alignment in computations and easily manipulate disorganised data into an organized form;
- Flexible reshaping and turning of data;
- Intellectual label-based slicing, indexing, and sub setting of large datasets;
- Columns can be inserted and deleted from data structures for size mutability;
- Combining or converting data with a powerful set by engine allowing split-apply-combine actions on datasets;
- High performance integration and linking of datasets;
- Time series-functionality: date range generation and frequency conversion, date shifting and lagging. Creating domain-specific time offsets and join time series without losing data;
- Highly enhanced for performance, with critical code paths written in Cython or C.

3.7.5 TensorFlow

Is an open source platform for machine learning. It has a flexible, complete system6i6i of libraries, community resources and tools and that lets researchers and developers to easily build and deploy ML powered applications [46]. It has the following features:

- Easy model building: Build and train machine learning models easily using spontaneous high-level APIs like Keras, which makes for immediate model iteration and easy debugging.

- TensorFlow ecosystem: delivers a group of workflows to develop and train models using Python, JavaScript, or Swift.
- Robust ML production anywhere: Easily train and deploy models in the browser, cloud, or on-device.
- Powerful experimentation for research: Easy and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

3.7.6 Keras

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow [47].

- It was developed to make implementing deep learning models as fast and easy as possible for research and development.

- It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.
- Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:
    - **Modularity**: A model can be understood as a sequence or a graph alone.
    - **Minimalism**: The library provides just enough to achieve an outcome, no frills and maximizing readability.
    - **Extensibility**: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.
    - **Python**: No separate model files with custom file formats. Everything is native Python.

Chapter 4

DESIGN AND MODELLING

4.1 Data Analysis

Historical crime records have been a main element for crime prediction. Many studies have found that future crimes often occur in vicinity of some specific areas. Each incident is recorded with detailed information about its time, location and crime type. In this study we focus on using Bole sub city crime dataset. Table 1 shows the sample dataset we used for the research.

Date	Time	Crime Type	Location
1/1/2014	2:00	Creating Violence	Ruwanda
1/6/2014	23:00	Assault	Ruwanda
1/7/2014	23:00	Assault	CMC Michael
1/8/2014	23:00	Check Fraud	Atlas Area
1/9/2014	23:00	Assault	Imperial
1/10/2014	19:00	Creating Violence	Asama Erbata
1/11/2014	22:00	Check Fraud	Weja
1/12/2014	19:00	Creating Violence	Bras Hospital
1/13/2014	23:00	Check Fraud	Atlas Area
...	…	...	...
12/21/2018	18:00	Theft	49 Mazoria
12/22/2018	6:00	Creating Violence	Meri
12/24/2018	5:00	Drug	Gurd Shola
12/26/2018	7:00	Drug	24 Area
12/27/2018	21:00	Check Fraud	Atlas Area
12/28/2018	4:00	Check Fraud	Atlas Area
12/29/2018	5:00	Assault	CMC Michael
12/30/2018	1:00	Creating Violence	22 Area
12/31/2018	9:00	Creating Violence	Ayat Area

Table 4.1-1: Sample of Crime Data

Table 4.1-2and 4.1-3 shows the related statistics about crime data in top 15 crime occurring areas and crime types respectively

Area	Incident Number	Percentage (%)
Atlas Area	598	12.52%
48 Mazoria	491	10.28%
Weja	205	4.29%
Goro	175	3.66%
Jackros	168	3.52%
Meri	147	3.08%
24 Area	141	2.95%
WhaLmat	137	2.87%
Urael	115	2.41%
SaliteMeheret	115	2.41%
Ednamall Area	112	2.34%
Gurd Shola	110	2.30%
Sunshine	107	2.24%
Lemmi Industry	101	2.11%
Altad Area	90	1.88%

Table 4.1-2: Statistics of crime data in different areas

Crime Type	Incident Number	Percentage
Check Fraud	1517	31.75%
Creating Violence	1237	25.89%
Drug	1319	27.61%
Theft	705	14.76%
Grand Hotel	4778	100.00%

Table 4.1-3:Statistics of crime data in different types

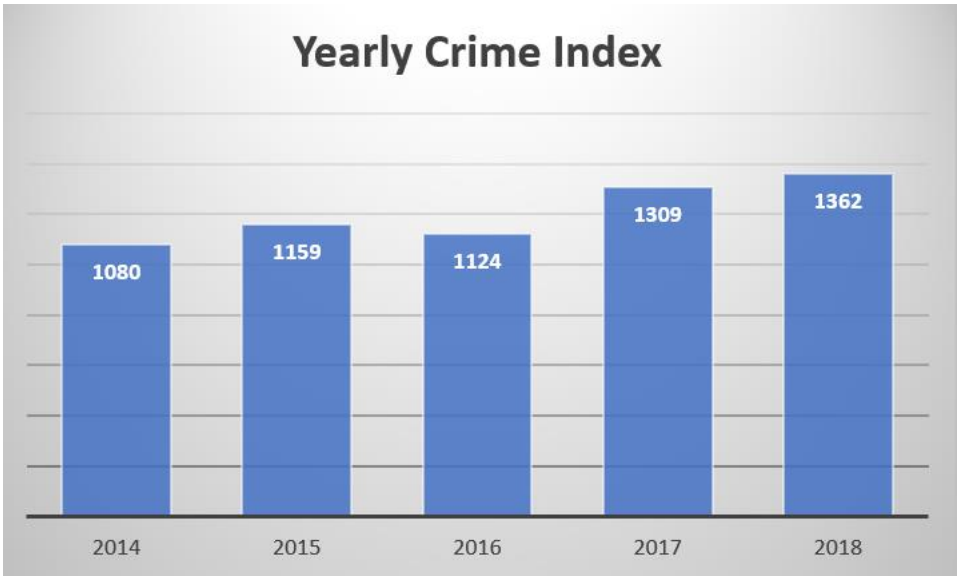


Figure 4.1-1:Yearly Crime Index

Figure 4.1-2:Monthly Crime Index

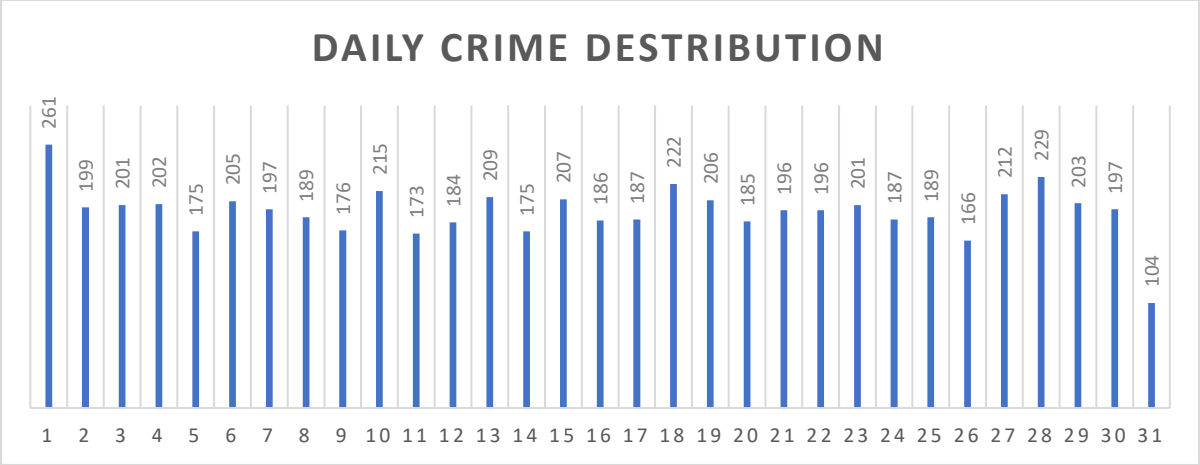
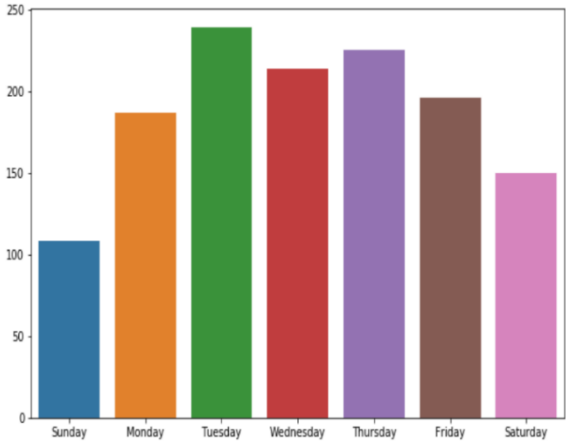


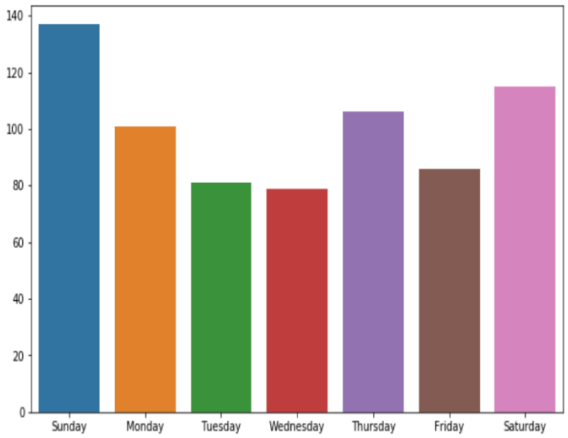
Figure 4.1-3:Daily Crime Index

Certain crimes tend to happen more often during the weekend. Others like drug show a notable drop on Sunday. The chart below shows the crime patterns during the week.

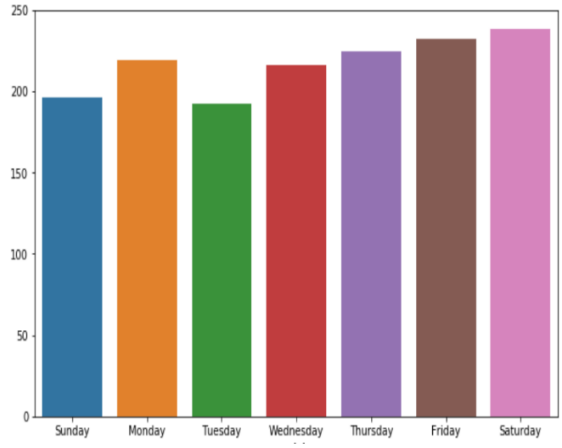
Drug



Theft



Fraud



Violence

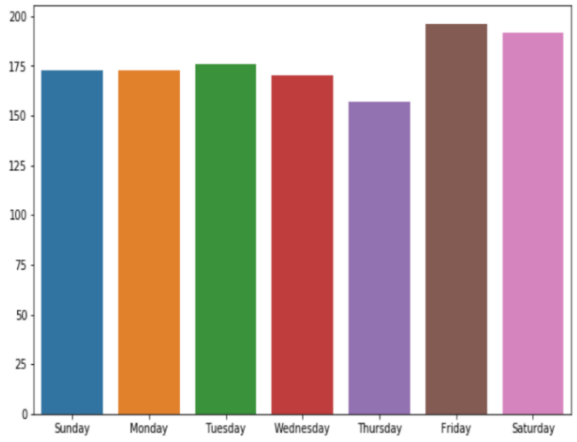


Figure 4.1-4:weekday crime counts per category:

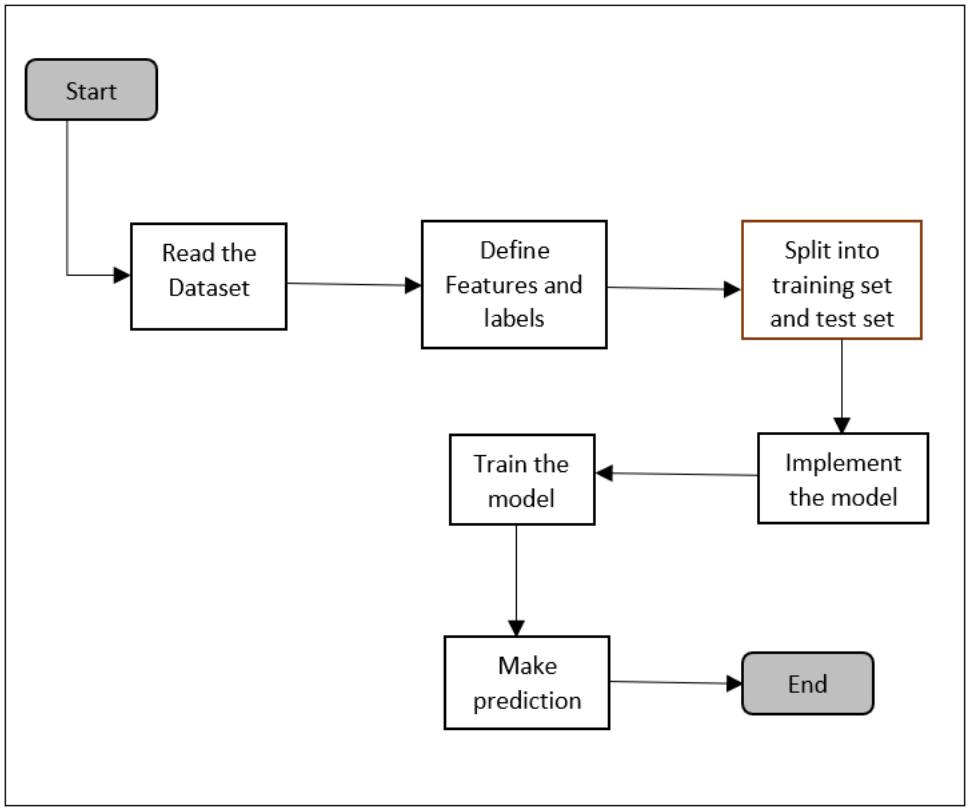


Figure 4.1-5: Process flow diagram

The above figure shows the algorithm flow of the research. It has mainly five phases: initialization phase, data pre-processing phase, model implementation phase and model training phase.

4.2 Initialization Phase

In initialization phase, all the necessary libraries and packages are installed through conda.Libraries and packageslike:NumPy, Pandas, Matplotlib, Sckit-Learn, TensorFlow, keras. We imported Pandas NumPy and pandas to our Jupiter notebook and feed our csv data into pandas Dataframe through the function called read_csv.

Matplotlib library is used to plot 2D figures in a variety of formats. It makes it easy to visualize data cause a picture worth a thousand words. It's a very powerful plotting library useful for peoples using python and NumPy. Pyplot is the most used module of matplotlib. It provides an interface like MATLAB but instead it uses Python. Matplotlib figure can be categorized into four parts: Figure, Axes, Axis and Artist. Figure contains one or more Axes/ plots and it's a whole figure. Axes or plots contains two or three Axis objects. It has

title, x-label and y-label. Axis take care of generating the graph limits. Artist is anything we can see on the figure like texts, line2D and collection objects.

Sckit-Learn is one of the best-known python libraries which provide solid implementations of range of machine learning algorithms. Sckit-learn is built upon the SciPy (scientific python) which has to be installed before installing Sckit-learn. This library has pre-built functionality in sklearn.preprocessing. sklearn.preprocessing provides common utility functions and transformer classes to change raw data into representational form that is more suitable for the models. The other module that we imported from Sckit-learn library is sklearn.metrics. This module that holds score, functions, performance metrics and pairwise metrics and distant computations.

The proposed model will be done using TensorFlow backend. TensorFlow is an open source software library for numerical tensor computation which is developed from Google Brain team. Huge companies like Google, NVidia, Uber and many more uses TensorFlow due to its flexible system architecture. It supports many types of devices including desktop, mobiles and some embedded devices. TensorFlow has an API for Python, JavaScript, Java, GO and swift. It allows for easy deployment of computations across a variety of platforms like CPUs, GPUs, and TPUs. It provides strong support for machine learning and deep learning.

Keras is a neural network which is capable of running over one of the backend engines like CNTK, TensorFlow or Theano. It allows easy and fast prototyping, supports both recurrent and convolutional networks and combinations of two it also runs seamlessly on CPU and GPU [48]. We can deploy keras on any device like iOS with CoreML, Android with TensorFlow Android, cloud engine, web browser, Raspberry pi. This API handles models, defining layers, or setup multiple input-output models. It can compile our model with loss and optimizer functions.

4.3 Data Pre-processing Phase

In the data pre-processing phase, we transform raw data into machine understandable format. Raw data are mostly incomplete: they lack attribute values or attributes of interest, noisy: they contain errors and outliers, inconsistent: they contain discrepancies in codes.

After importing libraries and the raw data, we check out for the missing values which is none and in the next step we change the categorical values. As we can see from our data in table 4.3-1, we have a categorical feature such as crime Type and neighborhood. Label encoding provides very efficient tool for encoding labels of categorical data to numeric values. Then the numerical values are converted to float by using astype(float) method.

Minmax scaling is considered as a way to normalize data in which we scale each feature to a fixed range. It is probably the most famous scaling algorithm. In this scaling technique values will end up ranging from 0 to 1 by shifting and rescaling. This is done by subtracting the minimum value and dividing by the maximum value minus the minimum values. [49]

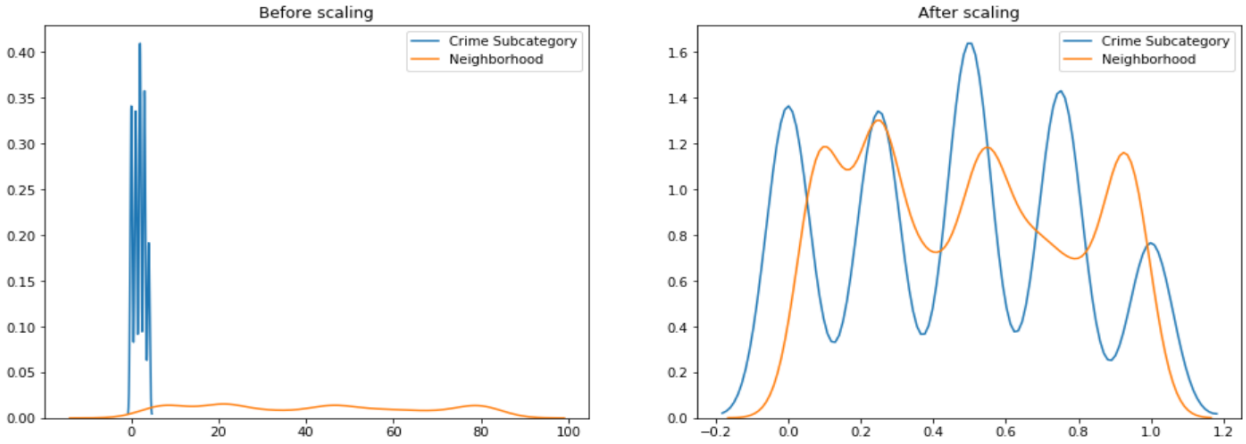


Figure 4.3-1:Normalized data

When scaling the values between zero to one, the categorical values that are changed to numeric values will also be scaled too. Table 4.3-1 and 4.3-2 will present the scaled and categorical values of crime type and location respectively

Crime type	
Categorical	**Scaled**
Assault	0
Check fraud	0.25
Creating violence	0.5
Drug	0.75
Theft	1

Table 4.3-1: Scaled crime types

Location		
No.	Categorical	Scaled
1	17 Health Center	0
2	19 meznagna	0.012987
3	22 Area	0.025974
4	23 Kebele	0.038961
5	24 Area	0.051948
6	48 Mazoria	0.064935
7	49 Mazoria	0.077922
8	93 Mazoria	0.090909
9	Addisu Asphalt	0.103896
10	Airlines	0.116883
11	Alem bank	0.12987
12	Altad Area	0.142857
13	Amche	0.155844
14	Amen Bldg	0.168831
15	AnbesaGaraj	0.181818
16	Anwar Mosque	0.194805
17	Arabsa	0.207792
18	ArogeGumruk	0.220779
19	Asama Erbata	0.233766
20	Atlas Area	0.246753
21	Awraris Hotel	0.25974
22	Ayat Area	0.272727
23	Beshale	0.285714
24	Bole Homes	0.298701
25	Bole Michael	0.311688
26	Bole School	0.324675
27	Bras Hospital	0.337662
28	Bulbula	0.350649
29	Chefe	0.376623
30	ChereqaSefer	0.38961
31	Chichinia	0.402597
32	CMC Michael	0.363636
33	Dldy	0.415584
34	Downtown	0.428571
36	Ednamall	0.441558
37	EgziabherAb Church	0.467532
38	Figa	0.480519
39	Friendship	0.493506
40	Gabon Embassy	0.506494
41	Gerji	0.519481
42	Goro	0.532468

Location		
No.	Categorical	Scaled
43	GrarSefer	0.545455
44	Gurd Shola	0.558442
45	Imperial	0.571429
46	Jackros	0.584416
47	JafarMusque	0.597403
48	Karamara	0.61039
49	Kasanchis	0.623377
50	Lemmi Industry	0.636364
51	Mariam Area	0.649351
52	Mebrathayl	0.662338
53	Megenagna	0.675325
54	Meri	0.688312
55	MeskelAdebabay	0.701299
56	Millinium	0.714286
57	Moenco	0.727273
58	Nyala Motors	0.74026
59	Roba Bakery	0.753247
60	Ruwanda	0.766234
61	Safeway Supermarket	0.779221
62	SaliteMeheret	0.792208
63	Samit	0.805195
64	Samit Condominium	0.818182
65	Sefera area	0.831169
66	SefereSelam	0.844156
67	SelamAmba	0.857143
68	ShalaMenafesha	0.87013
69	Sheger Homes	0.883117
70	Shola	0.896104
71	Sunshine	0.909091
72	Unknown	0.922078
73	Urael	0.935065
74	Weja	0.948052
75	WelloSefer	0.961039
76	Weregenu	0.974026
77	WhaLmat	0.987013
78	Yerer Area	1

Table 4.3-2: Scaled crime location

After normalizing the data, the data has to be transformed to be executed in the right order. Sckit-learn provides pipeline class to help with transformations. Fit_transform()joins fit() and then transform() on the same data and is used for initial fitting of parameters.

We slice it into training set and test set by using train_test_split. We imported train_test_split from sub library model selection in order to split the data to train set and test set. It is important when doing time series to split train and test with respect to a certain date. So, we don't want the test data to come before training data. when splitting the data, we make sure that our test set is large enough to yield statistically meaningful results and it is a representative of the whole dataset. In splitting the data our goal is to create a model that generalizes well to new data. It's usually around 80/20 or 70/30 depending on the amount of data we have. We used 80/20 train test split; 80 for train and 20 for test respectively. This enables us to make a good evaluation of the model we used by measuring how well it simplifies on new data it has not yet seen. If the test set isn't well created, we won't be able to draw any meaningful decisions about our model. [50]

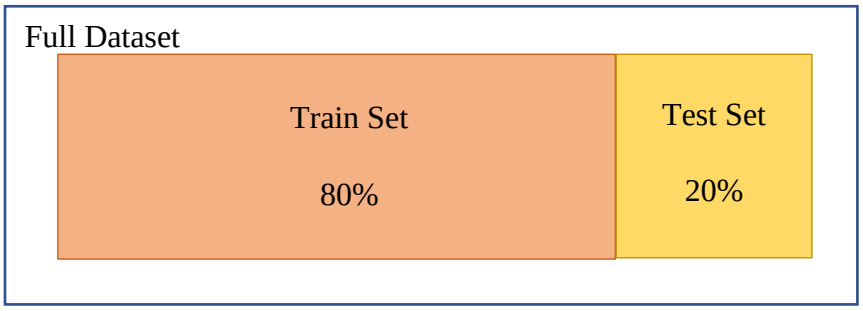


Figure 4.3-2:Train test split

4.4 Model Implementation Phase

After going through initialization and data pre-processing phase we will pass on implementation phase. In our case we will implement a time series analysis using LSTM to predict crime.

4.4.1 Forecast method

There are many types of LSTM networks that can be used for different types of problems. For this research, we used vanilla LSTM model. This LSTM model has a single hidden layer and an output layer which is used to predict. Input shape argument of first hidden

layer specifies the input shape for each sample. The current hidden state h(t) is generated from the previous hidden state h(t-1).

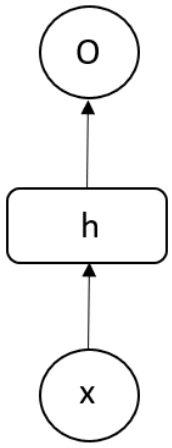


Figure 4.4-1:Vanilla LSTM

4.4.2 Dropout

Dropout regularization works by dropping out inputs to a layer, which could be input variables in the data or activations from the previous layer. This regularization method is supported by keras which is provided by dropout core layer. This will prevent overfitting by ignoring randomly selected neurons in training. 20% of dropout regularization is often used and gives a good result in preventing overfitting and retaining model accuracy. So, we go by 20%/ 0.2 dropout regularization.

4.4.3 Activation

converts an input signal of a node in RNN to an output signal. And the output signal will serve as an input in the next layer in the stack. We also need this activation function to perform backpropagation optimization strategy. While backpropagating in the network, it computes the loss with respect to weights. In this study we used sigmoid activation function because this activation function used as the getting function for the input gate, output gate and forget gate. In feed forward networks the vanishing gradient problem is dealt with by using different activation function. But, unlike the other feed forward networks the vanishing gradient problem is resolved by the network structure of the LSTM.

4.4.4 Loss

It's a method that evaluates how well the algorithm models the data. There are a lot of factors in choosing loss function. We used mean absolute error loss function for this study. Mean absolute error is a measure of difference between two continuous variables. It is the average of the absolute errors of the predicted and the actual value [51]. Mean absolute equation is given by:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n}$$

Where, x and y are variables of paired observations that expresses the same phenomenon;

$|e_i|$ which is equal to $|y_i - x_i|$ is an average of absolute errors and,

y_i is the prediction and x_i is the true value.

4.4.5 Optimizer

We used 'adam' optimizer which is the combination of two optimizers Adaptive gradient algorithm (ADAgrad) and Root Mean Squared Propagation (RMSprop). ADAgrad optimizer adaptively scaled the learning rate for each dimension. It improves the performance on problems with sparse gradients by maintaining a per-parameter learning rate. The learning rate is calculated depending on the past gradients that have been computed for each parameter.

$$\Theta_{t+1} = \Theta_t - \frac{n}{\sqrt{G_t + \epsilon}} * g_t$$

Where, Θ is the parameter to be updated

η is the initial learning rate

ϵ is some small value that is used to avoid division of zero

G_t is the sum of the squares of the past gradients and,

g_t is the gradient estimate in time step t.

The problem with this optimization is that the learning rate starts vanishing very quickly as the iterations increase. RMSprop considers fixing the vanishing rate by using a certain number of previous gradients. The calculation will be updated as:

$$\Theta_{t+1} = \Theta_t - \frac{n}{\sqrt{\mathrm{E}[g^2]_t + \epsilon}} * g_t$$

Where,

$$\mathrm{E}[g^2]_t = 0.9\mathrm{E}[g^2]_{t-1} + 0.1g_t^2$$

Adaptive moment estimation or adam considers exponentially decaying average of past gradients and exponentially decaying average of past squared gradients to compute the adaptive learning rates for each parameter. Adam optimizer can be represented as

$$\Theta_{t+1} = \Theta_t - \frac{n}{\sqrt{\acute{\upsilon}_t + \epsilon}} * m_t$$

ύ and ḿ are considered as the estimates of the first and the second moment of gradient respectively.

4.4.6 Metrics

We used regression metrics called R-square (R^2) which explains the amount of variance in a relationship between variables. it is called the coefficient of determination. [52]

$$R^2 = \frac{SSR}{SST} + \frac{SST - SSE}{SSE}$$

SST is the measure of total variation in Y variable which is defined as,

$$SST = \sum_{i=0}^{n} (Y_i - \tilde{Y})^2$$

Where $\tilde{Y}$ is the sample mean of Y variables that is,

$$\tilde{Y} = \frac{1}{n} + \sum_{i=1}^{n} Y$$

SSE is the sum squared errors which measures the amount of variability in Y an is given by

$$SSE = \sum_{i=0}^{n} (Y_i - \tilde{Y}_i)^2$$

Ideally, R^2 should be as close to 1 as possible, suggesting that the predicted values are very close to the actual values.

4.5 Model Training Phase

Training a machine learning algorithm includes a machine learning algorithm and training set to learn from. The training set have to contain the target attribute. In training process, we need to have training data, data attribute that contains the target variable and training parameters to control learning parameters. We used fit() function from the keras API to train the model. It contains X, Y, Epochs, Batch_size, test_data, and verbose

X: training data of list of NumPy arrays

Y: target data of list of NumPy arrays

Batch_size: defines the number of samples to work through before updating the internal model parameters. Batch is like a for-loop iterating over one or more samples and make predictions. The predictions are compared to the expected output variables and error is calculated at the end of the batch.

Epochs: the number of epochs determines the time that the learning algorithm will work through the entire training. It's an iteration over the entire X and Y data provided. The model is trained until the epoch of index `epochs` is reached

Verbose: setting verbosity mode expresses the way we see the training progress for each epoch. If verbose is 0 it means its silent or will show us nothing; if its 1 it will show progress bar if its 2 it shows one line per epoch it will only mention the number of epochs.

Test_data: it’s a data that is holdback from training our model and is used to give an estimate of model. It’s used to give the unbiased estimation of the model when comparing or selecting between final models. It evaluates loss and model metrics at the end of the epoch.

Chapter 5

RESULT AND DISCUSSION

In this chapter, with reference to the aim of the study which was to determine the crime mapping, the results of the study are presented and discussed. The aim to apply machine learning models on the preprocessed datasets and making predictions were covered in the previous chapter that presented the design and implementation of the model.

As discussed in section 4 we trained our model based on vanilla LSTM model. The following sub-sections explain the results of the model based on performance measures.

we predicted one-year data of crime type and location. To get clear pattern, we divide the time into month, day and hour. The graph below shows the monthly crime prediction.

5.1 Results on monthly distribution

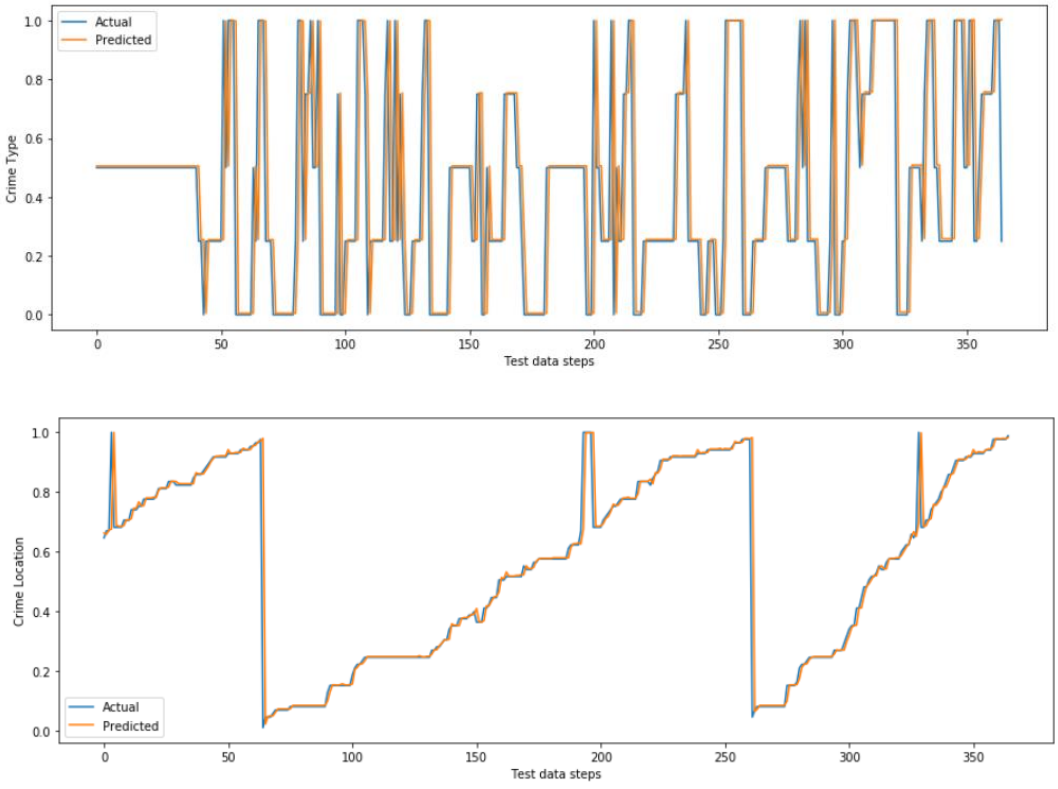


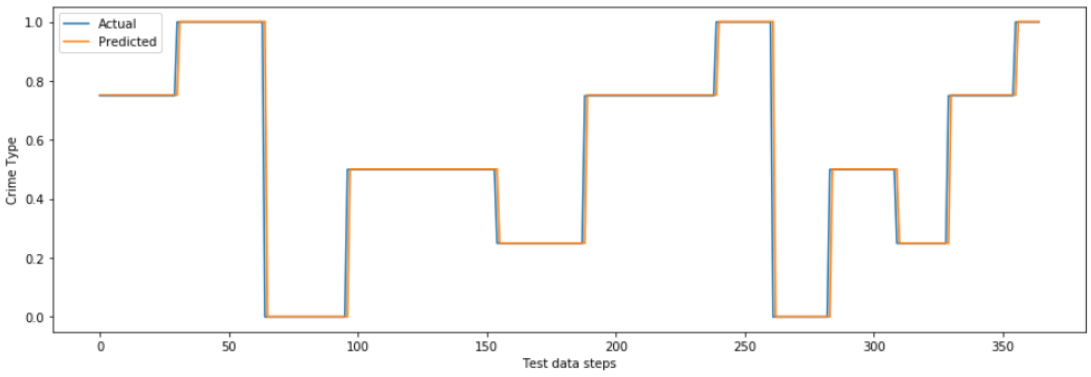
Figure 5.1-1:Visualization Data: a) Monthly crime type prediction, b) Monthly crime location prediction

In the above figure, the first graph shows crime type prediction of the actual and predicted graph. We get an R-square score of 0.879 and MAE of 0.020. As we can see from the graph, the crime type varies from as the time differs. As described in the prior chapter we have five crime types. There are five peaks in the graph that represents these five crime types. The highest peak represents theft crime type and the lowest peak represents assault crime type and the middle peaks are creating violence, cheque fraud and drug. Looking at this graph we can tell that there will be a high percentage of occurrence of cheque fraud crime type throughout the year. At the end of the year, theft crime type will happen more often than the others.

The second graph shows the neighborhood prediction of the actual and predicted graph we get R-square score of 0.930 and MAE of 0.017. There happens to be 81 crime location in our data and each location will vary based on specific time step.

5.2 Results on daily distribution

Data mining turns out to be a stepping stone for predictive policing. By using data and data mining technologies, predictive policing has the potential to provide the best evaluation of what will happen. Researches have shown that data mining can greatly improve crime analysis and aid in reducing and preventing crime.



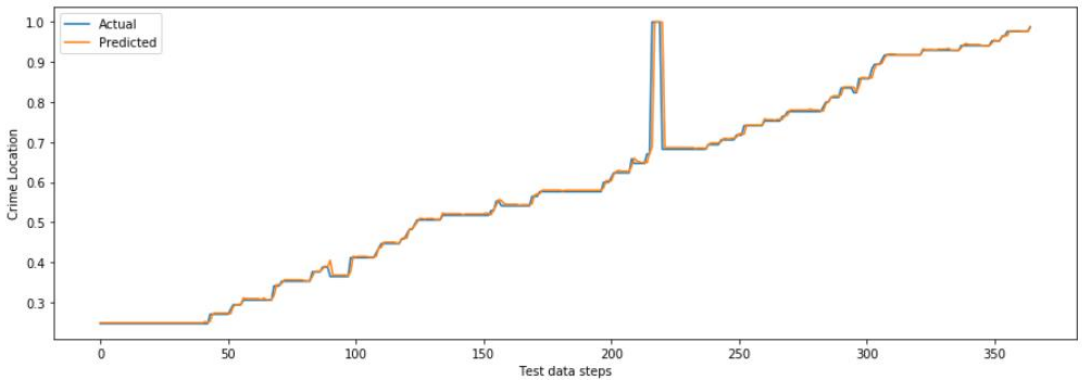


Figure 5.2-1: Visualization Data: a) Daily crime type prediction, b) Daily crime location prediction

The daily prediction shows the crime type and location predicted vs. the ground truth graph. In the first graph, which is the crime type prediction in a daily basis, the R-square is 0.925 and MAE is 0.005. Drug crime type happens more frequently and assault seems to happen rarely. Creating violence, cheque fraud and theft have average crime rate comparing to the others. The beginning and end of the month will be busy by handling theft and during crimes. They will occur in the middle of the month too. But other crimes rather than these two have the probability of occurrence

Figure 5.1-2: b) shows the daily crime location prediction. it shows the prediction quality of the test data. The R-square is 0.989 and the MAE is 0.015. This tells us that there's an enormous divergence between locations as the days diverge. Looking at both graphs, the daily prediction mingles with the month prediction graph. There will happen to be crime incidents in 17 health centers at the beginning of the month and; in the middle of the month Yerer Areas will have crime incidents.

There's also an hourly prediction to help us see the patterns of what kind of crimes could occur at which place. The figure below demonstrates this type and location prediction also comparing the predicted and true value.

5.3 Results on hourly distribution

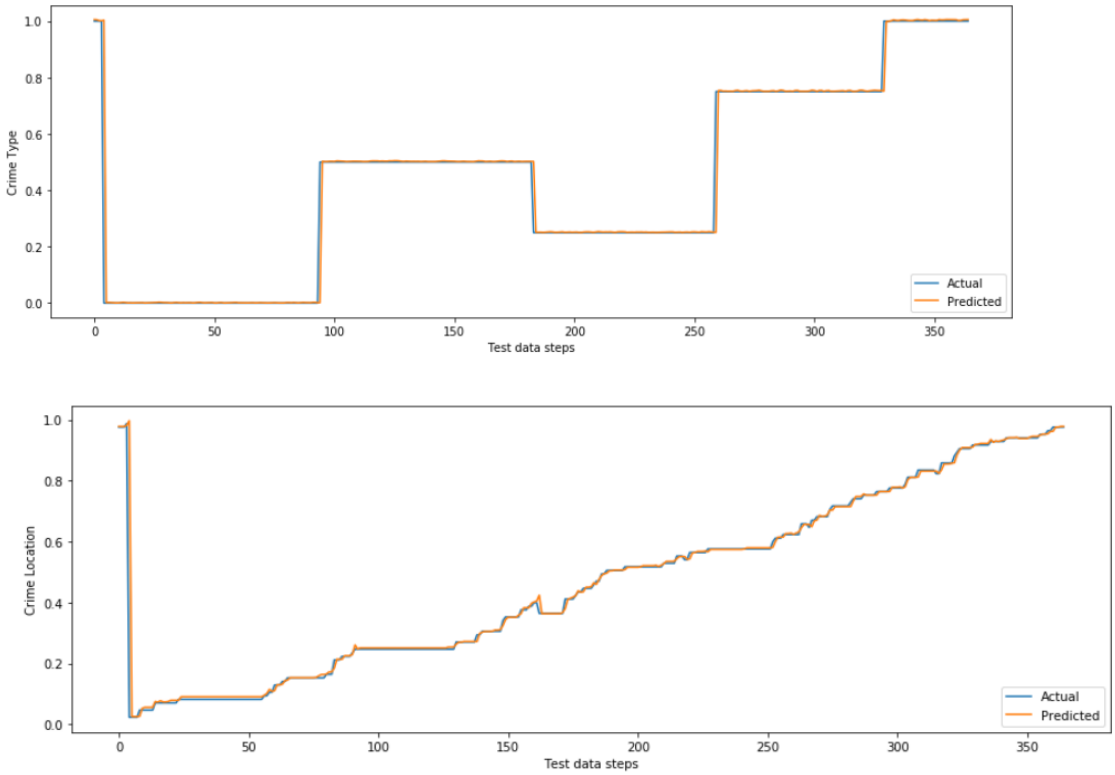


Figure 5.3-3: Visualization Data: a) Hourly crime type prediction, b) Hourly crime location prediction

We get an R-square score of 0.959 and MAE of 0.009 from figure 5.1-3 a). This hourly crime type prediction presents that there is almost a slight difference between crime types at each hour but as the result shown in 5.1-3 b), the location becomes dense as the time increases.

Theft will get higher rate in the night time. Drug are also high in night time. Crime like assault, check fraud, and violence will occur in day time.

Areas like Samit, SefereArea, SefereSelam, SelamAmba, ShalaMenafesha, Shola, Sunshine, Urael, Weja, WelloSefer andYererArea will get high crime incidents in night times and the rest will be exposed to crime at day time.

Combining these two graphs we will get a visualization that theft and drug will be higher in Sumit, Sefera Area, SefereSelam, SelamAmba, ShalaMenafesha, Sheger Homes, Shola,

Sunshine, Urael, Weja, WelloSefer, Weregendu, WhaLmat and Yerer Area. And other crime types will occurin the rest of the locations in day time.

There will be a high probability in Yerer area in the night time which is between the night time 23:00 and 00:00. Areas like Bras Hospital bulbula, chefe, chereqasefer, CMC Michael, Dldy, Downtown, Ednamall, and Egziaberab church are likely to have cheque fraud and creating violence crimes respectively. The accuracy score is 0.968 and MAE is 0.007 which is closer to the actual value. The table below puts the accuracy and error values in a summarized method. Combining the two graphswe can tell that there will be theft in the mid night.

	Monthly		**Daily**		**Hourly**	
	R-Square	**MAE**	**R-Square**	**MAE**	**R-Square**	**MAE**
Type	0.879	0.020	0.925	0.015	0.959	0.009
Location	0.930	0.017	0.989	0.005	0.968	0.007

Table 5.3-1: Summary table for error and accuracy score

Above table demonstrates the predictive capability of LSTM models on the incidence types and location. We can see that this model delivers better result especially on crime location; which is a good start for bole sub city police commission where there is no predictive policing strategy.

Chapter 6

CONCLUSION AND RECOMMANDATION

6.1 Conclusion

The foremost objective of this research is applying the most known neural network model for time series problems which is long short-term memory network and predict crime based on the time and location of committed crimes and also evaluate models by comparing predicted result with the actual value.

An exhaustive study is performed on the given data with one of the most known machine learning (neural network) approaches which is Long short-term memory (LSTM). The method for this research incorporates past criminal activity records in bole sub city that models them based on RNN LSTM to predict occurrence of crimes. The research approach consists five phases. Firstly, we collected the crime data from bole sub city police commission. Time series data is used to extract context information. We then undergo with the process of data pre-processing and generate more efficient data that can be used in predicting crime. Thirdly, LSTM RNN model is employed to accurately predict crime incidences. Then after, train dataset has undergone through data cleaning and data transformation process. Through experiments and analysis, we are able to determine that LSTM model gives high quality of prediction. R-square score is used to evaluate the model and the result on crime type prediction on monthly, daily and hourly basis is 0.879, 0.925 and 0.959 respectively. Also, the results on location prediction in monthly, daily and hourly manner are 0.930, 0.989 and 0.968 respectively. The LSTM model has high prediction accuracy and the low error rate, showing best generalization capability. The pick points where crime can happen in a specific period of time were defined and conferred with respect to crime locations.

6.2 Recommandation and future work

Using this method, law enforcement agencies can deploy resources more accurately and in timely manner. With respect to locations, these techniques can help identify patterns where the incidence takes place.

It is obvious that for more detailed analysis further research has to be done. It would be of interest to build models considering other factors on top of time and location such as weather, environmental factors and others. Future work may also involve applying different models in a large dataset and analysing their effects. Predicting crimes accurately helps to improve crime prevention and decision and advance the justice system.

BIBLIOGRAPHY

[1] S. Ford, "Benefits of Crime Prevention," Australian Crime Prevention Council.

[2] X. Li, "Application of data mining methods in the study of crime based on international data sources," Tampere University Press, 2014.

[3] H. O. Quarshie, "Journal of Emerging Trends in Computing and Information Sciences Using ICT to Fight Crime -A Case of Africa," January 2014.

[4] K. Gebremedhin, "Ethiopia 2016 crime rate said high: OSAC 2016 report," THE ETHIOPIA OBSERVATORY (TEO), 2016.

[5] A. G. TAMIRAT, "The Police and Human Rights in Ethiopia," Abyssinia Law, 2015.

[6] S.-S. Kim, "Theoretical Foundations of Data Mining and Predictive Analytics," sungsoo, 2014.

[7] J. Bachner, "Improving Performance Series Predictive Policing Preventing Crime with Data and Analytics," Johns Hopkins University, 2013.

[8] J. Rosen, "Data-driven policing helps predict when, where crimes will occur," Johns Hopkins University, 2013.

[9] R. P. S. a. R. F. Crowther, "Quantitative Methods for Optimizing the," *Journal of Criminal Law and Criminology,* vol. 57, no. 2.

[10] A. M. &. M. Wessels, "Predictive Policing: Review of Benefits and Drawbacks," *International Journal of Public Administration,* vol. 42, no. 12, 2019.

[11] "Oxford," in *Dictionary of Sociology*, Littlefield, Adams & Company., p. 136.

[12] "Wikipedia," 7 September 2019. Available: https://en.wikipedia.org/wiki/Forecasting.

[13] "Time Series : A Data Measured With The Passage Of Time," 2016.

[14] M. Rouse, "An enterprise guide to big data in cloud computing".

[15] A. G. Josh Patterson, "Deep Learning: A Practitioner's Approach," in *O'Reilly*, O'Reilly Media, Inc., 2017 , p. 2.

[16] M. Zhang, "Time Series: Autoregressive models," University of Pittsburgh, 2018.

[17] R. A. &. R. K. Agrawal, *An Introductory Study on Time Series Modeling and Forecasting.*

[18] S. C. Neal Stephenson, "Deep Learning," in *O'Reilly* , 2017.

[19] J. Le, "The 10 Deep Learning Methods AI Practitioners Need to Apply," 2017.

[20] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, Inc, Aurélien Géron.

[21] A. G. Josh Patterson, Deep Learning: A Practitioner's Approach, 2017.

[22] A. GOYAL, "Long Short Term Memory(LSTM) RNN," 2018.

[23] H. Jaeger, "Echo state network," Jacobs University Bremen, Bremen, Germany, 2007.

[24] N. A. Boukary, "A COMPARISON OF TIME SERIES FORECASTING," 2016.

[25] Sangarshanan, "Time series Forecasting — ARIMA models," Towards Data Science, 2018.

[26] D. K. Alexander Stec, "Forecasting Crime with Deep Learning," 2018.

[27] L. M. a. N. Meghanathan, "USING MACHINE LEARNING ALGORITHMS TO ANALYZE CRIME DATA," *Machine Learning and Applications: An International Journal (MLAIJ)*, vol. 2, no. 1, 2015.

[28] G. Tesemma, "The Nexus Between Urban Crime and Poverty: the Case of Addis Ababa City Administration," 2017.

[29] A. O. O. e. al., "A Decision Tree Algorithm Based System for Predicting Crime in the University," Nigeria, 2017.

[30] Y.-L. L. e. al., "Grid-Based Crime Prediction UsingGeographical Features," Taiwan, 2018.

[31] M. Mowafy, " Building Unstructured Crime Data Prediction Model (Practical Approach)," *INTERNATIONAL JOURNAL OF COMPUTER APPLICATION*, vol. 4, 2018.

[32] S. Sathyadevan, "Crime Analysis and Prediction Using Data Mining".

[33] A. Nadathur, "Crime analysis and prediction using big data," *International Journal of Pure and Applied* , vol. 119, 2018.

[34] M. V. Pednekar, "Crime Rate Prediction using KNN," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 6, no. 1, 2018.

[35] P. Y. a. V. Gudur, "Predictive Modelling of Crime data set using Data Mining Techniques," *IJDKP*, vol. 7, no. 4, 2017.

[36] T. T. N. e. al., "Building a Learning Machine Classifier with Inadequate Data for Crime Prediction," *Journal of Advances in Information Technology*, vol. 8, no. 2, 2017.

[37] H.-B. K. Hyeon-Woo Kang, "Prediction of crime occurrence from multi-modal data using deep learning," 2017.

[38] S. K. e. al., "Crime Analysis Through Machine Learning," 2018.

[39] M. K. &. S. Kajita, "Crime Prediction by Data-Driven Green’s Function method".

[40] S. S. e. al., "A Study on Classification Learning Algorithms to Predict Crime Status," *International Journal of Digital Content Technology and its Applications,* vol. 7, no. 9, pp. 361-369, 2013.

[41] J. A. U. T. a. K. J. P. Espinosa, "Crime Modelling and Prediction Using Neural Networks," *Theory and Practice of Computation,* pp. 218-228, 2017.

[42] E. Culurciello, "The fall of RNN/LSTM," 2018.

[43] E. Kang, "Long Short-Term Memory(LSTM)," 2017.

[44] P. 7. I. R. W. Y. S. U. Python, "Mindfire Solutions," 2017.

[45] Available: https://pandas.pydata.org/.

[46] Available: https://github.com/tensorflow/tensorflow.

[47] J. Brownlee, "Introduction to Python Deep Learning with Keras," 2016.

[48] Y. Upadhyay, "Deep Learning Frameworks: Tensorflow, Theano, Keras and popularly used libraries," jan, 8.

[49] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, United States of America: O'Reilly Media Gravenstein Highway North, Sebastopol, CA 95472. , 2017.

[50] N. Buduma, Fundamentals of Deep Learning, United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017.

[51] Available: https://runawayhorse001.github.io/LearningApacheSpark/stats.html.

[52] B. k. V. a. A. Orr, "R-Square".

[53] "Vinod Sharma's Blog,". Available: https://vinodsblog.com/2019/01/07/deep-learning-introduction-to-recurrent-neural-networks/. [Accessed 7 Jan 2019].

[54] "nVIDIA," Developer. Available: https://developer.nvidia.com/deep-learning.

[55] M. K. &. D. HANBAY, "Effective Classification of Phishing Web Pages," *Anatolian Journal of Computer Sciences,* vol. 2, no. 1, pp. 15-36, 2017.

[56] Q. Gao, "Stock market forecasting using recurrent neural network," 2016.

[57] J. Martinsson, "Automatic blood glucose prediction with confidence using recurrent neural networks," 2018.

[58] F. e. a. Okubo, A neural network approach for students' performance prediction, 2017.

[59] N. M. &. S. L. Waslander, Multistep Prediction of Dynamic Systems With Recurrent Neural Networks, IEEE.

[60] Z. Qiao, Pairwise-Ranking based Collaborative Recurrent Neural Networks for Clinical Event Prediction, 2018.

APPENDIX

```
from math import sqrt
import numpy as np
from numpy import concatenate
from matplotlib import pyplot
from pandas import read_csv
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
# convert series to supervised learning
defseries_to_supervised(data, n_in=1, n_out=1, dropnan=True):
        n_vars = 1 if type(data) is list else data.shape[1]
        df = DataFrame(data)
```

```
        cols, names = list(), list()
        # input sequence (t-n, ... t-1)
        for i in range(n_in, 0, -1):
                cols.append(df.shift(i))
                names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
        # forecast sequence (t, t+1, ... t+n)
        for i in range(0, n_out):
                cols.append(df.shift(-i))
                if i == 0:
                        names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
                else:
                        names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
        # put it all together
        agg = concat(cols, axis=1)
        agg.columns = names
        # drop rows with NaN values
        if dropnan:
                agg.dropna(inplace=True)
        return agg
# load dataset
dataset = read_csv('CDatasetD.csv', header=0, index_col=0)
print(dataset.isnull())
values = dataset.values
```

```
# integer encode direction
encoder = LabelEncoder()
values[:,0] = encoder.fit_transform(values[:,0])
values[:,1] = encoder.fit_transform(values[:,1])
values =values.astype('float32')
#normalize
sc =MinMaxScaler(feature_range=(0,1))
scaled = sc.fit_transform(values)
print(scaled)
reframed = series_to_supervised(scaled, 1, 1)
reframed.drop(reframed.columns[[0]], axis=1, inplace =True)
print(reframed.head())
values = reframed.values
x = values
y = values[:,1]
trainX, testX, trainY, testY = train_test_split(x, y, test_size=0.2)
print (trainX.shape, trainY.shape)
print (testX.shape, testY.shape)
trainX = trainX.reshape((trainX.shape[0], 1, trainX.shape[1]))
testX = testX.reshape((testX.shape[0], 1, testX.shape[1]))
print(trainX.shape, trainY.shape, testX.shape, testY.shape)
# design network
```

```
model = Sequential()
model.add(LSTM(100, input_shape=(trainX.shape[1], trainX.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(trainX, trainY, epochs=50, batch_size=72, validation_data=(testX, testY), verbose=2, shuffle=False)
# plot history
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
yhat = model.predict(testX)
test_y = testY.reshape((len(testY), 1))
pyplot.plot( testY, label='Actual', color='blue')
pyplot.plot( yhat, label='Predicted', color='red')
pyplot.xlabel('month')
pyplot.ylabel('Woreda')
pyplot.legend()
pyplot.show()
r2_score(testY, yhat)
mean_absolute_error(testY, yhat)
```