SCHOOL OF GRADUATE STUDIES

TIME SERIES CRIME PREDICTION ANALYSIS USING RNN: A CASE OF WOLKITE CITY POLICE DEPARTMENT

MSc. THESIS

SOLOMON KASSAYE ESHETU

JANUARY, 2024

WOLKITE, ETHIOPIA

Wolkite University

School of Graduate Studies

Time Series Crime Prediction Analysis Using RNN: A Case of Wolkite City Police Department

A Thesis Submitted to School of Graduate Studies, in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science and Engineering (Specialization: Computer Science)

Solomon Kassaye Eshetu

Major Advisor: Mesfin Abebe (Ph.D.)

Co-Advisor: Berihun Getnet (M.Sc.)

January, 2024

Wolkite, Ethiopia

APPROVAL SHEET

School of Graduate Studies

Wolkite University

Time Series Crime Prediction Analysis Using RNN: A Case of Wolkite City Police Department

Submitted by:		
Solomon Kassaye		1-1-2024
Name of Student	Signature	Date

Approved by:

1. Dr. MESFIN ABEBE		01/01/2024
Major Advisors Name	Signature	Date
2. Berihun Getnet		01/01/2024
Co- Advisors Name	Signature	Date
3.		02/05/2016
Department Head	Signature	Date
4.		
Name of Chairman, DGC	Signature	Date
5.		02/05/2016
Name of Dean, SGS	Signature	Date

WOLKITE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby certify that we have read and evaluated Thesis titled "**Time Series Crime Prediction Analysis Using RNN: A Case of Wolkite City Police Department**" Prepared under our guidance by Solomon Kassaye Eshetu. We recommend that the Thesis shall be submitted as fulfilling the requirements for the award of a MSc. degree in computer science and engineering.

Dr. MESFIN ABEBE ______ 01/01/2024

Major Advisor signature date

Berihun Getnet ______ 01/01/2024

Co-Advisor signature date

As members of the Board of Examiner of the Master of Science Thesis open defense examination, we have read and evaluated this Thesis prepared by **Solomon Kassaye Eshetu** and examined the candidate. We hereby certify that; the thesis is accepted for fulfilling the requirements for the award of the degree of Master of Science (M.Sc.) in computer science and engineering.

1. Kindie Biredagn (Ph.D.) ______ Dec-31-2023
   Name of external examiner signature date
2. Worku Muluye ______ 01/01/2024
   Name of internal examiner signature date
3. Kuma Yadi ______ 01/01/2024
   Name of chairman signature date

Final approval and acceptance of the Thesis is contingent upon the submission of its final copy to the Council of Postgraduate Program (CPGS) through the candidate's department or school graduate committee (DGC or SGC).

DECLARATION

By my signature below, I declare and affirm that this Thesis is my own work. I have followed all ethical principles of scholarship in the preparation, data collection, data analysis and completion of this thesis. All scholarly matter that is included in the thesis has been given recognition through citation. I affirm that I have cited and referenced all sources used in this document. Every serious effort has been made to Avoid any plagiarism in the preparation of this thesis.

This thesis is submitted in partial fulfillment of the requirement for a degree from the School of Graduate Studies at Wolkite University. The thesis is deposited in the Wolkite University Library and is made available to borrowers under the rules of the library. I solemnly declare that this thesis has been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Brief quotations from this Thesis may be used without special permission provide that accurate and complete acknowledgement of the source is made. Request for permission for extended quotations from, or reproduction of, this thesis in whole or in part may be granted by the Head of the School or Department or the Dean of the School of Graduate Studies when in his or her judgement the proposed use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the thesis.

Name: Solomon Kassaye Eshetu Signature: ______________

Date: 1-1-2024

School/Department: Computer Science and Engineering

ACKNOWLEDGEMENT

To begin with, I want to express my thanks to God, but I'm at a loss for words to convey the depth of my gratitude. Following that and most importantly, I wish to convey my appreciation to Dr. Mesfin Abebe, my thesis advisor, for helping me to do this research from start to finish. Next, thanks my Co-advisor Mr. Berihun Getnet, who helps me by giving valuable comments. And my depth gratitude to Dr. Erkihun Massresha, who gives me general comments and support. Also, I would like to thank to Crime investigator Vice Abdil Jelil Kedir and Crime investigator Vice Solomon Nmani who helps me to prepare Crime dataset. My profound thanks go out to my family, who have provided unwavering support, offered prayers, and shown incredible patience and encouragement during the course of accomplishing this task of research work. Additionally, I owe a debt of gratitude to my friends who assisted me in this thesis by sharing valuable information and insights.

Name: Solomon Kassaye Eshetu

Signature: ____________

Date: 1-1-2024

DEDICATION

I desire to the Ethiopians who were wronged and lost their lives in an unfair way, as well as to the law enforcement agencies that are trying to stop Crime.

LIST OF ACRONYMS AND ABBREVIATIONS

Adam	Adaptive Moment Estimation
AM	Ante Meridiem
ANN	Artificial Neural Network
ARIMA	Auto Regressive Integrated Moving Average
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CSV	Comma Separated Value
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
EC	Ethiopian Calendar
FFNN	Feed Forward Neural Network
GCN	Graph Convolution Network
GGConv	Graph with Gated Convolution
GHz	Giga Hertz
GRU	Gated Recurrent Unit
HALR	Historical Average Logistic Regression
HMM	Hidden Markov Model
KNN	K-Nearest Nighbor
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

MATLAB	Matrix Laboratory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NbConv	Nighbor Convolution
NLP	Natural Language Processing
NNM	Neural Network Model
PM	Post Meridiem
R^2	Coefficient Determination
RAM	Random Access Memory
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SARIMA	Seasonal Auto Regressive Integrated Moving Average
SEBCPM	Stack Based Crime Prediction Method
SEMMA	Sample, Explore, Modify, Model, and Assess
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
US	United States
XGBoost	Extreme Gradient Boost

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF FIGURES IN THE APPENDIX

ABSTRACT

Crime is an undesirable phenomenon and a global concern that impacts both society and individuals. Annually, we observe an increase in the number of criminal incidents, posing a threat to both public safety and the well-being of the community. Demanding facilities at unequal times is one problem observed in police workforce assignment. Our study aims to determine and examine the relationship between Crime date-time and the number of Crime incidents, as well as their types and locations. We collected nine thousand eight hundred twenty (9,820) criminal offenses handled by Wolkite City ranging from 2008-to-2014 E.C and we include seven (7) most frequently occur Crime types and fifty-two (52) Crime locations in our study. Different preprocessing techniques are applied such as label encoder and Minmax scaler. And we employed RNN models, including Long Short-Term Memory, Gated Recurrent Unit, Bidirectional Long Short-Term Memory and Bidirectional Gated Recurrent Unit, also train the models using training dataset and predict Crime type and location, finally evaluate the model's using metrics like MSE, R^2 and others by testing dataset. For Crime type prediction LSTM has MSE of 0.0125, 0.0126 and 0.0468, Bi-LSTM has MSE of 0.0126, 0.0125 and 0.0466, GRU has MSE of 0.0127, 0.0128 and 0.0501, Bi-GRU has MSE of 0.0126, 0.026, and 0.0468, for hourly, daily and monthly respectively for each model. For Crime location prediction LSTM has MSE of 0.0108, 0.0109 and 0.0617, Bi- LSTM has MSE of 0.0108, 0.0110 and 0.0506, GRU has MSE of 0.0106, 0.0105 and 0.0582, Bi-GRU has MSE of 0.0105, 0.0106, and 0.0513, for hourly, daily and monthly respectively for each model. For Crime type prediction Bi-GRU, Bi-LSTM, LSTM, GRU perform R^2 of 0.9995, 0.9994, 0.9899, and 0.9811 respectively. Fo Crime location prediction Bi-LSTM, LSTM, Bi-Bi-GRU and GRU gained R^2 of 0.9938 0.9937, 0.9937 and 0.9934, respectively. For hourly Crime type prediction LSTM is slightly better and for daily and monthly Bi-LSTM is better. For hourly and monthly Crime location Bi-GRU is slightly better and for daily, GRU slightly better. In terms of R^2, Bi-GRU slightly higher score than others for Crime type and for Crime location Bi-LSTM is slightly higher R^2 values. In general, Bi-LSTM and Bi-GRU gained better score for Crime prediction with low error for our dataset.

Keywords: *RNN, LSTM, Bidirectional LSTM, GRU, Bidirectional GRU, Time Series Crime Prediction.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Crime is global concern impacts for society and individuals and it has negative effect on society[1]. Criminality is undesirable occurrence, which happens globally, in both developed and developing states. Criminal activities have the potential to significantly impact the economy and compromise the quality of life and welfare of individuals, consequently giving rise to social and communal issues[2]. Criminal offenses and unlawful actions can incur expenses for both the society as a whole and specific sector[3]. The safety of the community is a significant consideration when individuals relocate to new areas to protect their environment. In reality, different kinds of offenses can result in various consequences and ramifications. Generally, crimes emerge from a variety of factors, including specific triggers, human conduct and tendencies, challenging situations, and economic hardship. Furthermore, multiple elements like joblessness, gender disparities, high population density, and lack of education can contribute to a rise in violent crimes[4]. The growth of densely populated urban areas is closely connected to the rising Crime rates in various environments, including commercial buildings and public housing areas. A socially sustainable community places significant importance on Crime reduction to ensure that everyone can lead peaceful and safe lives. In the absence of peace, communities plagued by corruption cannot prosper both economically and socially. Therefore, the analysis of Crime reports and statistics is essential for improving the safety and protection of humanity while sustaining long-term growth[5].

Ethiopia is a developing country and is being attacked by various crimes. Especially now, the spread of Crime increases the risk that people will not live in peace. Crime is an act that has a lot of impact, it prevents people from having a healthy life, it prevents them from returning to work in peace, and their human rights are revoked. And Crime causes many other harms to human beings.

The concept of predicting crimes has become increasingly popular in recent years as it enhances the ability of law enforcement agencies to address Crime through computational methods. It is crucial to develop more advanced predictive algorithms that can guide police patrols to focus on potential criminals[6]. Various research endeavors have been undertaken to forecast types of crimes and their rates by utilizing Crime datasets or corpora. For instance, studies have been conducted in regions such as South Korea and the United States to achieve this goal (including Portland) [7],[8]. Recent instances of effective Crime prediction involve the application of Machine Learning (ML) and Deep Learning (DL) algorithms. Notable machine learning algorithms for Crime prediction encompass Naïve Bayes, Random Forest, Support Vector Machine (SVM), Decision Trees, and regression techniques[9].

Precise Crime prediction is a challenging yet vital endeavor in the fight against unlawful activities. Effectively estimating Crime rates, types, and locations based on historical patterns offers numerous computational prospects. Machine learning has emerged as the contemporary standard for predictive analysis in Crime prevention, although only a limited number of studies have systematically compared various machine learning techniques. Machine learning algorithms, particularly those designed for non-linear data processing, have demonstrated their efficacy in several domains, including Crime prediction. These algorithms are proficient at handling high-dimensional data swiftly and extracting significant data features[10]. Considering that Crime data typically follows a time series format, revealing patterns that correspond to specific time periods and indicating the evolution of criminal activities over the years. Consequently, it becomes necessary to employ time series analysis in order to visualize these patterns effectively. Furthermore, utilizing a deep learning algorithm, especially Long Short-Term Memory (LSTM), proves to be advantageous for classifying crimes over time by incorporating comprehensive measures[11]. Additionally, recent research strongly recommends the use of the Auto Regressive Integrated Moving Average (ARIMA) model for predicting Crime trends[12]. Hence, the objective of this study is to conduct a Time Series Crime Prediction Analysis employing Recurrent Neural Network (RNN), which includes LSTM, Bi-LSTM, GRU, and Bi-GRU models, along with a visual summary facilitated by exploratory data analysis. Crime dataset collected from Wolkite City Police Department ranging from 2008 to 2014 totally seven-year Crime offence. Data collection done manually because of the data is un available on the internet and again written to computer using excel application. Wolkite City police department, still no new tools and techniques working to simplify processing of records. In the dataset collection process, we include the date, time, Crime type, and Crime location.

1.2. Motivation of the Study

As machine learning and deep learning relies primarily on historical data to predict future outcomes[13],[14]. Crime records from the past are necessary for analyzing and forecasting criminal incidents[15]. These records must be in structured and standard form in order to be used by learning algorithms. In our country Ethiopia including Wolkite City police office have no, structured day to day-based Crime data is available in computerized way. In the city different criminal types occur such as: Assault, Murder, Robbery, Beat, Violation and Fraud. In this study we include such Crime types and with respective locations to develop Crime type and location prediction model. We motivated to improving community safety. Crime remains a persistent concern within numerous communities, presenting a significant challenge to the overall safety of the public. The development of robust Crime prediction models holds the promise of aiding law enforcement agencies in proactively distributing resources, thereby enhancing their ability to prevent criminal activities and respond with greater effectiveness to incidents and rational resource allocation which is the law enforcement agencies operate under resource constraints, underscoring the importance of efficient allocation, Crime prediction models play a pivotal role in optimizing the allocation of resources by guiding the deployment of patrols and investigations toward regions where higher predicted Crime rates are anticipated. Also embracing technological progress. The advent of technological advancements, notably in the realm of machine learning, with a specific focus on Recurrent Neural Networks (RNNs), presents an unprecedented opportunity to conduct more profound and efficient analyses of extensive historical Crime data.

1.3. Statement of the Problem

Crime represents a prominent issue in contemporary society. It is an undesirable phenomenon that transcends borders and is prevalent in both advanced and less-developed nations[4]. Notably, in developing countries, there is a consistent upward trend in the incidence of crimes. Also, in our current city, we notice that various crimes are committed in different neighborhoods. Crime exerts significant impacts on victims, society, and the social fabric. Various research has demonstrated that various factors, including the time of day, weather conditions, environmental attributes, and previous Crime occurrences, have an influence on future criminal activity[16]. Crimes occur haphazardly in Wolkite, caused many damages on people's life and their property for instance, on February 15, six people lost their lives and more than fifteen people were seriously injured in Wolkite city[17]. Furthermore, the allocation of police resources presents another challenge, which pertains to the overall amount of police time needed for various tasks. An issue in police workforce assignment is the demand for services at different times, which can be uneven. Various machine learning techniques, such as decision trees, support vector machines, and random forests, are available. For time-series data, models like Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and Seasonal ARIMA are based on a linear combination of past observations and past errors. They assume that the time series data is stationary, meaning that the statistical properties of the data do not change over time, but these models have not long dependencies[18]. Existing related work papers, such as [19], [20],[21]and [22], have not specifically investigated the application of RNN-based models for time series data with daily, hourly, and monthly granularities. Moreover, while [4] and [23] implemented LSTM for time series data, they have focused on crime prediction and have not compared the results with other RNN-based models such as GRU, Bi-LSTM, and Bi-GRU. This comparison is crucial in determining the most suitable RNN-based model for accurate and reliable time series forecasting in crime prediction tasks.

To address these challenges, this study integrates a collection of related Crime events from both the present and the past, with a focus on recent crimes to develop a predictive model. Deep learning techniques, widely applied in various domains, including Crime prediction, are employed. Specifically, this study utilizes recurrent neural network-based models like Long Short-Term Memory Networks, Bidirectional Long Short-Term Memory Networks, Gated Recurrent Units, and Bidirectional Gated Recurrent Units for analyzing Crime datasets in Wolkite City. These models have long dependencies for time series Crime predictions.

To find a solution to the above challenges, this study investigates and answer the following research questions.

RQ1: For the task of predicting time series Crime type and Crime location, which recurrent neural network is the most appropriate choice?

RQ2: What type of crime will occur in which areas every day in the future?

RQ3: Which area is more prone to crime in the future?

1.4. Objective of the Study

1.4.1. General Objective

The primary aim of this study is to develop Time Series Crime Prediction model using Recurrent Neural Network in the Case of Wolkite City Police Department.

1.4.2. Specific Objectives

In pursuit of the overarching goal of this study, the following specific objectives are pursued throughout the entire research process.

- To collect dataset for Crime prediction models
- To pre-process the collected Crime dataset and to make it suitable for the model building
- To investigate exploratory data analysis
- To build the Crime prediction models utilizing Bi-LSTM, LSTM, GRU, and Bi-GRU
- To evaluate the performance of Bi-LSTM, LSTM, GRU, and Bi-GRU Crime prediction models using MSE, MAE and R_square
- To identify the high crime rates on each day of the week and where they occurred
- To identify areas with high crime rates per day and month

1.5. Significance of the Study

This research can provide valuable insights into the prevention of recurrent Crimes in a specific area. It does so by identifying patterns of past criminal activities and recognizing the most prevalent Crime types in that area. It can also give information to various departments instead of just one department for best resources in more communities around the city. It also allows department to better allocate their resources to protect areas with heavy Crime for best care of these extents by police officers. This can aid in the strategic allocation of police resources to the area's most likely to experience criminal activity at any given time.

1.6. Scope and Limitation of the Study

The scope of this study is to utilize a recurrent neural network (RNN) for time series crime prediction analysis, aiming to develop a model that can accurately forecast crime rates based on historical data. This study will involve data collection, preprocessing, model building (such as LSTM, Bi-LSTM, GRU and Bi-GRU), model training, evaluation, hyperparameter tuning, prediction, and visualization, with the ultimate goal of providing insights and tools that can aid in proactive crime prevention and resource allocation. This study is focused on Wolkite Police Department because we thought that it would be convenient to collect information easily and also because we have seen various crimes being noticed every now and then. We considered only the location of Crime, Crime time, Crime date and Crime type.

1.7. Organization of the Study

The subsequent chapters of our research are structured as stated below: Chapter two offers an extensive literature review and related work, encompassing various concepts and models pertinent to Crime Prediction Analysis. In the third chapter, we delve into the methodologies employed for Crime Prediction Analysis. Chapter four introduces the Crime prediction models proposed in this study. Chapter five is dedicated to the practical implementation of these proposed Crime prediction models. Chapter six, we conduct an in-depth examination of the results and engage in comprehensive discussions. Here, we scrutinize the outcomes of experiments and the various test cases. Ultimately, the concluding chapter encapsulates the summarizing our work and paving the way for future research endeavors.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

2.1. Introduction

Within this chapter, we delve into the notion of crime and the associated theoretical frameworks for Crime prediction. This encompasses an introduction to Crime itself, as well as an exploration of time series forecasting, statistical forecasting models, neural network forecasting models, and hybrid models. The chapter culminates with a comprehensive examination of previous research and related works pertaining to Crime prediction.

2.2. Definition of Crime

A Crime is an unlawful act that someone can be lawfully punished for example damaging someone's goods. Crime more generally to define something that's wrong, unfair[24]. Crime constitutes one of the predominant issues in contemporary society. It is un wanted phenomenon that transpires on a global scale, affecting both advanced and less developed nations. Especially in developing countries, the number of Crimes is increasing from time to time[20].

Numbeo is a platform that provides numerical data, and it reports the Crime Index for Ethiopia as 49.3 (49.3%) [25]. The lists indicate the Crime rate has seen a significant increase over the past three years, reaching 68.79%, indicating a substantial rise.

2.3. Time Series Prediction

Prediction involves the act of anticipating future events by considering past and current occurrences. In essence, it serves as a decision-making tool for businesses to manage and mitigate the impact of future uncertainties, particularly in the context of Crime prevention, through the analysis of historical data[26]. Historical time series data or information undergo processing using forecasting techniques to make predictions about future outcomes based on this data[16]. A time series comprises a set of observations of well-defined data elements collected through periodic measurements over a duration[27]. A time series is characterized as a collection of vectors denoted as x(t), where t represents the progression of time (t = 0, 1, 2...). The variable x(t) is treated as a generic variable, and the measurements recorded at different time points in a time series are arranged in a systematic sequential manner. When dealing with time-dependent data, time series forecasting is employed as a predictive method to anticipate future events based on a chronological sequence[28]. This approach offers a wide range of practical applications, including weather forecasting, earthquake prediction, Crime prediction, pattern recognition, and signal processing. Time series forecasting is accomplished using various technologies such as machine learning, Gaussian processes, fuzzy logic, and hidden Markov models.

Machine learning a software application that allows predicting outcomes without being programmed[29]. Various organizations harness machine learning algorithms to enhance their decision-making processes, identify diseases, boost productivity, predict weather conditions, and much more. At its core, machine learning is about converting data into valuable insights. It operates at the intersection of statistics, computer science, engineering, and other relevant fields, making it applicable to a wide range of domains, from politics to geosciences. Machine learning provides solutions for numerous problems by employing algorithms that derive structural patterns from data examples. In this process, a computer learns to understand the underlying structures that represent information within raw data. These structural models capture insights extracted from the original data, enabling us to predict unknown data. These models can take various forms, each with its own methodology for applying rules to uncover data patterns and make predictions.

2.4. Statistical Prediction Models

2.4.1. Autoregressive-Moving-Average (ARMA) Model

ARMA model is a forecasting technique that predicts upcoming values using past values. Forecasting is a serious task for many business objectives, for instance budgeting, product planning, predictive maintenance, predictive analytics, etc. The main advantage of Autoregressive Moving Average models is that relatively simple. It only requires a small data to make a prediction, and correct for short forecasts, and it work on dataset without a trend also it takes the AR(p) and MA(q) features[30]. An Autoregressive Moving Average calculated as:

$$y_t = c + e_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} \qquad (2.1)$$

In the given equation, yt represents the series, and e_t is a series of random errors with unknown characteristics, assumed to conform to a standard probability distribution. A more complex iteration of the auto-regressive moving average (ARMA) model is referred to as (VARMA), which is formally defined as:

$$y_t = v + A_1 y_{t-1} + \ldots + A_p y_{t-p} + u_t - M_1 u_{t-1} + \ldots + M_q u_{t-q} \quad (2.2)$$

In this equation, yt represents an output variable, while $yt{-}p$ and $ut{-}q$ represent the preceding output and input exogenous variables, respectively. Ap and Mq are matrices that contain the model's parameter values.

2.4.2. Autoregressive-Integrated-Moving Average (ARIMA) Model

The ARIMA model forecasts future values in a time series by considering the past prediction errors (MA terms) and past values (AR terms). This enables the model to adjust for abrupt changes in the time series. Therefore, the Autoregressive Integrated Moving Average (ARIMA) forecasting for a stationary time series is essentially a linear regression equation[31]. In this equation, the predictors are the lagged values of the dependent variable and the lagged values of the prediction errors[32]. The equation for the ARIMA with parameter (p, d, q) where, p signifies the autoregressive order, d signify the degree of differencing and q signify moving average order[33],[34]. The process can be expressed as follows:

$$y_t = c + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \ldots + \theta_q e_{t-q} + e_t \quad (2.3)$$

Where, the parameters φ1 . . . φp are associated with the autoregressive model, and they quantify the influence of past values of the time series on the current value, the parameters θ1. . . θq are associated with the moving-average model, and they capture the influence of previous error terms on the current value. $y_{t-1} \ldots y_{t-p}$ represent the lagged values of the time series, meaning they are past values that are used to predict the current value.e_t represents the white noise, which represents the random or unpredictable component of the series.et represent white noise, yt represents the differenced series of the original time series, where the difference is taken at degree d.

2.4.3. Seasonal Autoregressive-Integrated-Moving Average (SARIMA) Model

As implied by its name, this model is employed when the time series displays a seasonal pattern. It shares similarities with ARIMA models, but to accommodate the seasonality, we need to introduce a few additional parameters. SARIMA can be seen as an extension of ARIMA, denoted as ARIMA (p, d, q) (P, D, Q) with the inclusion of a seasonality parameter m[35].

$$\text{SARIMA as ARIMA } (p, d, q)\,(P, D, Q)\, m \tag{2.4}$$

Where:

- P: indicates the quantity of autoregressive terms.
- d: signifies the extent of differencing.
- q: represents the count of moving average terms.
- m: denotes the number of time periods within each season.
- (P, D, Q): conveys the (p, d, q) parameters applied to the seasonal aspect of the time series.

2.5. Deep Learning Technique

Deep learning is a recent and prominent area of exploration within the machine learning (ML) domain[36]. It excels at uncovering the intricate structures within data and has the capability to effectively characterize data using advanced modeling techniques. Since 2006, deep structured learning, often referred to as deep learning, has emerged as a novel domain within the field of machine learning (ML). In recent years, the methods developed through deep learning analysis have significantly influenced a diverse array of signal processing and information handling tasks, spanning both traditional and emerging domains. These impacts extend to fundamental aspects of machine learning (Machine Learning) and artificial intelligence (AI)[37].

A deep learning (DL) model will learn important features automatically by means of less engineered feature, in its place of requiring the data expert to manually excerpt the important features. Deep learning is popular by two mains[38]. The first reason, it was created that convolutional Neural Networks (CNNs) quickly run on graphical processing unit, for example

NVidia's Tesla K80 processor and the second reason, data experts realized that the vast stockpiles of data benign collected can serve as a huge training corpus and so supercharge the convolutional Neural Networks into yielding a substantial upgrading Deep learning has made substantial advancements in enhancing the precision of natural language processing (NLP) and computer vision (CV) algorithms. Deep learning incorporates a hierarchical structure of concepts or features, in which lower-level concepts can be derived from higher-level ones, and conversely, higher-level concepts can be defined based on the lower-level ones.

2.6. Neural Network Forecasting Technique

Neural networks have a history of over 50 years. They are a type of machine learning method designed as an abstraction of the mathematical model of the biological nervous system. These networks find extensive application in tasks like time series forecasting[39]. Deep neural networks (DNNs) are structured to acquire features through multiple interconnected layers. Each layer exclusively receives input connections from the previous layer and then transmits its processed connections to the subsequent hidden layer [40]. The fundamental element of a neural network is the node, which draws its inspiration from the biological neurons found in the human brain. These networks between nodes are constructed to mimic the way neural connections develop in biological brains over time.

2.7. Recurrent Neural Networks

Scientists have introduced Recurrent Neural Networks (RNN) as an alternative neural network architecture to address classification and text mining tasks[41]. It allocates numerous weights to past data points in a sequence of processing, making it a powerful technique for tasks involving strings, text, and sequential data classification. Recurrent neural networks belong to a class of networks designed for making predictions about the future. They can analyze time series data, such as stock prices, to suggest optimal times for buying and selling. Additionally, they can forecast car routes and contribute to accident prevention in self-driving systems. RNNs are frequently employed in various natural language processing tasks, including speech recognition, speech synthesis, and sentiment analysis. Their strength lies in their ability to capture substantial time-related information across multiple input vectors. RNNs feature a feedback loop that aids in learning from sequential data, even when dealing with sequences of varying lengths. The networks within RNNs incorporate an additional parameter matrix for capturing the time-based relationships within the data. They are trained based on current and previous inputs from all previous time steps to generate sequential outputs at each time step.

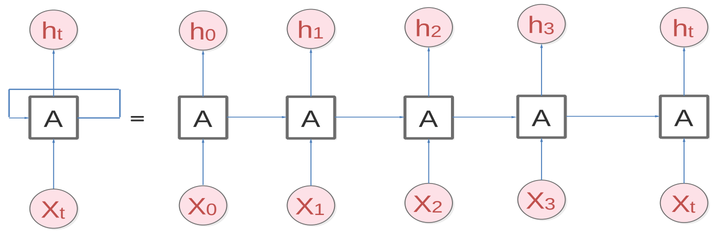


Figure 2.1. Recurrent Neural Network

As depicted in Figure 2.1, the RNN cell receives two inputs: the outputs from the previous hidden state and the observation at time t within the hidden state.

2.7.1. LSTM Network

Long Short-Term Memory (LSTM) is a distinctive variant of Recurrent Neural Network (RNN) that addresses issues by preserving long term dependency more knowledgeably in contrast to the basic recurrent neural network[42]. It is mainly useful for solving the vanishing gradient problem. Similar to RNN, it has a chain-like structure. LSTM model has good sequential predictive power[43]. LSTM (Long Short-Term Memory) Network cells are a unique category of Recurrent Neural Networks designed to meticulously control the flow of information into each node state. Their specialty lies in their ability to manage and handle long-term dependencies [44]. LSTM has three gates and one cell state, this is used to regulate the network and store relevant information for long period.

- ***The Forget Gate:*** is plays a pivotal role in determining which information from the previous state should be retained and what should be discarded. It operates by considering both the previous cell state (C(t-1)) and the current input (X(t)), and it generates a forget gate output (f(t)) that can vary between 0 (indicating complete forgetting) and 1 (indicating complete retention).
- ***The Input Gate:*** is responsible for determining the new information to be stored in the cell state. It comprises two components: a sigmoid layer that generates an input gate (i(t)) and a tanh layer that generates a candidate cell state (C(t)). The input gate plays a key role in deciding which values from the candidate cell state will be incorporated into the cell state.

- ***The Output Gate:*** is responsible for deciding the composition of the upcoming hidden state (h(t)). It utilizes the current cell state (C(t)) to generate an output gate (o(t)), which, when applied to the cell state through multiplication, yields the hidden state. The mathematical formulation of LSTM can be expressed as follows[45].

$$\text{Forget Gate } (f(t)) = Wf * [h(t-1), x(t)] \quad (2.5)$$

$$\text{Input Gate } (i(t)) = Wi * [h(t-1), x(t)] \quad (2.6)$$

$$\text{Candidate Cell State } (C\sim(t)) = Wc * [h(t-1), x(t)] \quad (2.7)$$

$$\text{New Cell State } (C(t)) = f(t) * C(t-1) + i(t) * C\sim(t) \quad (2.8)$$

$$\text{Output Gate } (o(t)) = Wo * [h(t-1), x(t)] \quad (2.9)$$

$$\text{New Hidden State } (h(t)) = o(t) * C(t) \quad (2.10)$$

2.7.2. Bi-LSTM Network

Schuster and Paliwal are the creators of Bi-LSTM, which stands for Bidirectional Long Short-Term Memory[46]. The network is trained using input sequences from both the past and the future. Input data is passed through a series of connected layers. [47]. Bi-LSTM has the capability to forecast each element within a sequence by leveraging the context from both the past and the future of that sequence. It has two directions forward and backward directions. The rightward function of a bidirectional LSTM with L input units and H hidden units can be computed using the following equations.

$$\text{New Hidden State } (h\_f(t)) = \text{LSTM}(Wf * [h\_f(t-1), x(t)]) \quad (2.11)$$

$$\text{New Hidden State } (h\_b(t)) = \text{LSTM}(Wb * [h\_b(t+1), x(t)]) \quad (2.12)$$

The ultimate representation at time step t is typically derived by merging or joining the hidden states obtained from both the forward (h_f(t)) and backward (h_b(t)) LSTMs, often through concatenation.

Bi-LSTMs are known for their ability to capture contextual information from both directions, which can be essential in various NLP tasks. For example, in part-of-speech tagging, knowing the words before and after a given word can help determine its grammatical role. In the hidden layer of Bi-LSTM, it stores values obtained from both the forward and backward computations, namely h_f(t) for forward calculation and h_b(t) for reverse calculation. The ultimate output, Y, relies on both h_f(t) and h_b(t) [48].

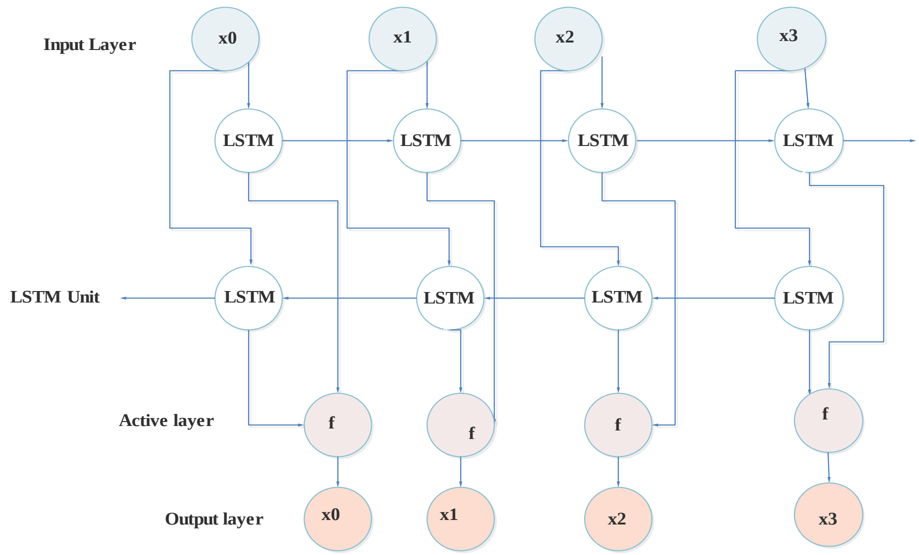


Figure 2.2. Structure of Bi-directional LSTM

2.7.3. Gated Recurrent Unit (GRU)

A Gated Recurrent Unit (GRU) is a component of a particular type of recurrent neural network. It is designed to harness connections between a series of nodes to execute machine learning tasks related to memory and grouping data[49]. Its purpose is to tackle certain drawbacks found in conventional RNNs, such as the Vanishing Gradient problem, which can pose challenges for RNNs in capturing extended patterns or dependencies within sequences[50]. GRU is important for range utility that calls for chronological or temporal data[51]. The mathematical expression for GRU can be described as follows[45]:

$$\text{Update Gate } (z) = Wz * [h(t-1), x(t)] \tag{2.13}$$

$$\text{Reset Gate } (r) = Wr * [h(t-1), x(t)] \tag{2.14}$$

$$\text{Candidate Hidden State } (h\sim) = W * [r * h(t-1), x(t)] \tag{2.15}$$

$$\text{New Hidden State } (h(t)) = (1 - z) * h(t-1) + z * h\sim \tag{2.16}$$

GRUs are similar in function to LSTM networks, but they have a somewhat simpler architecture with fewer gates. GRUs have found extensive application in a range of sequence-to-sequence assignments, such as machine translation, speech recognition, and text generation. This is mainly attributed to their capacity to effectively capture extensive patterns and relationships in data sequences and mitigate vanishing gradient problems.

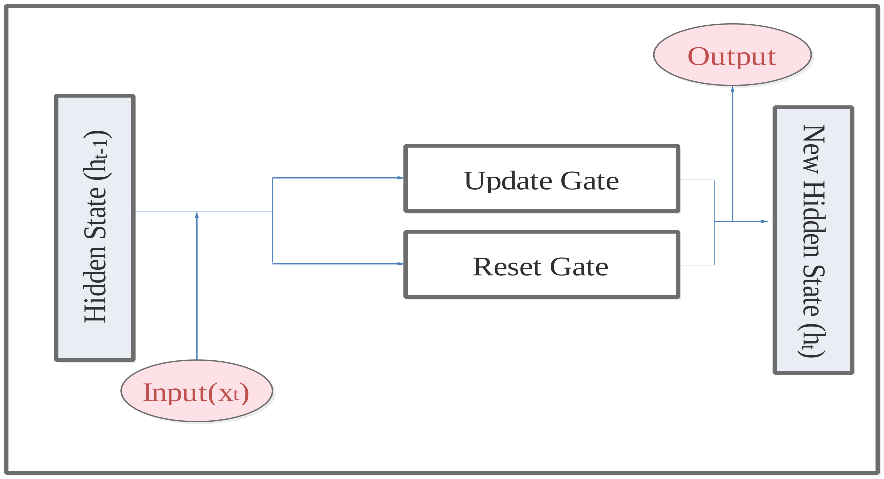


Figure 2.3. Structure of GRU

2.7.4. Bidirectional Gated Recurrent Unit (Bi-GRU)

A Bidirectional Gated Recurrent Unit (Bi-GRU) is a modified version of the typical Gated Recurrent Unit (GRU) that boosts the model's capability to encompass context information from both preceding and forthcoming time steps within a sequence. Similar to the standard GRU, it is employed in a variety of natural language processing (NLP) and sequence modeling undertakings. The mathematical expression for Bi-GRU can be articulated as follows:

$$\text{Update Gate } (z\_f) = Wz\_f * [h\_f(t-1), x(t)] \tag{2.17}$$

$$\text{Reset Gate } (r\_f) = Wr\_f * [h\_f(t-1), x(t)] \tag{2.18}$$

$$\text{Candidate Hidden State } (h\sim\_f) = W * [r\_f * h\_f(t-1), x(t)] \tag{2.19}$$

$$\text{New Hidden State } (h\_f(t)) = (1 - z\_f) * h\_f(t-1) + z\_f * h\sim\_f \tag{2.20}$$

$$\text{Update Gate } (z\_b) = Wz\_b * [h\_b(t+1), x(t)] \tag{2.21}$$

$$\text{Reset Gate } (r\_b) = Wr\_b * [h\_b(t+1), x(t)] \quad (2.22)$$

$$\text{Candidate Hidden State } (h\sim\_b) = W * [r\_b * h\_b(t+1), x(t)] \quad (2.23)$$

$$\text{New Hidden State } (h\_b(t)) = (1 - z\_b) * h\_b(t+1) + z\_b * h\sim\_b \quad (2.24)$$

The ultimate representation at time step t is typically generated by joining or merging the hidden states obtained from both the forward (h_f(t)) and backward (h_b(t)) GRUs, frequently through concatenation [52],[53].

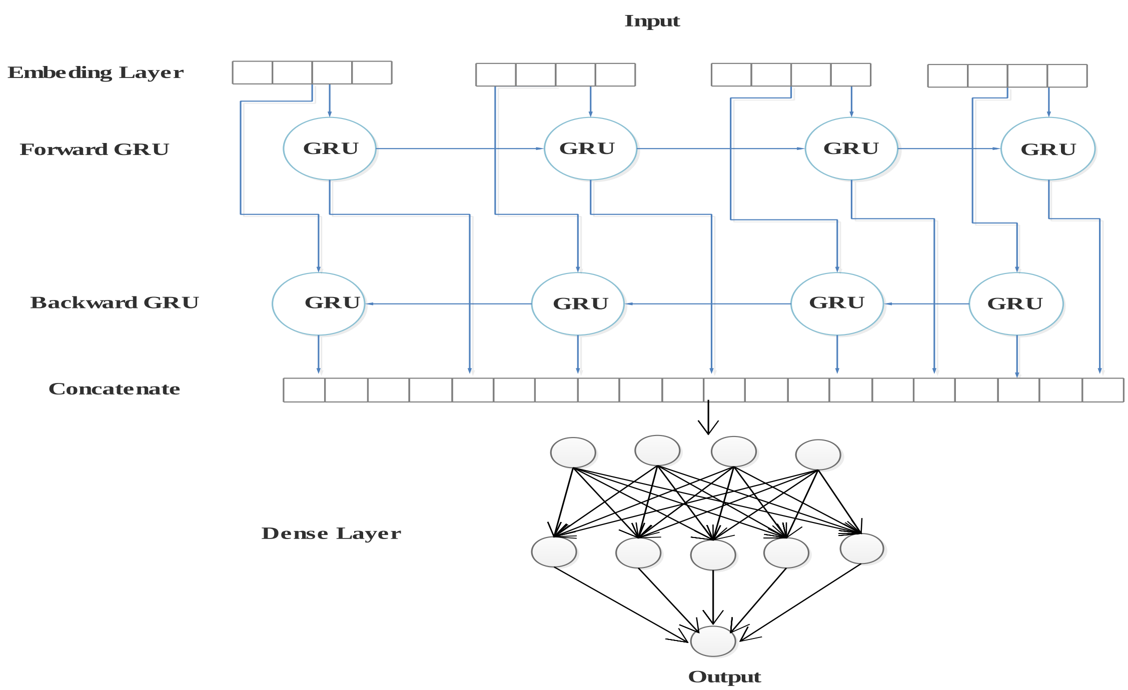


Figure 2.4. Structure of Bi-directional GRU

2.8. Convolutional-Neural Network

A Convolutional Neural Network (CNN) is a sophisticated deep learning algorithm capable of accepting input and allocating appropriate weights to different features or elements within those input[54]. A CNN is comprised of several layers, including an input layer, multiple hidden layers (akin to a multilayer neural network), and an output layer. These layers also include pooling layers and optimization layers[55] . CNN is a blend of both unsupervised and supervised learning methods, known for its multiple hidden layers, which are commonly employed in image classification. It is often referred to as a deep learning neural network[55].

2.9. Feed-Forward Neural Network

This network serves as the foundation for deep learning models, including recurrent neural networks (RNN) and convolutional neural networks (CNN), and is often referred to as a multilayer perceptron (MLP). It is widely applicable in supervised machine learning. The network is structured with layers, with the first layer receiving input and the last layer providing output. The intermediary layers are hidden layers, having no direct connection with external inputs. This architecture is significant for numerous commercial applications, such as stock market prediction. The calculation for a Feed Forward Neural Network (FFNN) is as follows.

$$y_l = p_l \left(w_l p_{l-1} \left(w_{l-1} x_{l-1} + b_{l-1} \right) + b_t \right) \tag{2.25}$$

Where:

- xl−1: represents the input signal from layer (l − 1)
- bl−1: represents an arbitrary offset vector or bias term associated with layer (l − 1)
- wl: represents the weights connecting each neuron in layer (l − 1) to neurons in layer l
- pl−1: represents a vector of activation functions applied to the weighted sum of inputs from layer (l − 1)
- bt: represents the bias term associated with layer l
- pl: represents a vector of activation functions applied to the resulting sum of weighted inputs and biases

2.10. Echo State Network

Echo state networks the principles of supervised learning and architecture for recurrent neural network. The fundamental idea involves feeding the input signal into a large, randomly structured recurrent neural network. This process induces a nonlinear response signal in each neuron within the network, often referred to as the reservoir. The final output signal is then created by combining these response signals linearly.

Echo-state networks can be designed with various configurations, including the option to have or not have trainable input-output connections, output reservoir feedback, different types of neurons, distinct patterns of internal reservoir connectivity, and more. Furthermore, the computation of output weights can be accomplished through a range of offline or online linear regression algorithms. These include margin-maximization criteria, inspired by training Support Vector Machines (SVM), which are employed to control the output weights, along with least Mean Square Error (MSE) solutions[56].

2.11. Hybrid Models

Hybrid models have gained significant importance recently because they offer us effective tools for forecasting and prediction. These models are essential for understanding both the linear and non-linear aspects of a given dataset. Time series forecasting is an incredibly important application of machine learning and plays a pivotal role in today's business landscape. A time series consists of data points collected at regular time intervals over a period. Time Series Analysis is employed to identify the primary components in time series data, enabling the prediction of future data based on past behavior. Combining various models and algorithms in a hybrid approach yields more substantial and efficient results when handling forecasting tasks[57]. Various forecasting methods are available, including Simple Exponential Smoothing Models, Hidden Markov Models (HMM), and Artificial Neural Networks (ANN), each with a critical role in predicting data points. Among the different techniques, two widely employed methods are Auto-Regressive Integrated Moving Average (ARIMA) and Recurrent Neural Networks (RNN). These models are commonly used in everyday tasks and applications because they are proficient at handling both linear and non-linear data. Accurately forecasting a time series is a substantial and challenging endeavor, and ANN and ARIMA stand out as two highly popular and effective forecasting models. In hybrid forecasting techniques, both linear and non-linear algorithms are typically employed, and their outcomes are combined to enhance forecasting precision. The utilization of hybrid models has become a prevalent practice in recent years to achieve improved accuracy in forecasting. The preference for hybrid models stems from three main factors. First, it can be challenging to discern whether a time series is generated by a linear process. Second, real-time series data often exhibit a mixture of linear and non-linear patterns rather than being purely linear. Lastly, there is no one-size-fits-all method that can be applied to all forecasting scenarios[58].

2.12. Related Work

In this related work necessary literatures related to Crime prediction have reviewed to clearly state and furtherly explore the research problem. In this day the number of Crimes increase time to time so Crime prediction analysis is important task for identifying the occurrences of Crimes with time. Different studies done in different year to predict the Crime specially for other countries. In 2019 Designing Time Series Crime Prediction Model using Long Short-Term Memory Recurrent Neural Network presented by Meskel et al, they used Long Short-Term Memory Network with label encoder and min max scaler preprocessing techniques and MAE performance metrics to evaluate the model performance and gained MAE of 0.007,0.015,0.017 in hour, day and month respectively for location prediction and for Crime type prediction 0.009,0.005,0.02 by hour, day, month respectively[16]. In 2020 applied machine learning in social sciences: neural networks and Crime prediction presented by Forradellas et al, they used python Programming SEMMA, K-means Clustering and develop NNM as well as mean absolute error evaluation metrics and the gained MAE of 0.40 and MSE of 1.4609 [19] .In 2021 Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques done by Safat et al, they used Logistic Regression support Vector Machine, Naïve Bayes, K-nearest Neighbors, Auto Regressive Integrated Moving Average, Decision Tree, Multi-Layer Perceptron, Random Forest, Extreme Gradient Boosting (XGBoost), LSTM and KNN and XGBoost performance is achieve better accuracy 94% and 88% for chigago and loss angle dataset. And for time series LSTM achieves 12.66 RMSE and 11.70 MAE [4]. In 2021 An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach is done by Kshatri et al, Ensemble-stacking based method SVM bagging, SVM-Random Forest, SVM-stacking J48, and Naïve Bias, stack-based models achieve best accuracy 99.5% [20]. In 2022 Enhancing short-term Crime prediction with human mobility flows and deep learning architectures is done by Jiahui Wu et al, they used deep learning models like: Historical average logistic regression, Gated recurrent Unit, Attention Crime prediction and Graph Convolution Network and the gained F1 score using historical Crime +Mobility HALR, GRU, Attn, GCN, GGConv and NbConv has 0.403, 0.405,0.408,0,363,0.393 and 0.407, respectively [21].In 2022 Crime Prediction using Machine Learning with a Novel Crime Dataset is done by Shohan et al, they used five supervised machine learning classification algorithms such as Random Forest, XGBoost, Decision Tree, Ada Boost, Extra Tree. Random forest gain 50% F1-score over other algorithms [22].

Table 2.1. Related Work

Research Papers	***Authors and year***	***Methodology and Result Description***	***Research Gap***
Crime Prediction using Machine Learning with a Novel Crime Dataset[22]	Shohan et al. (2022)	-They used Random Forest, XGBoost, Decision Tree, Ada Boost, Extra Tree. Random forest gain 50% F1-score over other algorithms. They used 6574 crime datasets.	Researchers use traditional machine learning technique; these techniques have no dependencies. And also, not consider time series prediction.
Enhancing short-term Crime prediction with human mobility flows and deep learning architectures[21]	Jiahui Wu et at. (2022)	-They used HALR from machine learning and Deep learning like, GRU, Attn, GCN, GGConv, NbConv and using Average (standard deviation) of monthly F1 score using historical Crime +Mobility HALR, GRU, Attn, GCN, GGConv and NbConv has 0.403, 0.405,0.408,0,363,0.393 and 0.407, respectively	The researchers used HALR, and other deep learning architecture, but they didn't consider time series data means that it's normal prediction also not consider granularities like hourly, daily and monthly prediction.

Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques[4]	Safat et al, (2021)	Algorithms they used: Logistic Regression, SVM, Naïve Bayes, KNN, ARIMA, Decision Tree, MLP, Random Forest, XGBoost, and LSTM for time series, from all algorithms they used XGBoost performance is achieve better accuracy 94% and 88% for chigago and loss angle dataset. And for time series LSTM achieves 12.66 RMSE and 11.70 MAE. The used 7002821 instances for chigago and 2646463 crime instances for loss angle.	They used many machine learning algorithms to compare the accuracy which is the best but only used LSTM for time series forecasting and measured the performance by MAE and RMSE, for time series Crime forecasting other models should have been compared with Bidirectional LSTM, GRU RNN based models for better comparison and to forecast the best model for time series Crime.
An Empirical-Analysis of Machine Learning Algorithms for Crime-Prediction Using Stacked Generalization: An Ensemble Approach[20]	Kshatri et al, (2021)	-They used Ensemble-stacking based method: SVM bagging, SVM-Random Forest, SVM-stacking J48, SBCPM, SMO Naïve Bayes. -They implement learning-based methods using MATLAB.	The researchers not consider time series prediction only done normal prediction with machine learning techniques.

		Stack based models achieves best accuracy 99.5% accuracy. The used 60,000 Crime cases.	
Applied Machine Learning in Social Sciences: Neural Networks and Crime Prediction[19]	Forradellas et al, (2020)	-They used python Programming, SEMMA, NNM model and measure the performance using mean absolute error and mean square error, so using Neural network MLP gained 0.4095 MAE and 1.4609 MSE. They used 123,681 datasets.	The researcher not group the data in to granularities for hourly, daily and monthly prediction.
Designing Time Series Crime Prediction Model using Long Short-Term Memory Recurrent Neural Network [16]	Meskel et al, (2019)	-They used Long Short-Term Memory Network and R_square performance metrics and they gained 0.959 R_square value. They used 6033 datasets.	They did not compare other RNN based models such as Bi- LSTM, GRU and Bi-GRU to say LSTM is Best for time series Crime prediction.

In the above Table 2.1, we summarize the related papers in our study. In the proposed study we build RNN based Models including LSTM, Bi-LSTM, GRU and Bi-GRU for time series Crime prediction analysis instead of traditional machine learning technique RNN models beneficial for sequential information including Crime prediction. The existing papers like Shoham et al (2022) used traditional machine learning technique; these techniques have no dependencies. And also, not consider time series prediction, other paper Forradellas et al (2020), they didn't group the data in to granularities for hourly, daily and monthly prediction totally they didn't predict hourly, monthly and daily prediction, other paper Kslatri et al (2021), not consider time series prediction only done normal prediction with machine learning techniques. Jiahui Wu et al (2022), used HALR, and other deep learning architecture, but they didn't consider time series data means that it's normal prediction also not consider granularities like hourly, daily and monthly prediction and another papers Safat et al, (2021) and Meskel et al, (2019), they implement LSTM for time series data but to decide LSTM is best for time series Crime prediction, they didn't compare the results with other RNN based models like GRU, Bi-LSTM and Bi-GRU. For our study we employed RNN based models for time series Crime and Location prediction and we compared the result and forecast which of them suitable for time series prediction.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Introduction

This study focuses on developing and evaluating a time series crime prediction model using Recurrent Neural Networks (RNNs), specifically LSTM, GRU, Bi-LSTM, and Bi-GRU. The study adopts an experimental approach, involving data collection, preprocessing, and the design of controlled experiments, parameter tuning, and evaluates the performance of the RNN models using appropriate metrics. And also, implementation tools that we used included.

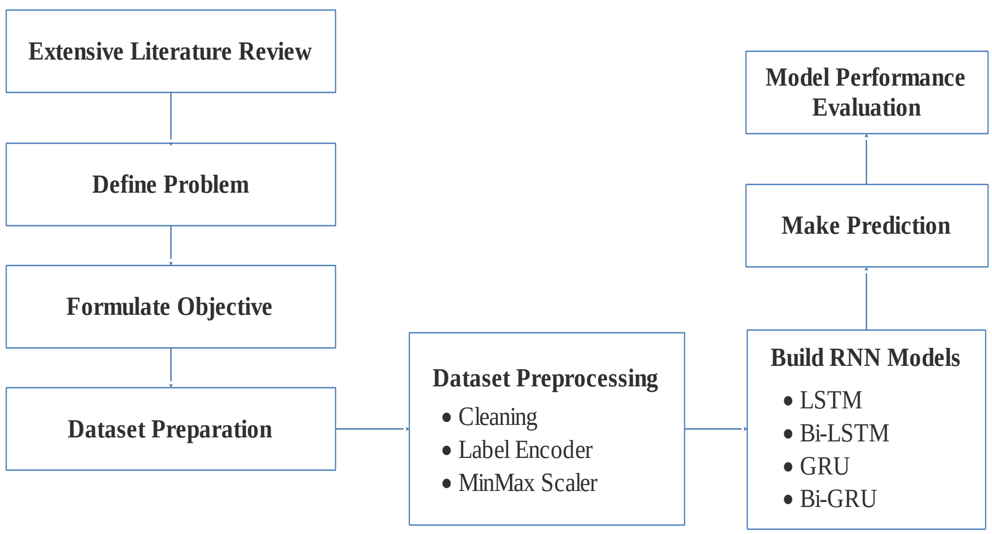


Figure 3.1. Research Flow

As shown in Figure 3.1 above, the research process flow is illustrated. It contains eight phases; the first sub-section of our research is an extensive literature review; we reviewed different research papers related to Crime prediction. In the second subsection, we have described the problems we found based on the literature review. The third sub-section is setting the research objective, so we put the general and specific objectives of the study here. In sub-section four, we collected all relevant datasets from Wolkite City police department to prepare the Crime prediction dataset, in sub-section five: we preprocessed the collected dataset to train the model easily, in sub-section six, we built RNN Based models such as LSTM, BiLSTM, GRU and Bi-GRU, in sub-section seven made Crime prediction using built models, in sub-section eight, we evaluated our Models using MSE, RMSE, MAE and R_square. From sub-section four to sub-section eight described in the following detailly.

3.2. Study Area Description and Crime Dataset

Wolkite is a town and separate woreda in southwestern Ethiopia. The administrative center of the central Ethiopia regional state. The town has a latitude and longitude of 8°17′N 37°47′E and an elevation between 1910 and 1935 meters above sea level[59]. Gurage is a zone in the Central Ethiopia Regional State of Ethiopia. The region is home to the Gurage people. Gurage is bordered on the southeast by Hadiya and Yem Zone, on the northwest by Kebena Special Woreda, north and east by the Oromia Region, and on the southeast by Silt'e. Its highest point is Mount Gurage. Wolkite is the administrative centre of the Region; Butajira is the largest city in this zone and the former administrative center[60]. Although the focus of this research is Wolkite City Police Department, in order to find more criminal records, Abeshge District Police also used the criminal record of 26 Kebeles. We used the most frequently committed Crimes, such as assault, robbery, fraud, creating violation, murder, drugs, and beat, which were investigated and recorded over the past seven years from 2008 to 2014 E.C. Each case file contains several characteristics like the date and time the Crime occurred, the nature of the Crime, and the specific location where the Crime took place.

3.2.1. Data Preparation

The research relied on data comprising Crime records from both Wolkite City and the Abeshge district. A large number of data collected from manually recorded books of the city's criminal records. When an individual is prosecuted for a Crime, the Crime committed is checked and recorded by the police. We also conducted this research by taking this filtered and recorded data. Upon registering the charge, the accused provides information regarding the nature of the offense, the location and time it occurred, as well as the date of the Crime. when the Wolkite Police Department recorded the information, it was in the register book, the work was tiring as it required re-entering it into the computer. First, we collected manually recorded criminal data and then entered the data to the computer using the Microsoft excel application. This task took us a very long time to prepare a lot of Crime data, it took us a total of seven months from November to May. To do this, the cooperation of the Criminal Police has allowed us to collect a large amount of Crime data.

Figure 3.2. Data Preparation

3.2.2. Data Statistics

The Crime dataset that we used contain seven most repeat Crime type like assault, creating violation, drug, fraud, robbery, beat and murder ranging from 2008 E.C to 2014 E.C. The data statistics we collected is shown below in the Table 3.1.

Table 3.1. Total Number of Data Collected in seven years

No	*Year*	*Number of data collected*
1	2008 E.C	1018
2	2009 E.C	1086
3	2010 E.C	1020
4	2011 E.C	1500
5	2012 E.C	1458
6	2013 E.C	1542
7	2014 E.C	2196
	Total	9,820

3.3. Dataset Pre-Processing

In the dataset preprocessing sub-section, we remove unnecessary violations. In the context of training a Crime prediction model, the application of machine learning techniques like clean, encode and scaling plays a crucial role in identifying and filling in missing values within specified attributes. Data cleaning is instrumental in both the training of the model and the overall performance of the process. Our data pre-processing phases illustrated in the following Figure 3.3.

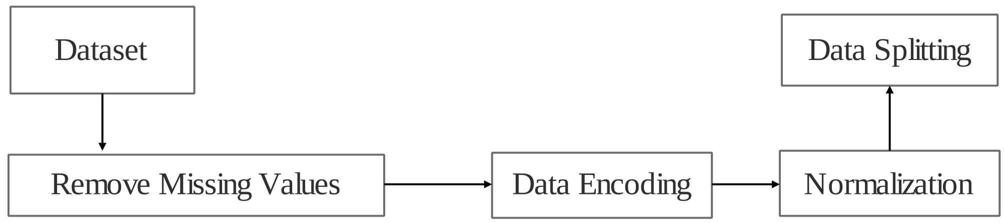


Figure 3.3. Data Preprocessing Phases

As shown in the Figure 3.3, we depicted the data pre-processing phases and in the following sub section we described each phase.

3.3.1. Replace Missing Values

Missing value is one problem for correct prediction so we need to fill missing with interpolation rather than eliminate the rows because of rows may contain necessary information. First, we checked the dataset is contain messing values using isnull () function. In our dataset there is 5 rows missing, so using linear interpolation replaced the missing rows using interpolate () method.

3.3.2. Categorical Data Encoding

Categorical data encoding involves the transformation of categorical values into numerical formats, which facilitates the training of machine learning or deep learning models, as these models primarily operate with numerical data. So, we have encoded our categorical values to number using Label encoder. assigns a unique integer value to each unique category. It is more suitable when the categorical variable has a hierarchy. so, our dataset contains ordered crime type and crime location with time manner in this case label encoder is preferable[61].

3.3.3. Normalization

Normalization helps each feature to be equally vital we need to normalize our dataset with the same. We employed the Minmax normalization technique, a well-known method for standardizing data. This technique entails transforming the minimum value to 0 and the maximum value to 1.

3.3.4. Data Splitting

Prior to inputting the data into the training and testing phases, it is essential to divide the dataset into two parts: training data and testing data. This division is accomplished by selecting a specified percentage of the dataset for training and the remaining portion for testing. In this case, we allocated 80% of the data for training and use the remaining portion for testing, because our dataset small to use 70% for training and 30% for testing and also there are other cross validation technique is there like k-fold, this is suitable for large dataset. Actually 80 by 20 train test splitting is better for model generalization and reduce over fitting.

3.4. Recurrent Neural Network Models

During this stage, we outline the architecture and the approach to learning used in our models. Specifically, we opt for RNN-based models when it comes to Crime prediction because our dataset is time series. For a sequence of information, RNN models is selected, and the proposed models are described in detail in the next chapter. Within our proposed models, we have integrated RNN-based models such as LSTM, Bi-LSTM, GRU, and Bi-GRU.

3.4.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an artificial neural network employed in the realms of artificial intelligence and deep learning. In contrast to conventional feedforward neural networks, LSTM incorporates feedback connections. LSTM finds utility in a wide range of applications, including unsegmented, connected handwriting recognition, speech recognition, machine translation, robot control, video games, healthcare, and the analysis of time series data.

Algorithm 1: Train and evaluate Long Short-Term Memory Model

Start

Input: Normalized sequenced training data

Output: Prediction of Crime type and Crime location

Initialization: epoch, batch size, lstm_unit, dropout rate

Step1: add LSTM Layer with defined lstm_units and input data

add dropout regularization

add dense layer with defined output dimension and tanh activation function

Step2: Compile LSTM model with MSE loss function and Adam optimizer

Step3: Fit the training data to LSTM Model with initialized hyper parameter

While maximum iteration not reached ***do***

Train LSTM Model

Step4: Evaluate LSTM model with testing data

Step5: Predict unseen data by LSTM Model

Step6: Measure the LSTM Model by MSE, MAE and RMSE

Stop

3.4.2. Bi directional Long Short-Term Memory (Bi-LSTM)

Bi-LSTM memory network is one type of recurrent neural network that can be trained using the past and future input information with specified time. BiLSTM capture the context from forward and backward sequence by amalgamating the results of two recurrent neural networks that convey information in both forward and backward directions.

Algorithm 2: Train and evaluate Bidirectional Long Short Term Memory Model

Start

Input: Normalized sequenced training data

Output: Prediction of Crime type and Crime location

Initialization: epoch, batch size, lstm_unit, dropout rate

Step1: add Bi-LSTM Layer with defined bilstm_units and input data

add dropout regularization

add dense layer with defined output dimension and tanh activation function

Step2: Compile Bi-LSTM model with MSE loss function and Adam optimizer

Step3: Fit the training data to Bi-LSTM Model with initialized hyper parameter

While maximum iteration not reached ***do***

Train Bi-LSTM Model

Step4: Evaluate Bi-LSTM model with testing data

Step5: Predict unseen data by Bi-LSTM Model

Step6: Measure the Bi-LSTM Model by MSE, MAE and RMSE

Stop

3.4.3. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a component within a particular type of recurrent neural network designed to establish connections across a sequence of nodes, primarily for tasks involving memory and grouping, such as in speech recognition. GRUs are instrumental in addressing the vanishing gradient problem, a prevalent challenge encountered in recurrent neural networks, by adapting input weights within the neural network.

Algorithm 3: Train and evaluate Gated Recurrent Unit Model

Start

Input: Normalized sequenced training data

Output: Prediction of Crime type and Crime location

Initialization: epoch, batch size, gru_unit, dropout rate

Step1: add GRU Layer with defined gru_units and input data
add dropout regularization
add dense layer with defined output dimension and tanh activation function

Step2: Compile GRU model with MSE loss function and Adam optimizer

Step3: Fit the training data to GRU Model with initialized hyper parameter
While maximum iteration not reached ***do***
Train GRU Model

Step4: Evaluate GRU model with testing data

Step5: Predict unseen data by GRU Model

Step6: Measure the GRU Model by MSE, MAE and RMSE

Stop

3.4.4. Bidirectional Gated Recurrent Unit (Bi-GRU)

The Bidirectional Gated Recurrent Unit (Bi-GRU) is a component within a particular type of recurrent neural network designed for tasks in machine learning, utilizing connections across a sequence of nodes. What sets Bi-GRU apart from GRU is its ability to grasp context from both the forward and backward directions, enhancing its understanding of sequences.

Algorithm 4: Train and evaluate Bidirectional Gated Recurrent Unit Model

Start

Input: Normalized sequenced training data

Output: Prediction of Crime type and Crime location

Initialization: epoch, batch size, bi-gru, dropout rate

Step1: add Bi-GRU Layer with defined bi-gru and input data
add dropout regularization
add dense layer with defined output dimension and tanh activation function

Step2: Compile Bi-GRU model with MSE loss function and Adam optimizer

Step3: Fit the training data to Bi-GRU Model with initialized hyper parameter
While maximum iteration not reached ***do***
Train Bi-GRU Model

Step4: Evaluate Bi-GRU model with testing data

Step5: Predict unseen data by Bi-GRU Model

Step6: Measure the Bi-GRU Model by MSE, MAE and RMSE

Stop

3.5. Evaluation Techniques

To assess the RNN Crime prediction models, we employ a set of evaluation metrics, including Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Square. These metrics are favored in various studies due to their effectiveness. RMSE is particularly useful for gauging the discrepancy between predicted output and the original data, while MAE quantifies the absolute error between the original and predicted data. A smaller MAE value indicates a more accurate model. Mean Squared Error (MSE) is a statistical measure that averages the squared differences between predicted and actual values, and R-Square assesses the goodness of fit of our models.

3.6. Experimentation Tools

3.6.1. Python

Python is a versatile programming language that is object-oriented, interpreted, and suitable for a wide range of purposes. It has powerful natural language processing packages and text processing. It has a huge standard library, which provides rich functions and modules. We used Python programming language to implement the Crime prediction analysis.

3.6.2. Anaconda

Anaconda is enterprise platform for data science, which contains python for machine learning, deep learning and data science tasks. This is a distribution of the R and Python programming languages tailored for applications in predictive analysis, machine learning, deep learning, and data science.

3.6.3. Jupiter Notebook

Jupyter Notebook is a web-based application used for generating and distributing computational documents. It contains live code, equations, narrative text and visualizations It is additionally employed for tasks such as data cleansing, numerical simulation, data transformation, statistical modeling, data visualization, machine learning, and various other functions.

3.6.4. Libraries

First, we decided to install all necessary libraries and packages. These Libraries are described as follow:

- ***SciPy:*** SciPy is a popular library for creating 2D plots and graphs.
- ***NumPy:*** NumPy presents fashionable capabilities for mathematics operations, trigonometric and managing complicated numbers.
- ***Pandas:*** Pandas is a popular python tool used for read Data Frame.
- ***TensorFlow:*** sequential method used to define architecture of deep learning.

- ***mpl_toolkits:*** A Python package for plotting the multi vectors of Geometric Algebra, from the Clifford package, using matplotlib.
- ***Keras:*** used for cresting deep learning models.
- ***Minmax Scaler:*** it takes the minimum value and maximum value scale everything to fit between 0 and 1.
- ***Sklearn:*** is a standard library for both deep and machine learning.
- ***Matplotlib:*** used to plot different timeseries graphs.
- ***Axes3D:*** Mplot3d toolkit gives the essential capabilities used to create three-D floor plots.
- ***Seaborn:*** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

CHAPTER FOUR

4. PROPOSED CRIME PREDICTION MODELS

Here we depicted the overall proposed Crime prediction models, each process included in chronological order.

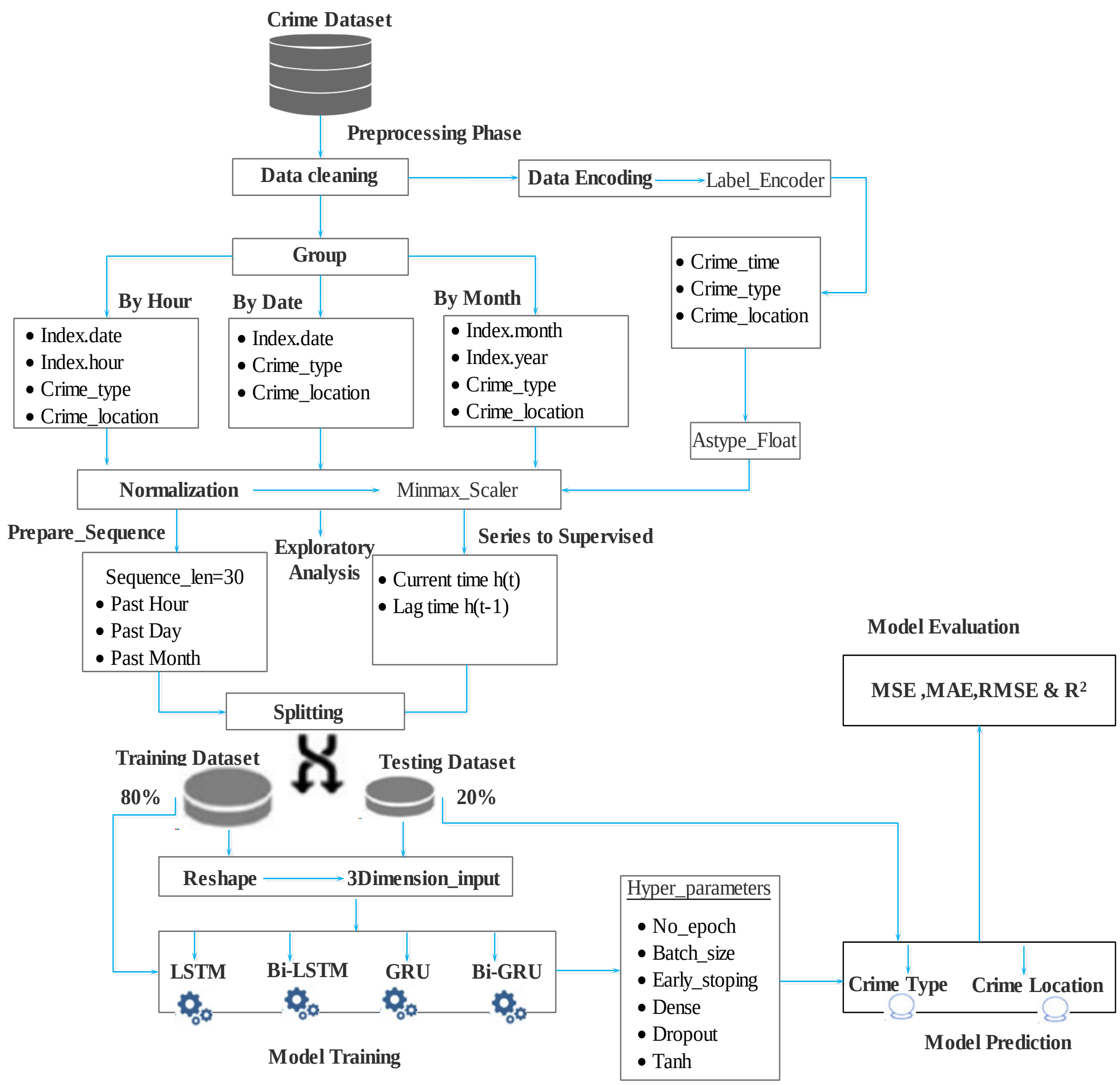


Figure 4.1. Proposed Crime Prediction Modeling

As shown in Figure 4.1 above, first we have collected dataset and preprocessed the dataset using a Label encoder to encode categorical values to integers such as Crime time, Crime type, and Crime location. We classified data into three volumes: hourly, daily and monthly. We converted integer values to floats using a function called astype(float). We used Min-max

Scalar to scale the value between 0 and 1.and also Time Series to Supervised transformation, to reframe time series data into a labeled learning format. Also prepared the sequence of data to consider past hour, day and month, and we analyze the data and visualize it in a plot graph to make sense of it. And we divide the data into 80% for training and 20% for testing, and convert the input into a three-dimensional shape that can easily be fed into models. We then train our RNN models using the training dataset and make predictions using the test dataset. RNN models like Long Short-Term Memory, Bidirectional-Long Short-Term Memory, Gated Recurrent Unit and Bidirectional Gated Recurrent Unit are included in the model-building phase. finally, we evaluate our model's performance using MSE, MAE, RMSE, and R_square.

4.1. Dataset Description

Data collection briefly discussed in chapter three, we collected the Crime record of Wolkite City and Abeshge district from the year 2008 to 2014 E.C. There are 9,820 Crime incidents in Wolkite and Abeshge district dataset and also, we include some data collected from Wolkite Zone Police that come from different district. For our proposed model our dataset has the following features.

- ***Date:*** The Crime incidences date
- ***Time:*** Timestamp of the Crime incident
- ***Crime type:*** Crime type that took place
- ***Location:*** Location name of the Crime type took place

4.2. Pre-Processing

Dataset preprocessing briefly described in the chapter three, during data pre-processing, we examine for any missing values, particularly when the raw data is often incomplete, error, outliers and inconsistent in then replace missing values by interpolation. Also, we transform the raw data in to machine understandable format mean that the categorical values change into numbers for this, we used Label encoder, grouped data into hourly, daily and monthly prediction, and scaled the data using Minmax scaler to scale each feature to fixed range (0,1). And also, we used series to supervised technique to reframed the data and prepared sequence of data with past hours, days and months as well as we explore the data, make to sense before trained the model and split the dataset into training and testing finally, RNN based models take three-dimensional input for that we reshape the input in to three dimensions (input, time stamp, feature).

4.3. Proposed Recurrent Neural Network Models

After preprocessing phase is finished, the next phase is model building. So, here we employed RNN based models such as: Long Short-Term Memory, Bidirectional Long Short-Term Memory, Gated Recurrent Unit and Bidirectional Gated Recurrent Unit. We described the structure and learning method of our model. For sequence of information or time series data or consider long time stamp past to predict the future output. An RNN (Recurrent Neural Network) is a type of artificial neural network designed for handling sequential data. RNNs are utilized in deep studying and withinside the improvement of fashions that simulate neuron hobby withinside the human brain. In particular effective in use instances wherein context is important to predicting an outcome, and also are awesome from other forms of synthetic neural networks due to the fact they use remarks loops to system a chain of information that informs the very last output. Each model we build described below.

4.3.1. Proposed LSTM Model

We opted for the Long Short-Term Memory (LSTM) model because our data is in the form of a time series. The LSTM is well-suited for such data due to its ability to maintain long-term dependencies. In contrast to conventional feedforward neural networks, LSTM possesses recurrent connections and is versatile, finding applications in various domains like cursive handwriting recognition, speech recognition, machine translation, robotic control, video gaming, healthcare, and the analysis of time series data. When making predictions in a time series, it's important to consider inputs from many time steps in the past, and the LSTM facilitates this by managing and accessing information through its gating mechanism. An LSTM cell represents a specialized variant of RNNs. Cell that is adept at managing relationships between distant elements in a sequence. It incorporates three gating mechanisms: forget gate (ft), input gate (it), and output gate (ot). Design of the LSTM cell is depicted in Figure 4.2[62].

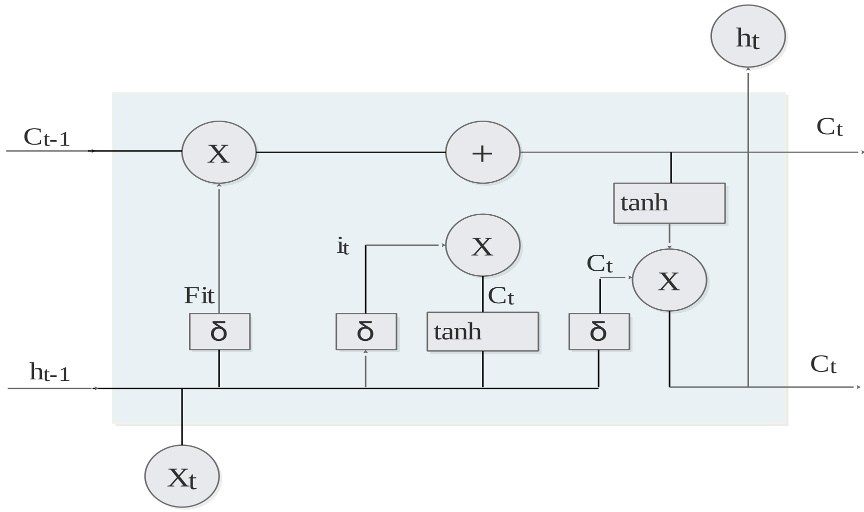


Figure 4.2. LSTM Cell

The mathematical expressions describing its behavior are as follows:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} * W \begin{pmatrix} h_{t}\text{-}1 \\ x_t \end{pmatrix} \tag{4.1}$$

$$Ct = f \oplus ct - 1 + i \oplus g$$

$$ht = o \oplus \tan h(Ct)$$

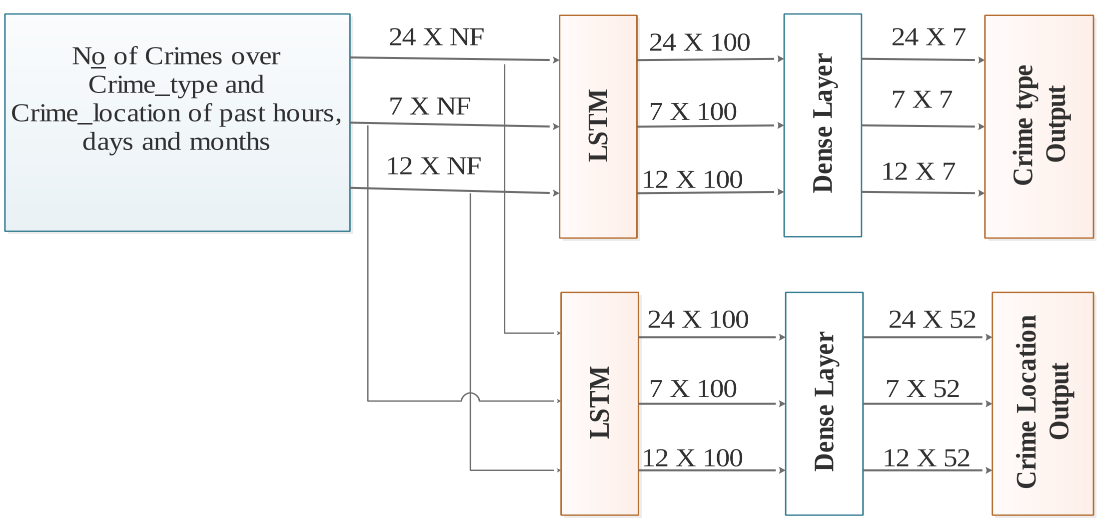


Figure 4.3. Proposed LSTM Model

As shown in the Figure 4.3 above, the proposed LSTM model takes numbers of Crimes for each Crime type and location as input. The input has Number of Features (NF), we have four features such as date, time, Crime type and Crime location, so the model takes datetime. i.e., (hour, day, month, year) and 100 unit of LSTM per layer and one dense layer. Dense layer has 7 units for Crime types and 52 units for Crime locations and produce the corresponding output.

4.3.2. Proposed Bi-LSTM Model

A Bidirectional Long Short-Term Memory (BiLSTM) network belongs to the family of recurrent neural networks and is capable of utilizing input information from both past and future time steps within a specified range. Bi-LSTMs effectively gather context by merging the outputs of two separate recurrent neural networks, each of which conveys information in opposite directions, encompassing both the forward and backward sequences.

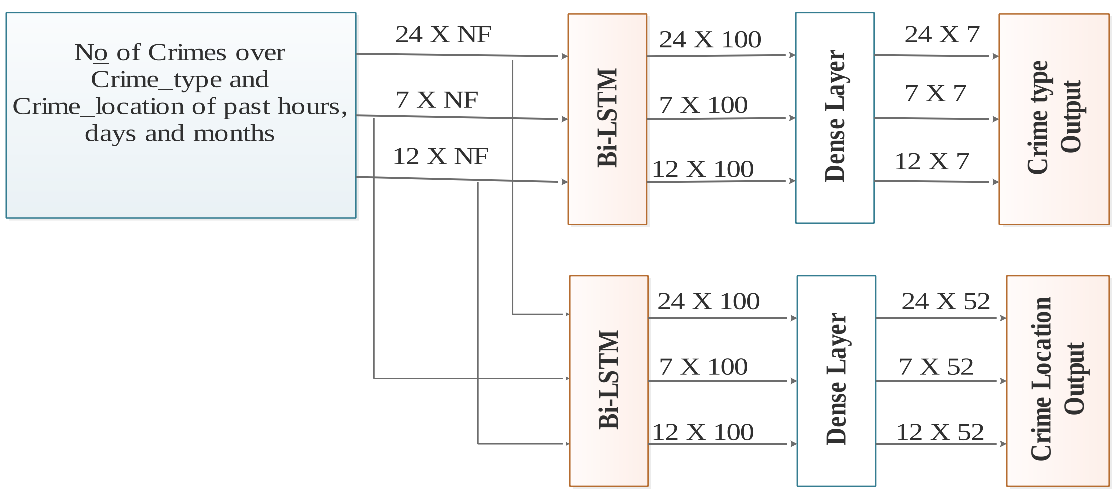


Figure 4.4. Proposed Bi-LSTM Model

As shown in the above Figure 4.4, Bi-LSTM take twenty-four hours, seven days in a week and twelve months in a year with other number of features (NF) like Crime type and location as input. The Bi-LSTM layer consists of 100 units of neurons. Following the Bi-LSTM layer, there is one dense layer in the network. This dense layer has 7 output units, corresponding to the 7 types of Crimes, and 52 output units, representing the 52 different locations. The dense layer performs computations on the information processed by the Bi-LSTM layer and generates the final output.

4.3.3. Proposed GRU Model

A Gated Recurrent Unit (GRU) is a fundamental component within a specific variant of recurrent neural networks (RNNs). Its primary purpose is to facilitate the execution of machine learning tasks that involve memory and sequence-based data clustering, such as those encountered in speech recognition applications. One of the noteworthy advantages of employing gated recurrent units lies in their inherent capacity to address the vanishing gradient problem, which frequently plagues conventional RNNs.

It's essential to underscore that, unlike LSTM and GRU units do not necessitate the use of a dedicated memory unit to regulate information flow. Instead, they directly harness all hidden states without additional control mechanisms. This streamlined architecture results in a reduced number of parameters, which in turn may lead to faster training or a decreased data requirement for effective generalization. Nonetheless, in scenarios involving extensive datasets, LSTMs, with their heightened expressiveness, may yield superior outcomes. In the context of GRUs (Gated Recurrent Units), two pivotal gates play a significant role: the reset gate and the update gate. The reset gate influences the manner in which new data is combined with existing memory, while the update gate controls the degree to which the previous state is preserved. Interestingly, the function of the update gate in GRUs is analogous to that of the input gate and the forget gate in LSTMs (Long Short-Term Memory networks).

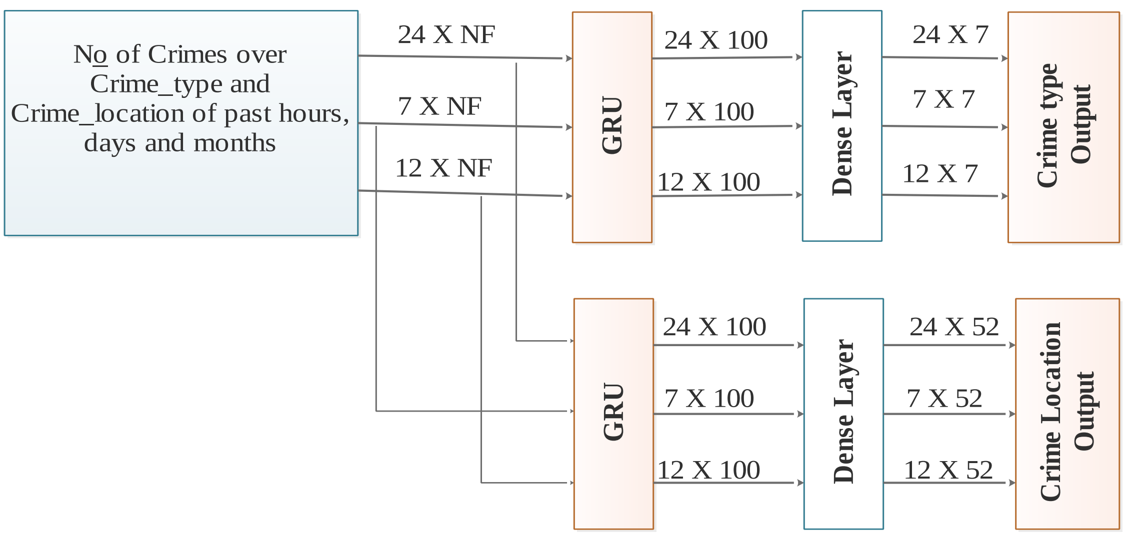


Figure 4.5. Proposed GRU Model

The proposed GRU model build with 100 units of GRU and one dense layer and it takes numbers of Crimes for each Crime type and location, twenty-four hours of a day, seven days of week and twelve months of a year. The input has Number of Features (NF), we have four features such as date, time, Crime type and Crime location, so the model takes datetime. i.e., (hour, day, month, year). Dense layer takes each hold 7 types of Crime and 52 locations for output. This shown in Figure 4.5.

4.3.4. Proposed Bi-GRU Model

A Bidirectional Gated Recurrent Unit (Bi-GRU) is a modified version of the typical Gated Recurrent Unit (GRU) designed to improve the model's capacity to grasp contextual information not only from past but also future time steps within a sequence.

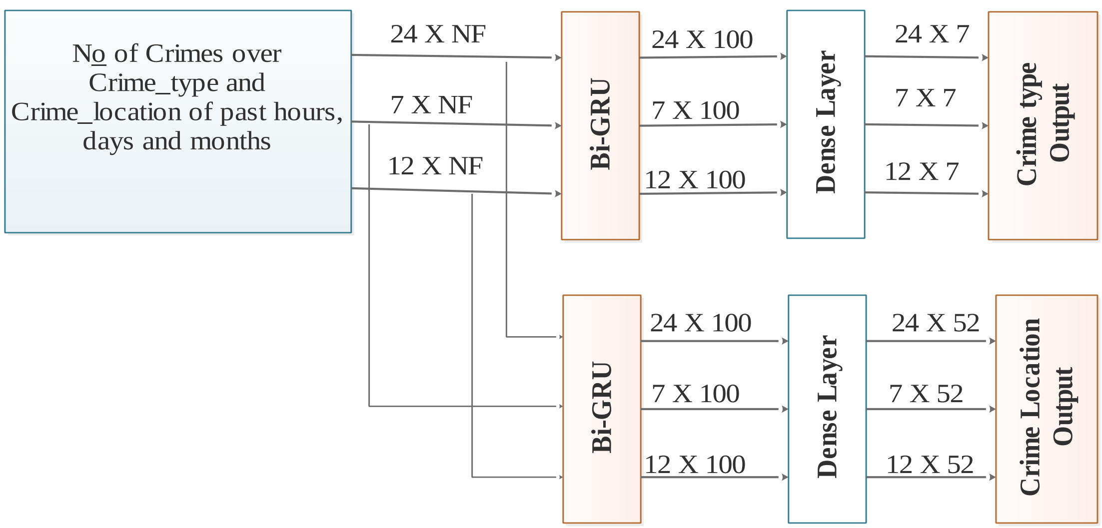


Figure 4.6. Proposed Bi-GRU Model

Our proposed Bi-GRU model illustrated in Figure 4.6, we build with 100 units of neurons and one dense layer and we feed numbers of Crimes for each Crime type and location, twenty-four hours of a day, seven days of week and twelve months of a year. The input has Number of Features (NF), we have four features such as date, time, Crime type and Crime location, so the model takes datetime. i.e., (hour, day, month, year). Dense layer consists of 7 types of Crime and 52 locations and give the corresponding output.

Table 4.1. LSTM, Bi-LSTM, GRU and Bi-GRU Model Parameters

Hyper Parameter	***Value***
No of instances	Four (4)
Sequence Size	Past hour, day, month consider: Past=30, future=1
	Series to supervised: Past =t-1, future= t+1
	Past day for future forecast: Past=14, future=1
Number of Unit per Layer	LSTM-100, Bi-LSTM-100, GRU-100 & Bi-GRU-100
	Dense-7 and Dense-52
Epochs	Epoch-30
Batch Size	Batch_size-64
Dropout	DrououtRate-0.2
Early_stopping	depending on testing Loss
	Patience: 5
	Reinstates Optimal Weight Configuration
Activation	Tanh
Optimizer	Adaptive Moment Estimation

The key model parameters are listed in the table provided in section 4.1 above.

4.4. Crime Prediction Models Evaluation

4.4.1. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a metric that quantifies the dissimilarity between two continuous variables. It is the absoluter error average of predicted and actual values. The calculation for Mean Squared Error (MSE) is as follows[62]:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

Where:

- n: signifies the total count of data points within our dataset.

- yi: signifies the genuine (actual) value for the i-th data point.
- ŷi: signifies the forecasted (predicted) value for the i-th data point.

4.4.2. Mean Square Error (MSE)

The Mean Squared Error (MSE) is a statistical metric that computes the mean of the squared variances between predicted values and the real (true) values. Here's how the Mean Squared Error (MSE) is computed[11], [62]:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.3}$$

Where:

- n: signifies the total count of data points within our dataset.
- yi: signifies the genuine (actual) value for the i-th data point.
- ŷi: signifies the forecasted (predicted) value for the i-th data point.

4.4.3. Root Mean Square Error

The Root Mean Square Error (RMSE) is a statistical measurement employed to assess the precision of a predictive model, often in regression analysis. It gauges the typical size of discrepancies (the disparities between predicted values and real values) within a dataset. The calculation for the Root Mean Squared Error (RMSE) is as follows[45]:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.4}$$

Where:

- n: signifies the total count of data points within our dataset.
- yi: signifies the genuine (actual) value for the i-th data point.
- ŷi: signifies the forecasted (predicted) value for the i-th data point.

4.4.4. R-squared or R^2

The Coefficient of Determination (R-squared or R^2) is a statistical metric that indicates the fraction of variability in the outcome variable that can be accounted for by a model. It tells us how well the actual values match the predicted values. We have a model that predicts values, and we want to assess how good that model is at explaining the variation in our data. R-squared comes into play. If R^2 is close to 1, it means the model is doing an excellent job of explaining the variation. On the other hand, if R^2 is close to 0, the model is not explaining much of the variation, and it might not be a good fit for the data[63]. The calculation for the Coefficient of Determination (R-squared or R^2) is as follows[62]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (4.5)$$

Where:

- n: represents the quantity of data points within our dataset.
- y_i: represents the actual value
- $\hat{y}_i$: represents the forecasted value
- $\bar{y}$: represents the average of the actual values.

CHAPTER FIVE

5. IMPLEMENTATION

5.1. Sample Collected Dataset

As shown in table 5.1, the total sample data collection of Wolkite City shows the data from 2008 to 2014 E.C. Our dataset has features like date, months, year, time, type and location.

Table 5.1. Samples Prepared Datasets

Datetime	*Crime date*	*Crime time*	*Crime type*	*Crime location*
2008-01-01 04:12:00	2008-01-01	04:12:00	Murder	Hudad seven
2008-01-01 04:50:00	2008-01-01	04:50:00	Fraud	Chisa
2008-01-01 05:30:00	2008-01-01	05:30:00	Fraud	Nacha kulit
2008-01-01 05:30:00	2008-01-01	05:30:00	Assault	Kamb sefer
2008-01-01 08:20:00	2008-01-01	08:20:00	Drug	Hudad seven
...	...	...	...	...
2014-12-30 05:09:00	2014-12-30	05:09:00	Fraud	Bole sefer
2014-12-30 13:54:00	2014-12-30	13:54:00	Robbery	Bole sefer
2014-12-30 14:00:00	2014-12-30	14:00:00	Assault	Kulit two
2014-12-30 17:37:00	2014-12-30	17:37:00	Drug	Wolkite School
2014-12-30 20:30:00	2014-12-30	20:30:00	Robbery	Abuko

The above sample dataset loaded using the following code.

```
# Load the collected dataset
data = pd.read_csv('Crime_dataset.csv')
data['Crime_date'] = pd.to_datetime(data['Crime_date'])
data = data.set_index('Crime_date')
data['Crime_time'] = pd.to_datetime(data['Crime_time']).dt.time
print(data)
```

Figure 5.1. Loading the Dataset Sample Code

Figure 5.1. It shows that we have loaded the prepared dataset and converted the Crime_date and Crime_time to the correct datetime format and indexed the Crime_date

5.2. Data Pre-processing Techniques

5.2.1. Data Cleaning

Data cleaning is the process of cleaning our dataset. In steady of eliminating the rows data we used interpolation technique, why because rows may contain necessary information. The following sample code shows replace missing values with interpolation.

```
def replace_missing (attribute):
    return attribute.interpolate(inplace=True)
replace_missing(series['Crime_date'])
replace_missing(series['Crime_time'])
replace_missing(series['Crime_type'])
replace_missing(series['Crime_location'])
```

Figure 5.2. Interpolate Sample Code

5.2.2. Data Encoding

Categorical data encoding is converting the categorical values to numbers in order to train the model easily.

```
values = series.values
encoder = LabelEncoder()
values[:,0] = encoder.fit_transform(values[:,0])
values[:,1] = encoder.fit_transform(values[:,1])
values[:,2] = encoder.fit_transform(values[:,2])
values =values.astype('float32')
print(values)
```

Figure 5.3. Data Encoding Sample Code

5.2.3. Data Grouping

Data grouping is arranging data to easily predict the Crime with three granularities including hourly, daily and monthly. As shown in the following Figure 5.4, we grouped our data with hourly, daily, monthly. This is done using groupby method, it takes parameter of data. index. date, data. index_hour, data. index month, Crime_type and Crime Location.

```python
# For Hourly prediction
data_hourly = data.groupby([data.index.date, data.index.hour, 'Crime_type']).size().unstack(fill_value=0)
hourly_scaler = MinMaxScaler()
scaled_hourly_data = hourly_scaler.fit_transform(data_hourly)

# For Daily
daily_data = data.groupby([data.index.date, 'Crime_type']).size().unstack(fill_value=0)
daily_scaler = MinMaxScaler()
scaled_daily_data = daily_scaler.fit_transform(daily_data)

# For Monthly
monthly_data = data.groupby([data.index.year, data.index.month, 'Crime_type']).size().unstack(fill_value=0)
monthly_scaler = MinMaxScaler()
scaled_monthly_data = monthly_scaler.fit_transform(monthly_data)
```

Figure 5.4. Data Grouping Sample Code for Crime Type

5.2.4. Data Scaling

We used Minmax scaling method to scale values between 0 and 1. This is achieved by taking the difference between the minimum value and dividing it by the range, which is the difference between the maximum and minimum values. In the following Table 5.2 we scaled Crime type, and in Table 5.3 scaled Crime location.

Table 5.2. Scaled Value of Crime Type

No	***Categorical Values***	***Scaled Value***
1	Robbery	0.8333334
2	Beat	0.16666667
3	Fraud	0.5
4	Assault	0.0
5	Violation	1.0
6	Drug	0.33333334
7	Murder	0.6666667

As shown in the above Table 5.2, we scaled the encoded Crime type values between 0 and 1. From Table 5.2, the maximum value is 1.0 (Violation) and the minimum value is 0.0 (Assault).

Table 5.3. Scaled Value of Crime Location

No̲	***Crime Location***	***Scaled Values***	***No̲***	***Crime Location***	***Scaled Values***
1	Embelta	0.29411766	27	Jejebana gasore	0.54901963
2	Darge	0.25490198	28	Kulit two	0.62745100
3	Chisa	0.19607843	29	Hudad five	0.47058827
4	Hudad seven	0.4901961	30	Lachena omancho	0.64705884
5	Borer	0.1764706	31	Boketana serite	0.13725491
6	Walga	0.92156863	32	Gebreal sefer	0.3529412
7	Nacha kulit	0.7647059	33	Raza sefer	0.78431374
8	Kera sefer	0.5882353	34	Gubre	0.4117647
9	Bole sefer	0.15686275	35	Mariam sefer	0.6862745
10	Arsema sefer	0.05882353	36	College sefer	0.21568628
11	Condominium sefer	0.23529413	37	Jayka School	0.5294118
12	Adebabay sefer	0.01960784	38	Alfa sefer	0.03921569
13	Wolkite High School	0.9803922	39	Wolkite	0.9607844
14	Selamber sefer	0.84313726	40	Gunchre	0.43137255
15	Wolkite School	1.0	41	Becho sefer	0.09803922
16	Kamb sefer	0.5686275	42	Kolotera sefer	0.60784316
17	Menaharia sefer	0.74509805	43	Sebakuteba sefer	0.8235294
18	Mchlena tereko	0.15686275	44	Hayakuteba sefer	0.4509804
19	Gibo bare	0.05882353	45	Mazoriya	0.7058824
20	Bido tedele	0.11764707	46	Enemor sefer	0.3137255
21	Fite jija	0.33333334	47	Welane sefer	0.94117653
22	Tatesa	0.88235295	48	Geritera sefer	0.37254903
23	Mamede	0.6666667	49	Rohbot School	0.8039216
24	Dire lafto	0.27450982	50	Basket sefer	0.07843138
25	Tawla	0.9019608	51	Stadium sefer	0.8627451
26	Abuko	0.0	52	JandP sefer	0.50980395

As shown in the above Table 5.3, we scaled the encoded Crime location using Min max scalar, the maximum value is 1.0 (Wolkite School) and the minimum value is 0.0 (Abuko).

5.2.5. Data Splitting

We allocate 80% of the data for training, and the remaining data will be used for testing.

```
# Placeholder for the split_train_test function
def split_train_test(X, y):
    train_size = int(0.8 * len(X))
    X_train, X_test = X[:train_size], X[train_size:]
    y_train, y_test = y[:train_size], y[train_size:]
    return X_train, X_test, y_train, y_test
```

.

Figure 5.5. Data Splitting Sample Code

5.3. Exploratory Data Analysis

We explore proportion of Crime according to Crime type.

```
premise_type = series.groupby('Crime_type').size()
premise_type.head()
labels = series['Crime_type'].unique()
count = series['Crime_type'].value_counts()

explode = (0, 0, 0, 0, 0,0,0)

fig, ax = plt.subplots(figsize = (6,6))
ax.pie(count, explode=explode, labels=labels, autopct='%1.1f%%',shadow=True, startangle=140)
ax.axis('equal')
plt.title("Proportion of Crime according to Crime Type", color='black', fontsize=12)
plt.show()
```

Figure 5.6. Plotting Count Percentage Sample Code

As shown in the Figure 5.6. Sample Code to plot 3D axis graph of Crime type and their percentage values and in then the output illustrated in the Figure 5.7 as follows.

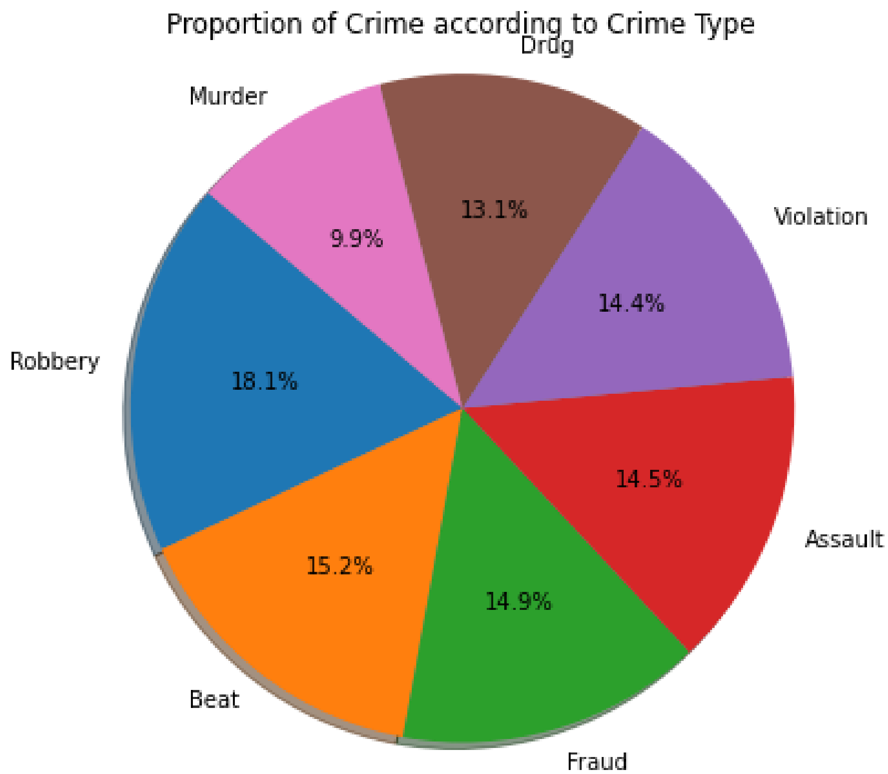


Figure 5.7. Number of Cout Percentage over Crime Category

As shown in the above Figure 5.7 the number of Crime type percentage represented with seven different colors, so it indicates the highest Crime type percentage is Robbery which is 18.1 % and the lowest Crime type percentage is Murder, which is 9.86% in our dataset. The rest of Crime type as pie chart indicates Beat has 15.2 %, Fraud has 14.9 %, Assault cover 14.5 %, Violation has 14.4 % and Drug has 13.1 %. To plot this pie chart, we have used groupby method this method is used for grouping time series dataset based on Crime type. And also, we specified the labels and counts using unique () and value_counts () method. Unique method used for capture unique Crime types from the dataset and value counts methods used to count the value of unique Crime types and other methods like subplots, pie, and show are applied. The sample code of this process depicted in Figure 5.6 above.

Table 5.4. Number of Counts over Each Location and Percentage

No	Location	Counts	Percentage	No	Location	Counts	Percentage
1	Walga	589	5.99 %	27	Selamber sefer	168	1.71 %
2	Darge	554	5.64%	28	Wolkite School	155	1.58 %
3	Borer	498	5.07%	29	Adebabay sefer	154	1.57 %
4	Chisa	473	4.82 %	30	Raza sefer	154	1.57 %
5	Hudad seven	451	4.59 %	31	JandP sefer	153	1.56%
6	Nacha kulit	405	4.12%	32	Condominium sefer	152	1.55 %
7	Abuko	390	3.97 %	33	Mariam sefer	137	1.39 %
8	Gibo bare	359	3.65 %	34	Gubre	48	0.49%
9	Embelta	341	3.47 %	35	College sefer	31	0.316%
10	Kulit two	340	3.46%	36	Kera sefer	27	0.27 %
11	Fite jija	304	3.09 %	37	Wolkite	25	0.25 %
12	Tatesa	302	3.07%	28	Jayka School	24	0.24 %
13	Bido tedele	300	3.05 %	39	Becho sefer	18	0.183%
14	Tawla	279	2.84%	40	Gunchre	18	0.18 %
15	Mamede	274	2.79%	41	Kolotera sefer	15	0.15 %
16	Jejebana gasore	270	2.75 %	42	Sebakuteba sefer	15	0.15 %
17	Bole sefer	268	2.73 %	43	Alfa sefer	11	0.11 %
18	Menaharia sefer	244	2.48%	44	Arsema sefer	11	1.33 %
19	Dire lafto	235	2.39 %	45	Hayakuteba sefer	10	0.10%
20	Hudad five	234	2.38%	46	Mazoriya	9	0.09 %
21	Lachena omancho	233	2.37%	47	Enemor sefer	7	0.07 %
22	Boketana serite	230	2.34 %	48	Welane sefer	7	0.07 %
23	Gebreal sefer	203	2.07 %	49	Geritera sefer	6	0.06 %
24	Wol.High School	200	2.04%	50	Rohbot School	6	0.06 %
25	Kamb sefer	183	1.86 %	51	Basket sefer	5	0.05%
26	Mchlena tereko	170	1.73 %	52	Stadium sefer	5	0.05 %

As Table 5.4 above shows, number of counts in every location and it's percentage putted. From table we understand the highest Crime count location is Walga which is 589 counts and 5.99% and also Basket sefer and Stadium sefer which have 5 count and 0.05%, so each location Crime counts and percentage arranged in decreasing order in the table.

```python
# Get unique crime types and colormap
unique_crime_types = data['Crime_type'].unique()
# Choose a colormap with enough colors
color_map = plt.cm.get_cmap('tab10', len(unique_crime_types))
# Count occurrences of each crime type
crime_type_counts = data['Crime_type'].value_counts()
# Create a bar plot with different colors for each crime type
plt.figure(figsize=(5, 5))

for idx, (crime_type, count) in enumerate(crime_type_counts.items()):
    plt.bar(crime_type, count, color=color_map(idx))

plt.title("Crime Type Counts")
plt.xlabel("Crime Type")
plt.ylabel("Number of Crimes")
plt.xticks(rotation=45)
plt.show()
```

Figure 5.8. Plotting Crime Type Sample Code

Figure 5.9 below, represented the sample code for plotting unique Crime type counts value.

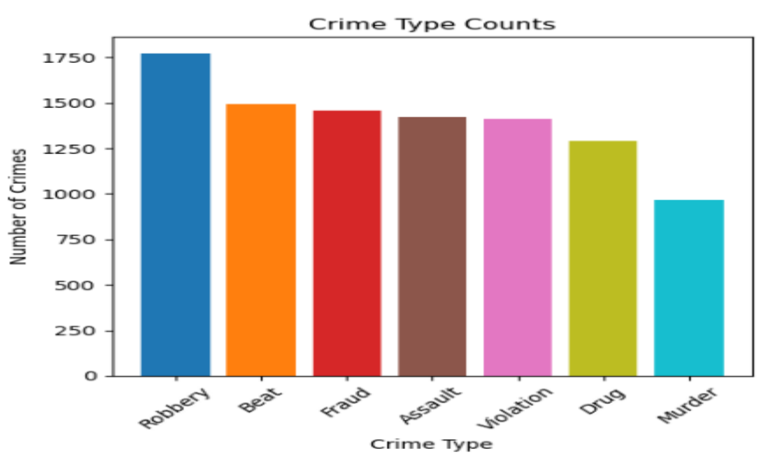


Figure 5.9. Number of Cout over Each Crime Category

As shown in the above Figure 5.9, We visualize Crime types using histogram representation. In the figure we have drawn 7 Crime type in different color, so we understand the highest Crime count is Robbery which is blue color and the lowest count is Murder which is cyan color in our Crime dataset.

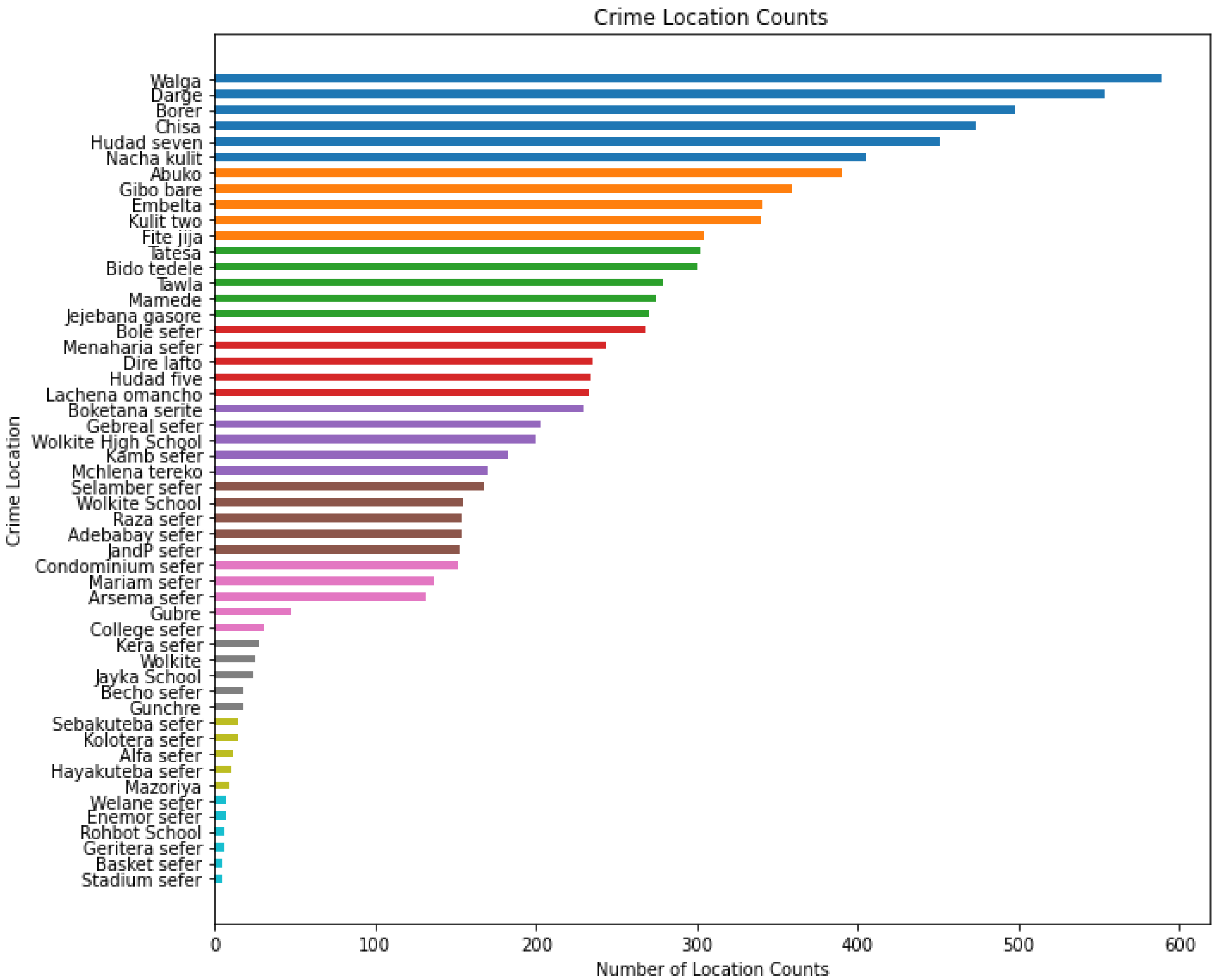


Figure 5.10. Number of Count over each Crime Location

As shown in the above Figure 5.10, We visualize Crime locations using histogram representation. The figure picks ten different color by grouping with number of counts. For example, location Walga, Darge, Borer, Chisa, Hudad seven and Nacha kulit picks blue color which indicates higher Crime count percentage on the other hand Mazoriya, Haya kuteba, Enemor sefer, Welane sefer Geritera sefer, Rohbot school, Basket sefer and Stadium sefer are picks cyan color which is lowest number of Crime rate, so fifty-two locations arranged in group of count to pick different color based on percentage range.

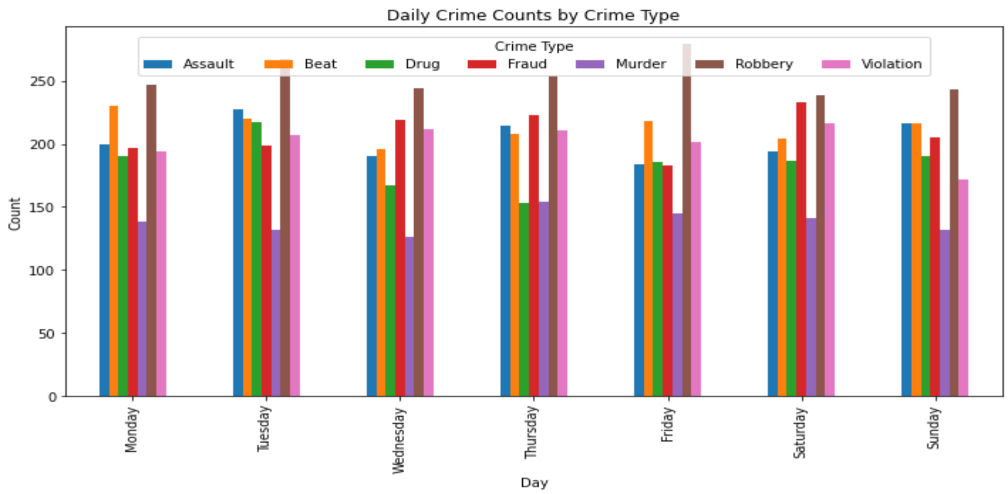


Figure 5.11. Common Crime Type over Days

As shown in the Figure 5.11, the most Crime occurred throughout the week is Robbery and next to this Fraud, Beat and Assault are most common Crime in the week. Crime type Fraud is higher frequency count in three days like Wednesday, Thursday and Saturday, Beat also occurs on Monday, Friday and Sunday and other is Assault, it occurs on Tuesday and Sunday. This frequency values detail shown in Table 5.5.

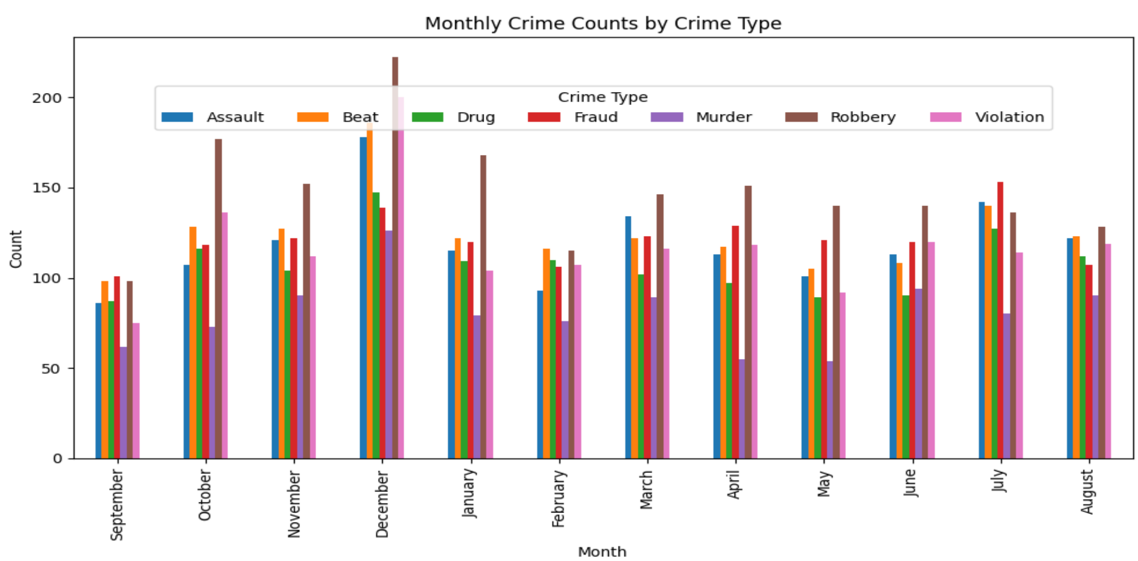


Figure 5.12. Common Crime Type over Months

Figure 5.12 represents Most common Crime type throughout the months, Robbery is the most occurred Crime in most months but in specific month February Beat is highly occurred and in

July and September Fraud is highest frequency count, but in other months Robbery is the highest. The detail Crime counts values over each month seated in Table 5.6.

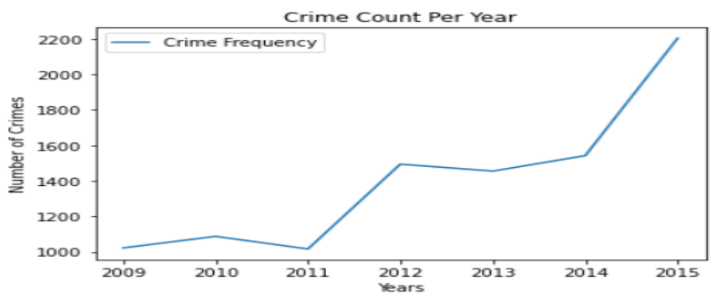


Figure 5.13. Number of Crime Counts Per Years

As shown in the above Figure 5.13, number of Crimes counts per year. From figure we observe that Crime count increase from 2012 to 2014. Specially in the year 2014 there is higher Crime rate.

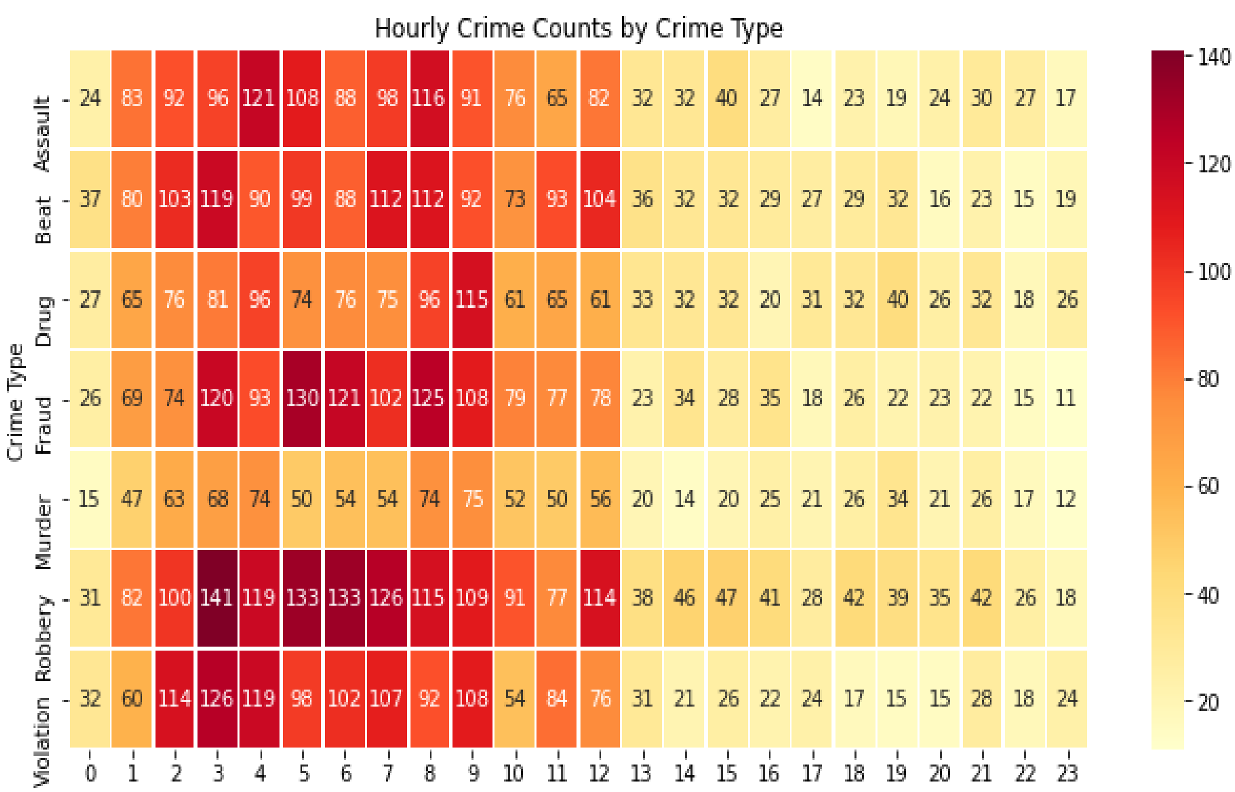


Figure 5.14. Number of Crime Type Counts by Hour

As shown in Figure 5.14, during Hour 1, the most frequently occurring Crime type is Beat with 37 occurrences. In Hour 2, Assault is most occurred with 83 occurrences. As we move to Hour 3, Violation takes the lead with 114 occurrences, while in Hour 4, Robbery becomes the most frequently reported Crime type, reaching a count of 141. Hour 5 sees a rise in Assault incidents with 121 occurrences. Moving forward to Hour 6, Robbery takes the spotlight with 133 occurrences. In Hour 7, Robbery remains a prevalent Crime type, reaching 133 occurrences. This trend continues in Hour 8, with Robbery recorded at 126 occurrences. Hour 9 witnesses an increase in Fraud incidents with 125 occurrences. During Hour 10, Drug becomes the most frequently occurring Crime type, reaching 115 occurrences. As we approach Hour 11, Robbery maintains a significant presence with 91 occurrences. Hour 12 sees a surge in Beat incidents with 93 occurrences, and Hour 13 Robbery become more frequently occurred with 114 occurrences. In Hour 14, Robbery with 38 occurrences. During Hour 15, Robbery rises to the forefront with 46 occurrences. Hour 16 maintains a similar trend, with Robbery reported at 47 occurrences. Moving to Hour 17, Robbery takes the lead with 41 occurrences. In Hour 18, Drug with 31 occurrences. Hour 19 sees Robbery continuing to be prominent with 42 occurrences Robbery. Hour 20 witnesses a rise in Drug incidents, reaching 40 occurrences. In Hour 21, remains frequent with 35 occurrences. Hour 22 exhibits Robbery incidents at 42 occurrences, and in Hour 23, Assault becomes the prominent Crime type with 27 occurrences. Finally, in Hour 24, Drug takes the lead with 26 occurrences

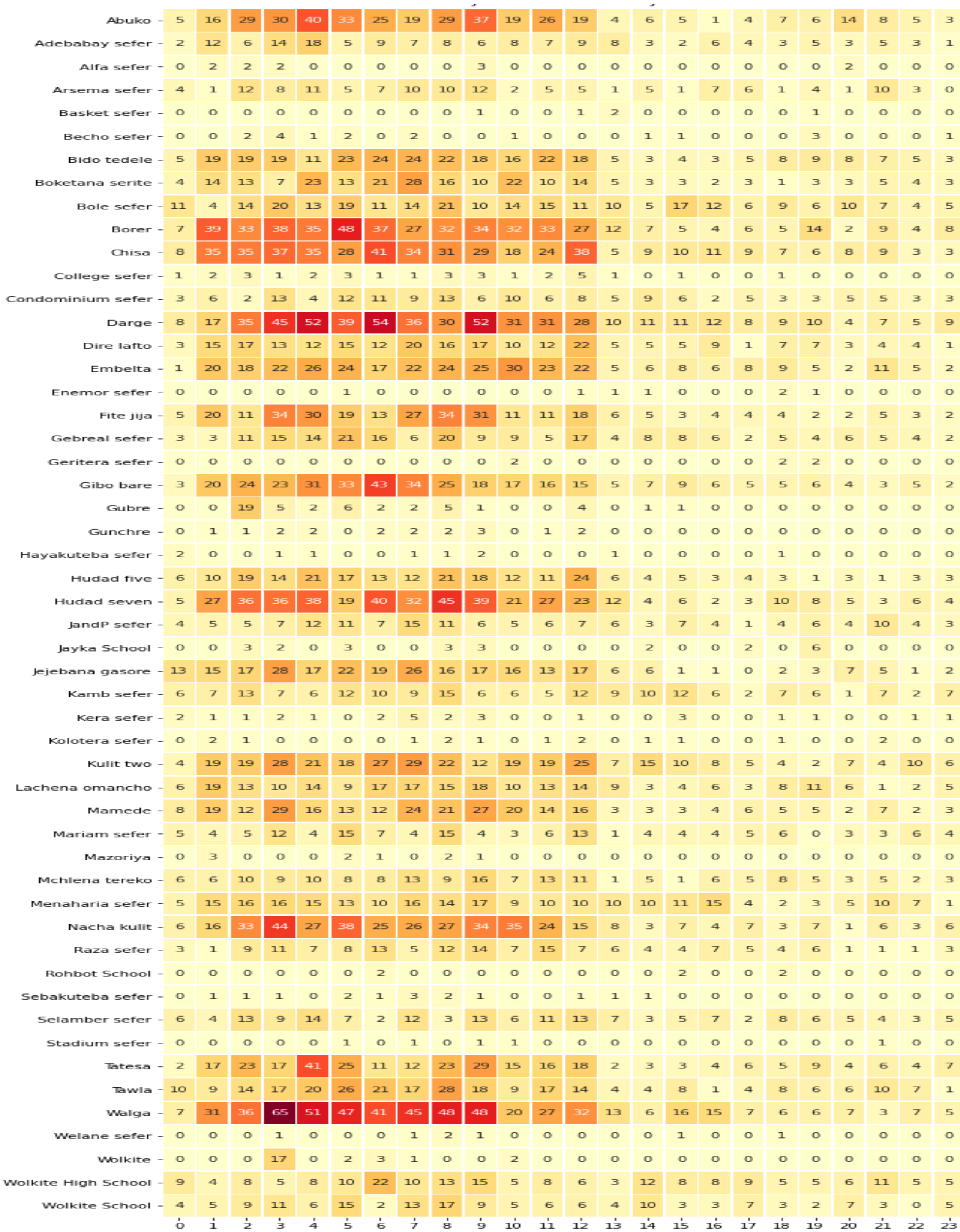


Figure 5.15. Number of Location Counts by Hour

As shown in Figure 5.15, in Hour 1, the most frequently occurred location is Jejebana gasore with 13 occurrences. In Hour 2, Borer has highest occurrences which is 39. In Hour 3, the highest occurrences are 36 in both Hudad seven and Walga. In Hour 4, the most occurred location Walga with 65 occurrences. In Hour 5, Darge is the highest with 52 occurrences. In Hour 6, the most frequently occurred Crime location is Borer with 48 occurrences. In Hour 7, Darge is the highest occurrence value which is 54. In Hour 8, Walga is highest occurrences that is 45. In Hour 9, also continues Walga with 48 occurrences. In Hour 10, Darge is the highest occurrences which is 52. In Hour 11, Nacha Kulit is highest frequency with 35 occurrences. In Hour 12, Borer is the highest occurrences with 33. In Hour 13, Chisa is the most occurred with 38 occurrences. In Hour 14, Walga is the highest occurrences value which is 13. In Hour 15, Kulit two is the highest occurrences with 15. In Hour 16, Bole sefer is the highest occurrences with 17. In Hour 17, both Walga and Menaharia sefer have highest frequency with 15occurrences. In Hour 18, in the same way Wolkite High School and Chisa has highest occurrences with 9. In Hour 19, Hudad seven is highest occurrences with 10. In Hour 20, the most Frequently appear Crime location is Borer with 14 occurrences. In Hour 21, Abuko is the Highest occurrences with 14. In Hour 22, Both Embelta and Wolkite High School have the highest occurrence with the same 11 counts. In Hour 23, Kulit two is the highest with 10 occurrences. In Hour 24, most occurred location is Darge with 9 counts.

Table 5.5. Crime Type Counts over Week

No	***Crime Category***	***Seven Dys of a Week***						
		0	***1***	***2***	***3***	***4***	***5***	***6***
1	Assault	200	227	190	214	184	194	216
2	Beat	230	220	196	208	218	204	216
3	Drug	190	217	167	153	186	187	190
4	Fraud	197	199	219	223	183	233	205
5	Murder	138	132	126	154	145	141	132
6	Robbery	247	267	244	254	279	239	243
7	Violation	194	207	212	211	201	216	172

As shown in Table 5.5 Crime type counts over week, throughout the week, different types of Crimes occur with varying frequencies. On Mondays (day 0), we see a relatively high number of robberies, with 247 incidents reported, as well as a significant number of beats at 230 cases. Tuesdays (day 1) continue to see a spike in Crime, with 267 robberies and 227 assaults. Wednesdays (day 2) see a decrease in both robbery and assault but an increase in fraud cases at 219. Thursdays (day 3) witness a rise in robberies again, with 254 instances, while fraud cases remain high at 223. Fridays (day 4) see the highest number of robberies during the week, reaching 279, and beat-related incidents at 218. Saturdays (day 5) and Sundays (day 6) tend to have more balanced Crime distributions, with Saturdays having a notable number of fraud cases at 233 and Sundays showing a decrease in violations with 172 cases. These trends highlight the varying patterns of criminal activity throughout the seven days of the week. In general, throughout the week robberies has highest incident.

Table 5.6. Crime Type Counts over Year

N<u>o</u>	***Crime_type***	***12 Months over Year***											
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***
1	Assault	86	107	121	178	115	93	134	113	101	113	142	122
2	Beat	98	128	127	186	122	116	122	117	105	108	140	123
3	Drug	87	116	104	147	109	110	102	97	89	90	127	112
4	Fraud	101	118	122	139	120	106	123	129	121	120	153	107
5	Murder	62	73	90	126	79	76	89	55	54	94	80	90
6	Robbery	98	177	152	222	168	115	146	151	140	140	136	128
7	Violation	75	136	112	200	104	107	116	118	92	120	114	119

As shown in Table 5.6 Crime type counts over months, the output shows the index (month) along with the corresponding most common Crime type for each month. In September (month 1), the most common Crime type is Fraud, in October (month 2), the most common Crime type is Robbery, in November (month 3), the most common Crime type is Robbery and in December (month 4), the most common Crime type is Robbery, in January (month 5), the most common Crime type is Robbery, in February (month 6), the most common Crime type is Robbery, in March (month 7), the most common Crime type is Robbery, in April (month 8), the most common Crime type is Robbery, in May (month 9), the most common Crime type is Robbery, in June (month 10), the most common Crime type is Robbery, in July (month 11), the most common Crime type is Fraud, in August (month 12), the most common Crime type is Robbery. So, throughout the months the highest occurrence of Crime is Robbery in our dataset.

5.4. Experimental Setup

Deep learning experimentation required high processor and graphical processing unit supported computing. In this study, we used a computer with a memory capacity of 8 gigabyte random access memory (RAM) and 2.4 GHz processor and 64-bit windows 11 operating system.

5.5. Build Crime Prediction Model

In this study we create a model LSTM, Bi-LSTM, GRU and Bi-GRU for predicting Crime type and location. The models aim to predict Crime patterns using time series analysis. The model consists of Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU) and Bidirectional Gated Recurrent Unit (Bi-GRU) models and also designed for hourly, daily, and monthly Crime type and Crime location predictions. These models leverage the temporal nature of Crime data to capture patterns and make accurate predictions.

For the hourly prediction model, we trained each model for 30 epochs using a batch size of 64. The LSTM, Bi-LSTM, GRU and Bi-GRU layer has 100 units, enabling it to learn complex dependencies within the Crime time series data. To prevent overfitting, a dropout rate of 0.2 is applied, allowing the model to generalize well to unseen Crime patterns. The tanh activation function is used to introduce non-linearity into the model. The model is trained using the mean squared error (MSE) loss function, which measures the discrepancy between predicted and actual Crime counts. We utilized the Adam optimizer to optimize the model's weights during training. To prevent overfitting early stopping with a patience of 5 is implemented, which restores the best weights based on validation loss.

The architecture and training process for the daily and monthly prediction models are similar to the hourly model. These models are designed to capture higher-level Crime patterns that occur over longer time intervals. The hyperparameters, including the number of epochs, batch size, LSTM units, dropout rate, activation function, loss function, optimizer, and early stopping configuration, remain consistent across all four models.

After training each model, we evaluated their performance using separate test datasets specific to hourly, daily, and monthly predictions. The test loss values were calculated using the MSE loss metric, providing insights into the accuracy of the predictions.

```python
# Build model for hourly prediction
model_hourly = Sequential()
model_hourly.add(LSTM(units=lstm_units, input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_hourly.add(Dropout(rate=dropout_rate))
model_hourly.add(Dense(units=y_train_hourly.shape[1], activation='tanh'))
model_hourly.compile(loss='mse', optimizer='adam')
early_stopping_hourly = EarlyStopping(patience=5, restore_best_weights=True)
# Build model for daily prediction
model_daily = Sequential()
model_daily.add(LSTM(units=lstm_units, input_shape=(X_train_daily.shape[1], X_train_daily.shape[2])))
model_daily.add(Dropout(rate=dropout_rate))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_daily.compile(loss='mse', optimizer='adam')
early_stopping_daily = EarlyStopping(patience=5, restore_best_weights=True)
# Build model for monthly prediction
model_monthly = Sequential()
model_monthly.add(LSTM(units=lstm_units, input_shape=(X_train_monthly.shape[1], X_train_monthly.shape[2])))
model_monthly.add(Dropout(rate=dropout_rate))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
model_monthly.compile(loss='mse', optimizer='adam')
early_stopping_monthly = EarlyStopping(patience=5, restore_best_weights=True)
```

Figure 5.16. Sample LSTM Model Building Code

5.5.1. Creating Input

Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bi-directional Long Short-Term Memory (BiLSTM), and Bidirectional Gated Recurrent Unit accept a three-dimensional input structure with dimensions as follows (number_samples, number_timesteps, number_features). To reconfigure this input, we establish a utility function called prepare sequences, where we set the sequence length to 30. This implies that the model makes predictions based on the earlier 30 hours, days, and months of data.

```python
# Number of previous hours/days/months to consider
sequence_length = 30

def prepare_sequences(data, sequence_length):
    X = []
    y = []
    for i in range(sequence_length, len(data)):
        X.append(data[i - sequence_length:i])
        y.append(data[i])
    return np.array(X), np.array(y)
```

Figure 5.17. Input Sequence Sample Code

5.5.2. Dropout

Dropout regularization randomly drop 20% of unit from the network. Keras offers support for the Dropout technique, which is available through the Dropout core layer. It prevents overfitting by ignoring randomly selected neurons in training. Often used 0.2 dropout regularization to get good result.

5.5.3. Dense Layer

A Dense layer is synonymous with a fully connected layer that accept input and transform to output using activation function. Dense layer consists of output dimension and activation function. We used tanh activation function whatever input is maps to -1 and 1.

```
model_hourly.add(Dense(units=y_train_hourly.shape[1],activation='tanh'))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
```

Figure 5.18. Adding Dense Layers Sample Code

5.5.4. Compile the Model

After the dense layer is defined the model needs to be compiled. In order to compile the model, the loss and optimizer used passed as a parameter. We used mean square error loss function and adam optimizer. Model compiled depicted in Figure 5.19 sample code.

```
# Compile the model
model_hourly.compile(loss='mse', optimizer='adam')
model_daily.compile(loss='mse', optimizer='adam')
model_monthly.compile(loss='mse', optimizer='adam')
```

Figure 5.19. Compile the Model Sample Code

5.5.5. Optimizer

We employed the Adam optimizer, which is a blend of two optimization techniques: the Adaptive Gradient Algorithm and Root Mean Square Propagation (RMSprop). The Adaptive Gradient optimizer adjusts the learning rate for each dimension dynamically, particularly benefiting problems with sparse gradients. It calculates the learning rate based on previous gradient information.

5.5.6. Fit the models

We establish a fit model operation to take the model and train it using the training data for 30 epochs with a batch size of 64. Additionally, we allocate 20% of the training data for validation purposes, with the shuffle parameter set to false for improved performance.

```python
# Define hyperparameters
epochs = 30
batch_size = 64

# Restore the best weight
early_stopping = EarlyStopping(patience=5, restore_best_weights=True)
# Fit the Model
LSTM_Hrly = model_hourly.fit(X_train_hourly, y_train_hourly, epochs=epochs, batch_size=batch_size,
                    validation_data=(X_val_hourly, y_val_hourly), callbacks=[early_stopping])
LSTM_Daily = model_daily.fit(X_train_daily, y_train_daily, epochs=epochs, batch_size=batch_size,
                    validation_data=(X_val_daily, y_val_daily), callbacks=[early_stopping])
LSTM_Monthly = model_monthly.fit(X_train_monthly, y_train_monthly, epochs=epochs, batch_size=batch_size,
                    validation_data=(X_val_monthly, y_val_monthly), callbacks=[early_stopping])
```

Figure 5.20. Fit the Model Sample Code

Figure 5.20 shows, model fitting sample code, when fit the input to the model we used epochs, batch sizes patience and other hyperparameters.

Table 5.7. Hyperparameters Chosen for Optimal

No	***Patience***	***Dropout***	***Epochs***	***Batch Size***	***Layer Unit***	***Dense Layer***	***Activation***
1	5	0.2	20	64	100	7	sigmoid
2	5	0.2	30	64	100	7	sigmoid
3	5	0.25	30	64	100	7	sigmoid
4	5	0.25	30	64	100	1	sigmoid
5	5	0.2	30	64	100	1	sigmoid
5	5	0.2	30	32	100	1	sigmoid
6	5	0.2	30	64	100	52	swish
7	5	0.2	30	64	100	52	tanh
7	10	0.2	40	64	100	7	tanh
8	10	0.2	50	64	100	7	tanh
9	5	0.2	20	64	100	1	swish
10	5	0.2	20	64	100	7	swish
11	5	0.2	30	64	100	1	tanh
12	5	0.2	30	64	100	7	tanh

In the above Table 5.7, we put hyperparameters used for optimality, we try different hyperparameters as we see in the table not only this, we try more times changing with the hyperparameter. Hyperparameters have great influence for our model performance.

5.5.7. Activation Function

Activation function is a mathematical function applied to the output of a neuron (or a layer of neurons) to introduce non-linearities into the network. Activation functions play a crucial role in determining the output of a neuron and enabling the neural network to learn complex relationships and make non-linear predictions. In this study we applied the following activation functions.

- ***Tanh (Hyperbolic Tangent):*** The tanh function squashes the input values between -1 and 1. It is symmetric around the origin, which means it can produce both positive and negative outputs. Tanh can be effective for capturing complex patterns and non-linear relationships in the data.
- ***Sigmoid:*** The sigmoid function maps the input values to a range between 0 and 1. It is commonly used for binary classification problems where the output needs to be interpreted as a probability. Sigmoid functions can saturate when the input is extremely positive or negative, which can lead to vanishing gradients.
- ***Swish:*** Swish is a self-gating activation function that was proposed as an alternative to ReLU. Swish decrease the vanishing gradient problem.

5.5.8. Evaluation Criteria

Our Proposed model evaluation techniques briefly discussed in chapter four, here we applied Mean Absolute Error (MAE, Mean Square Error (MSE, Root Mean Square Error, and R-squared or R^2.

Figure 5.21 below shows, Proposed LSTM model for Crime type prediction, so as depicted in the figure it is sample code using metrics like MSE, MAE and RMSE with different granularities including hourly, daily and monthly.

```python
# Crime Prediction Evaluation Metrics Code
# For hourly prediction
y_pred_hourly = model_hourly.predict(X_test_hourly)
mse_hourly = mean_squared_error(y_test_hourly, y_pred_hourly)
mae_hourly = mean_absolute_error(y_test_hourly, y_pred_hourly)
rmse_hourly = np.sqrt(mse_hourly)

print(f'Hourly MSE: {mse_hourly}')
print(f'Hourly MAE: {mae_hourly}')
print(f'Hourly RMSE: {rmse_hourly}')
# For daily prediction
y_pred_daily = model_daily.predict(X_test_daily)
mse_daily = mean_squared_error(y_test_daily, y_pred_daily)
mae_daily = mean_absolute_error(y_test_daily, y_pred_daily)
rmse_daily = np.sqrt(mse_daily)

print(f'Daily MSE: {mse_daily}')
print(f'Daily MAE: {mae_daily}')
print(f'Daily RMSE: {rmse_daily}')
# For monthly prediction
y_pred_monthly = model_monthly.predict(X_test_monthly)
mse_monthly = mean_squared_error(y_test_monthly, y_pred_monthly)
mae_monthly = mean_absolute_error(y_test_monthly, y_pred_monthly)
rmse_monthly = np.sqrt(mse_monthly)

print(f'Monthly MSE: {mse_monthly}')
print(f'Monthly MAE: {mae_monthly}')
print(f'Monthly RMSE: {rmse_monthly}')
```

Figure 5.21. Evaluate the Model Sample Code

CHAPTER SIX

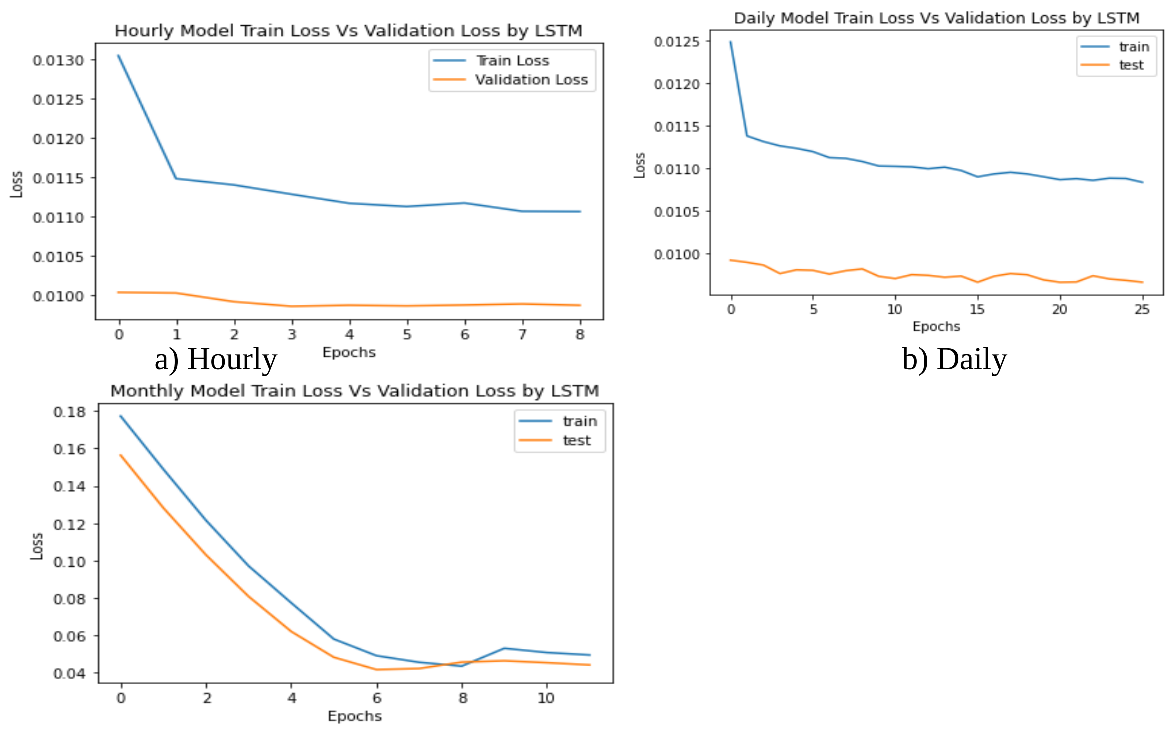
6. RESULTS AND DISCUSSIONS

This chapter, the outcomes of the model are showcased and analyzed in relation to various evaluation metrics, including the comparison of training loss and validation loss, the alignment between predictions and actual values, as well as the assessment using Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2).

6.1. Train Loss Versus Validation Loss Crime Type Results

6.1.1. Model Train vs Validation Loss of LSTM

Here we plot the LSTM testing loss versus validation loss results



a) Hourly b) Daily

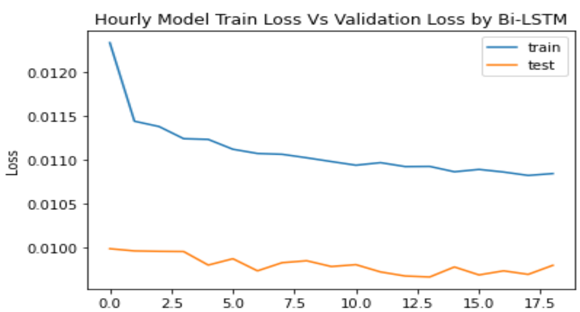
c) Monthly

Figure 6.1. LSTM Model Train vs Validation Loss for Crime Type Prediction

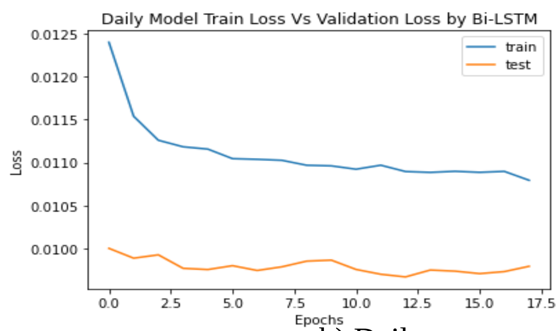
As shown in the above Figure 6.1, x-axis represents loss and y-axis represents number of epochs when training the model, the LSTM model achieved a testing loss of 0.01268 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0127 from the true values at an hourly level. At a daily granularity, the model's testing loss was slightly lower at 0.0126, suggesting a slightly better performance compared to the hourly granularity. However, at a monthly granularity, the model exhibited a higher testing loss of 0.04936, indicating a larger average deviation of approximately 0.0494 from the true values. These results suggest that the model's forecasts are better accurate and closest to the actual values when considering data at the finest granularity (hourly), while the performance decreases as the granularity becomes coarser (daily and monthly). Overall, the model demonstrates its highest precision when operating at the hourly level, making it more appropriate for taking short-term patterns and fluctuations in the dataset.

6.1.2. Model Train vs Validation Loss of BiLSTM

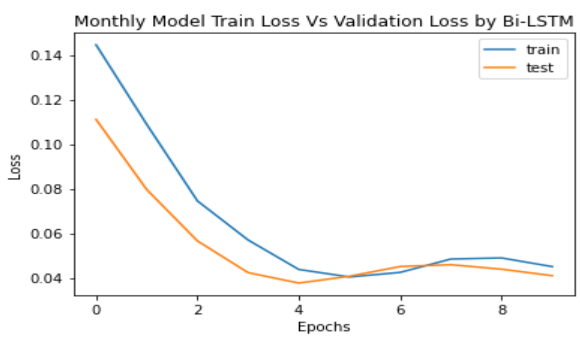
Here we plot the Bi-LSTM testing loss versus validation loss results



a) Hourly



b) Daily



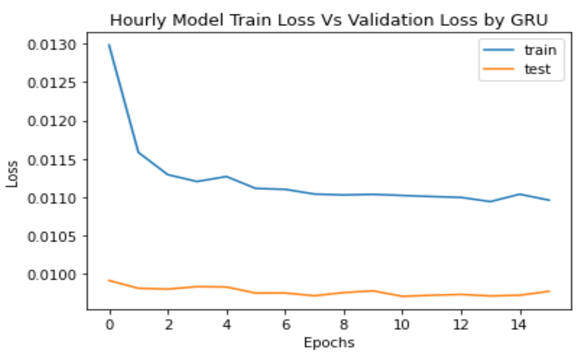
c) Monthly

Figure 6.2. Bi-LSTM Model Train vs Validation Loss for Crime Type Prediction

As shown in the above Figure 6.2, x-axis represents loss and y-axis represents number of epochs when training the model, the Bi-LSTM model achieved a testing loss of 0.01265 at an hourly granularity. This suggests that, on average, the model's forecasts differ by roughly 0.0127 from the actual values when examined on an hourly basis. At a daily granularity, the model's testing loss was slightly lower at 0.0125, indicating a slightly better performance compared to the hourly granularity. This indicates that, when looking at data on a daily basis, the model's forecasts are more precise and exhibit greater proximity to the actual values. Additionally, at a monthly granularity, the model exhibited a testing loss of 0.04665, which is slightly higher than the daily loss. This indicates that the model's predictions at a monthly level have a larger average deviation of roughly 0.0467 from the true values. Overall, the Bi-LSTM model demonstrates its highest precision when operating at the daily level, making it suitable for capturing patterns and trends on a daily basis. It also performs reasonably well at both the hourly and monthly granularities, albeit with slightly higher testing losses.

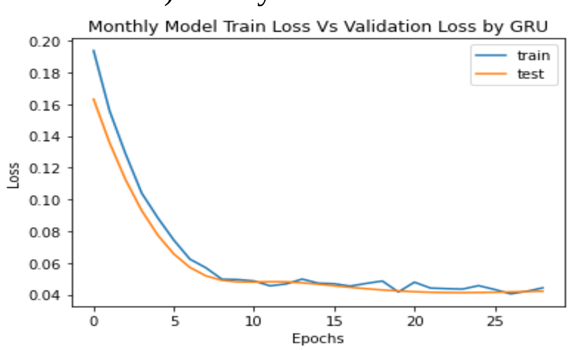
6.1.3. Model Train vs Validation Loss of GRU

Here we plot the GRU testing loss versus validation loss results.



a) Hourly

b) Daily



c) Monthly

Figure 6.3. GRU Model Train vs Validation Loss for Crime Type Prediction

As shown in the above Figure 6.3, x-axis represents loss and y-axis represents number of epochs when training the model, the GRU model achieved a testing loss of 0.01272 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0127 from the true values at an hourly level. At a daily granularity, the model's testing loss was slightly higher at 0.0128, suggesting a slightly lower performance compared to the hourly granularity.

This implies that the model's predictions have a larger average deviation of around 0.0128 from the true values at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.05019, which is slightly higher than the daily loss. This indicates that the model's predictions at a monthly level have a larger average deviation of approximately 0.0502 from the true values. Overall, the GRU model demonstrates its highest precision when operating at the hourly level, making it more appropriate for taking short-term patterns and fluctuations in the dataset. However, it still performs reasonably well at the daily and monthly granularities, although with slightly higher testing losses.

6.1.4. Model Train vs Validation Loss of Bi-GRU

Here we plot the Bi-GRU testing loss versus validation loss results.

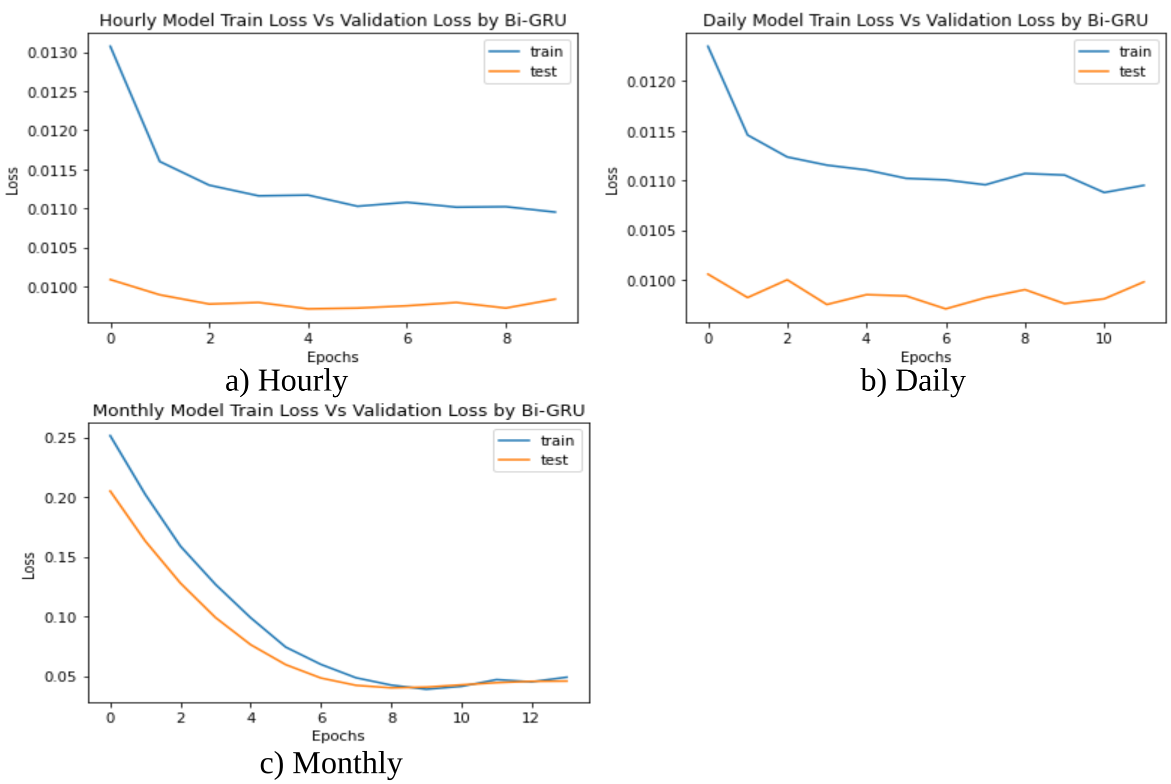


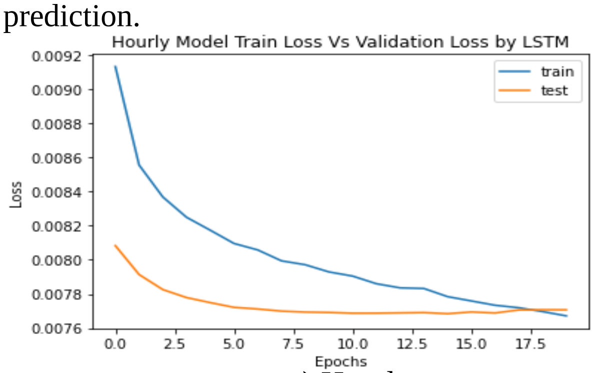
Figure 6.4. Bi-GRU Model Train vs Validation Loss for Crime Type Prediction

As shown in the above Figure 6.4, x-axis represents loss and y-axis represents number of epochs when training the model, the Bi-GRU model achieved a testing loss of 0.01260 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0126 from the true values at an hourly level. At a daily granularity, the model's testing loss was slightly higher at 0.01263, suggesting a similar performance compared to the hourly granularity. This implies that the model's predictions have a similar average deviation of around 0.0126 from the true values at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.04679, which is slightly higher than the daily loss. This indicates that the model's predictions at a monthly level have a larger average deviation of approximately 0.0468 from the true values. Overall, the Bi-GRU model demonstrates its consistent performance across different granularities. It shows its effectiveness in capturing both short-term and long-term patterns in the data.

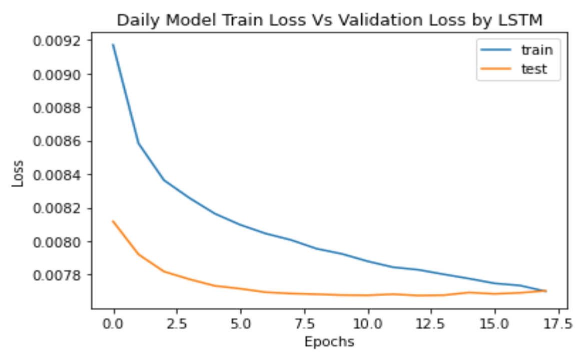
6.2. Train Loss Versus Validation Loss Crime Location Results

6.2.1. Model Train vs Validation Loss of LSTM

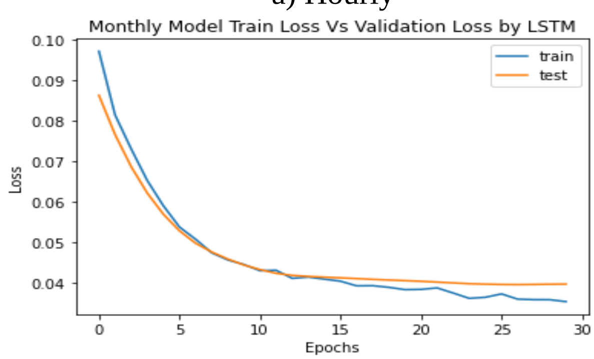
Here we plot the LSTM testing loss versus validation loss results of Crime location prediction.



a) Hourly



b) Daily



c) Monthly

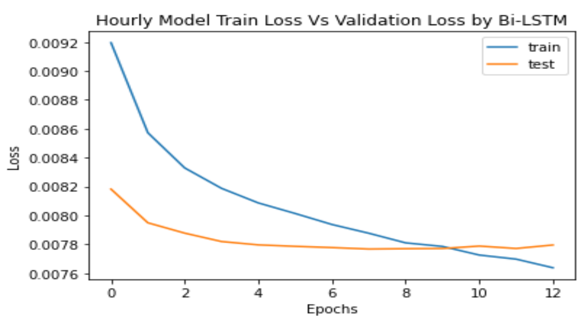
Figure 6.5. LSTM Model Train vs Validation Loss for Crime Location Prediction

As shown in the above Figure 6.5, x-axis represents loss and y-axis represents number of epochs when training the model, the LSTM model achieved a testing loss of 0.01084 at an hourly granularity, indicating that, on average, its predictions deviated by approximately

0.0108 from the true Crime locations at an hourly level. At a daily granularity, the model's testing loss was slightly higher at 0.0109, suggesting a similar performance compared to the hourly granularity. This implies that the model's predictions have a similar average deviation of around 0.0109 from the true Crime locations at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.06168, which is significantly higher than the hourly and daily losses. This indicates that the model's predictions at a monthly level have a larger average deviation of approximately 0.0617 from the true Crime locations. Overall, the LSTM Crime Location prediction model demonstrates its highest precision when operating at the hourly and daily levels, making it more suitable for capturing short-term patterns and fluctuations in Crime locations. However, it exhibits a higher testing loss at the monthly granularity, suggesting that capturing long-term trends in Crime location prediction might be more challenging.

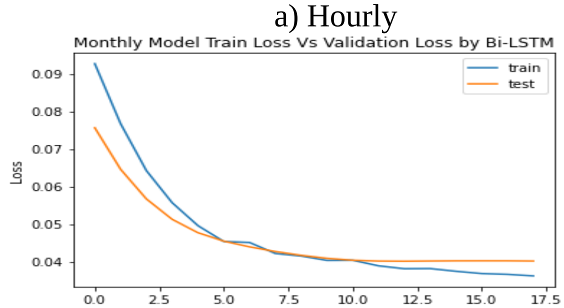
6.2.2. Model Train vs Validation Loss of BiLSTM

Here we plot the Bi-LSTM testing loss versus validation loss results of Crime location prediction.



a) Hourly

b) Daily



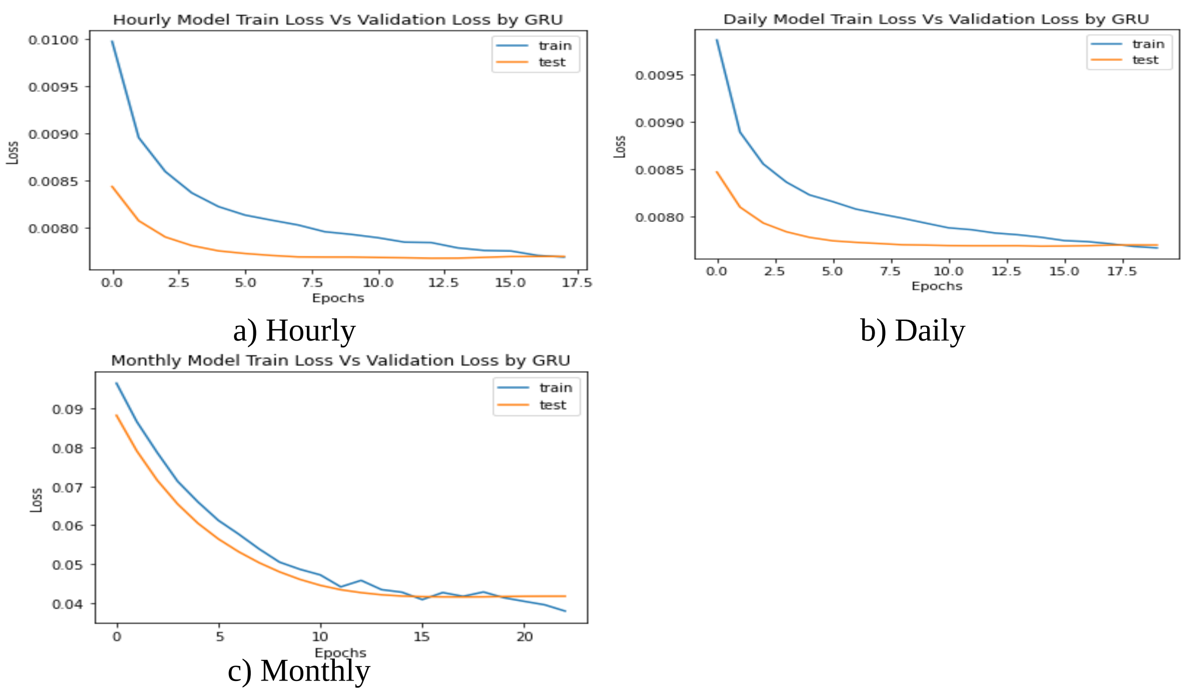
c) Monthly

Figure 6.6. BiLSTM Model Train vs Validation Loss for Crime Location Prediction

As shown in the above Figure 6.6, x-axis represents loss and y-axis represents number of epochs when training the model, the Bi-LSTM model achieved a testing loss of 0.01076 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0108 from the true Crime locations at an hourly level. At a daily granularity, the model's testing loss was slightly higher at 0.0110, suggesting a slightly lower performance compared to the hourly granularity. This implies that the model's predictions have a larger average deviation of around 0.0110 from the true Crime locations at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.05759, which is higher than the hourly and daily losses. This indicates that the model's predictions at a monthly level have a larger average deviation of approximately 0.0576 from the true Crime locations. Overall, the Bi-LSTM Crime Location prediction model demonstrates its highest precision when operating at the hourly granularity, making it more suitable for capturing short-term patterns and fluctuations in Crime locations. Although it performs reasonably well at the daily and monthly granularities, it shows slightly higher testing losses, suggesting that capturing long-term trends in Crime location prediction may be more challenging.

6.2.3. Model Train vs Validation Loss of GRU

Here we plot the GRU testing loss versus validation loss results of Crime location prediction.



a) Hourly b) Daily

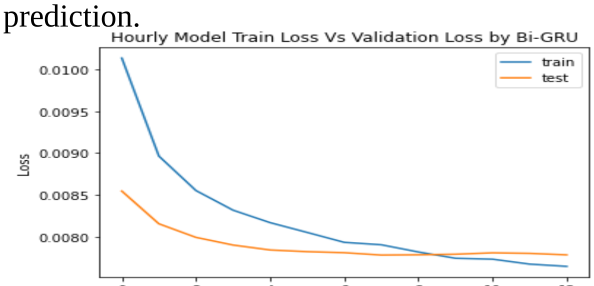
c) Monthly

Figure 6.7. GRU Model Train vs Validation Loss for Crime Location Prediction

As shown in the above Figure 6.7, x-axis represents loss and y-axis represents number of epochs when training the model, the GRU model achieved a testing loss of 0.01059 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0106 from the true Crime locations at an hourly level. At a daily granularity, the model's testing loss was slightly lower at 0.0105, suggesting a slightly better performance compared to the hourly granularity. This implies that the model's predictions have a smaller average deviation of around 0.0105 from the true Crime locations at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.05819, which is slightly higher than the hourly and daily losses. This indicates that the model's predictions at a monthly level have a larger average deviation of approximately 0.0582 from the true Crime locations. Overall, the GRU Crime Location prediction model demonstrates its highest precision when operating at the daily granularity, making it more suitable for capturing short-term patterns and fluctuations in Crime locations. It also performs well at the hourly granularity, with a slightly higher testing loss.

6.2.4. Model Train vs Validation Loss of Bi-GRU

Here we plot the Bi-GRU testing loss versus validation loss results of Crime location prediction.



a) Hourly

b) Daily

c) Monthly

Figure 6.8. Bi-GRU Model Train vs Validation Loss for Crime Location Prediction

As shown in the above Figure 6.8, x-axis represents loss and y-axis represents number of epochs when training the model, the Bi-GRU model achieved a testing loss of 0.01053 at an hourly granularity, indicating that, on average, its predictions deviated by approximately 0.0105 from the true Crime locations at an hourly level. At a daily granularity, the model's testing loss was slightly higher at 0.0106, suggesting a slightly lower performance compared to the hourly granularity. This implies that the model's predictions have a larger average deviation of around 0.0106 from the true Crime locations at a daily level. Furthermore, at a monthly granularity, the model exhibited a testing loss of 0.05128, which is significantly lower than the hourly and daily losses. This indicates that the model's predictions at a monthly level have a smaller average deviation of approximately 0.0513 from the true Crime locations. Overall, the Bi-GRU Crime Location prediction model demonstrates its highest precision when operating at the monthly granularity, making it more suitable for capturing long-term trends and patterns in Crime locations. It also performs well at the hourly and daily granularities, with slightly higher testing losses. The bidirectional nature of the Bi-GRU permits it to take dependencies in past and future contexts, enhancing its ability to understand temporal patterns and improve prediction accuracy.

6.3. Comparison of Prediction vs Test Data Crime Type Results

6.3.1. True Future vs Prediction of LSTM

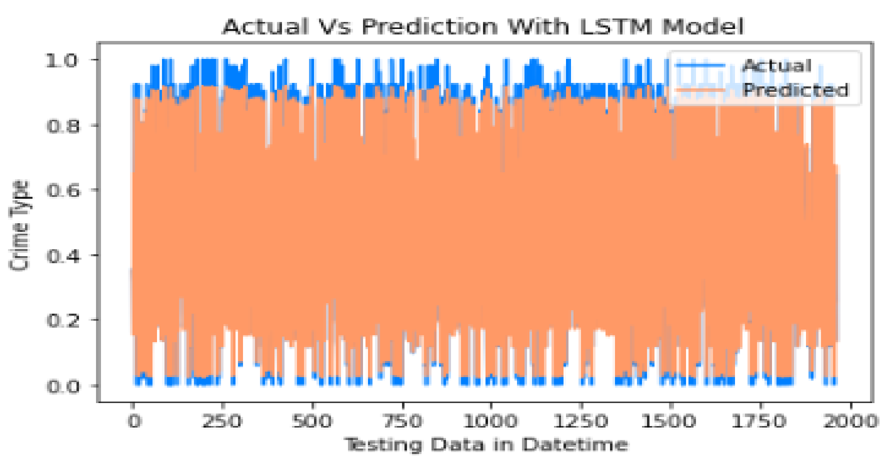


Figure 6.9. Crime Type True Future vs Prediction using LSTM

As shown in the above Figure 6.9, we plotted the actual value and predicted value of Crime type prediction result of LSTM model, the result shows, LSTM model can predict good.

6.3.2. True Future vs Prediction of BiLSTM

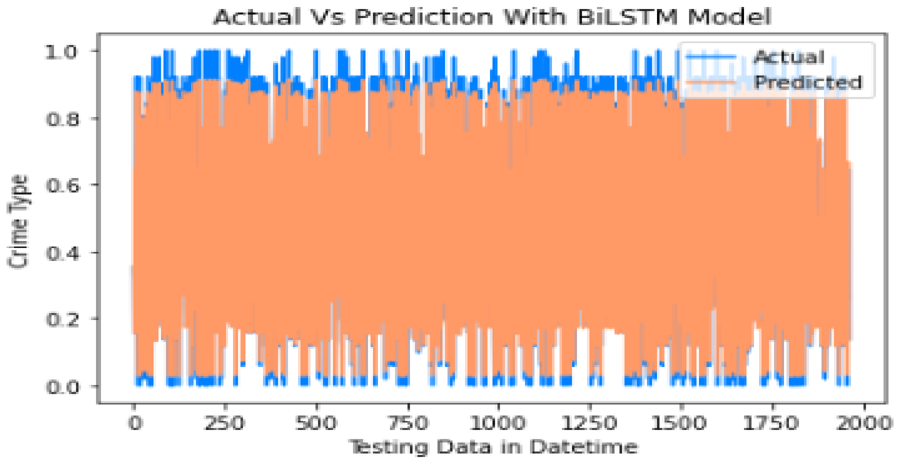


Figure 6.10. Crime Type True Future vs Prediction using BiLSTM

As shown in the above Figure 6.10, we plotted the actual value and predicted value of Crime type prediction result using Bi-LSTM model, the model result shows, Bi-LSTM model predicts better for our Crime dataset.

6.3.3. True Future vs Prediction of GRU

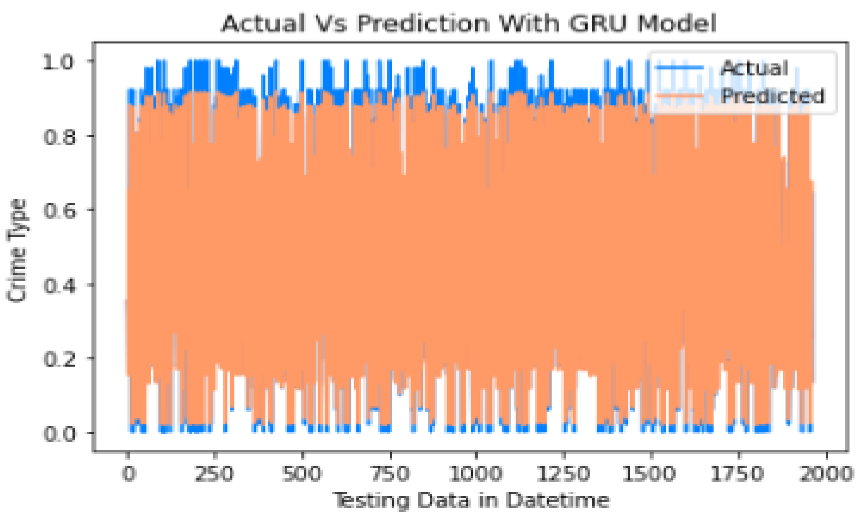


Figure 6.11. Crime Type True Future vs Prediction using GRU

As shown in the above Figure 6.11, we plotted the actual value and predicted value of Crime type prediction result of GRU model, the result shows, GRU model can predict also good but the prediction result little less than LSTM and Bi-LSTM.

6.3.4. True Future vs Prediction of Bi-GRU

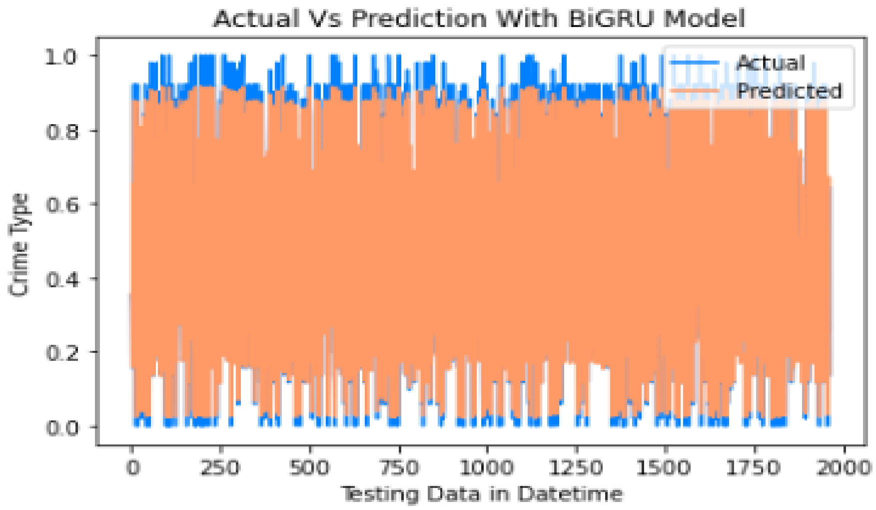


Figure 6.12. Crime Type True Future vs Prediction using Bi-GRU

As shown in the above Figure 6.12, we plotted the actual value and predicted value of Crime type prediction result of Bi-GRU model, the result shows, Bi-GRU model can predict also good but the prediction result little less than LSTM and Bi-LSTM.

6.4. Comparison of Prediction vs Test Data Crime Location Results

6.4.1. True Future vs Prediction of LSTM

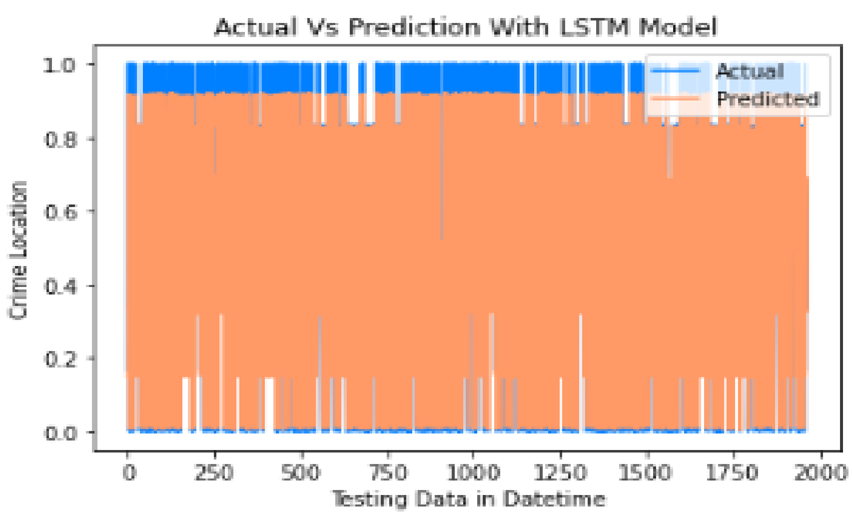


Figure 6.13. Crime Location Actual Future vs Prediction using LSTM

As shown in the above Figure 6.13, we plotted the actual value and predicted value of Crime location prediction result of LSTM model, the result shows, LSTM model predict better.

6.4.2. True Future vs Prediction of BiLSTM

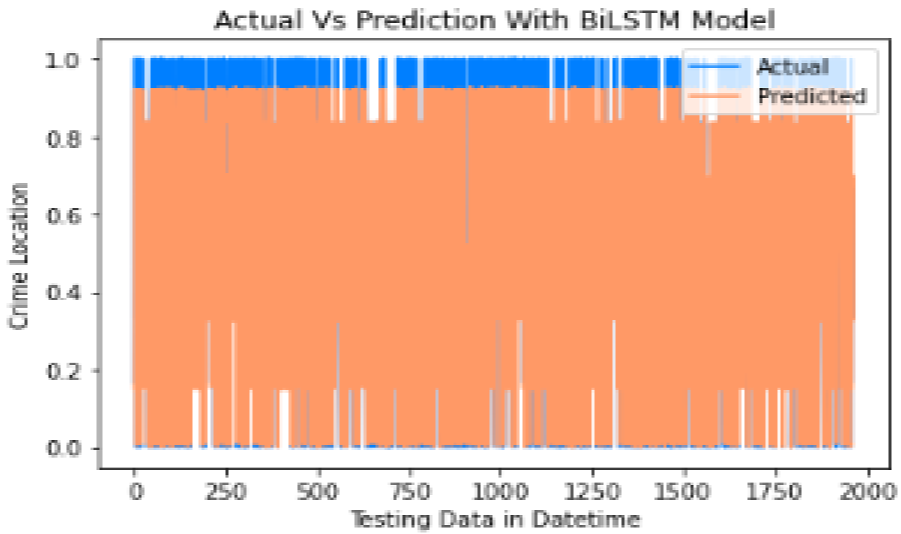


Figure 6.14. Crime Location True Future vs Prediction using BiLSTM

As shown in the above Figure 6.14, we plotted the actual value and predicted value of Crime location prediction result of Bi-LSTM model, the result shows, Bi-LSTM model predict better.

6.4.3. Actual Future vs Prediction of GRU

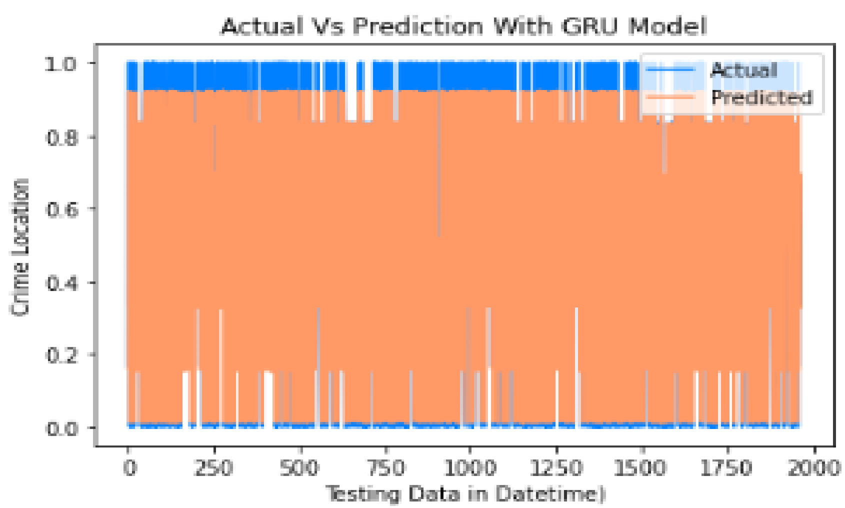


Figure 6.15. Crime Location Actual Future vs Prediction using GRU

As shown in the above Figure 6.15, we plotted the actual value and predicted value of Crime location prediction result of GRU model, the result shows, GRU model predict better.

6.4.4. Actual Future vs Prediction of Bi-GRU

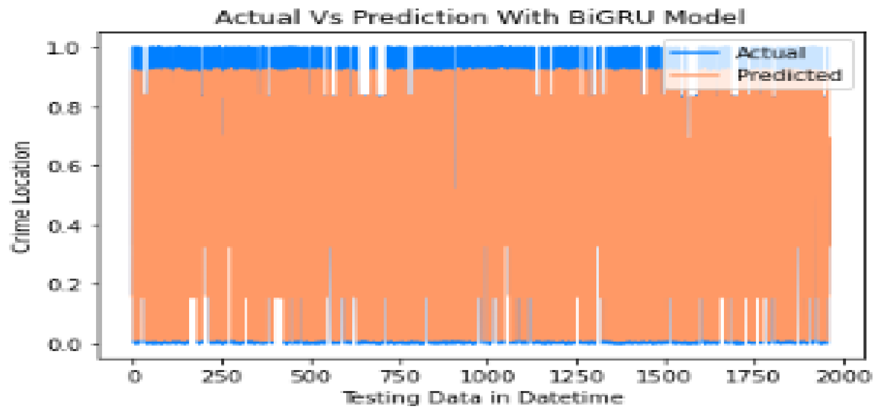


Figure 6.16. Crime Location Actual Future vs Prediction using Bi-GRU

As shown in the above Figure 6.16, we plotted the actual value and predicted value of Crime location prediction result of Bi-GRU model, the result shows, Bi-GRU model predict better.

6.5. Comparison of Actual vs Future Forecasting

6.5.1. True Future vs Future Forecast of LSTM

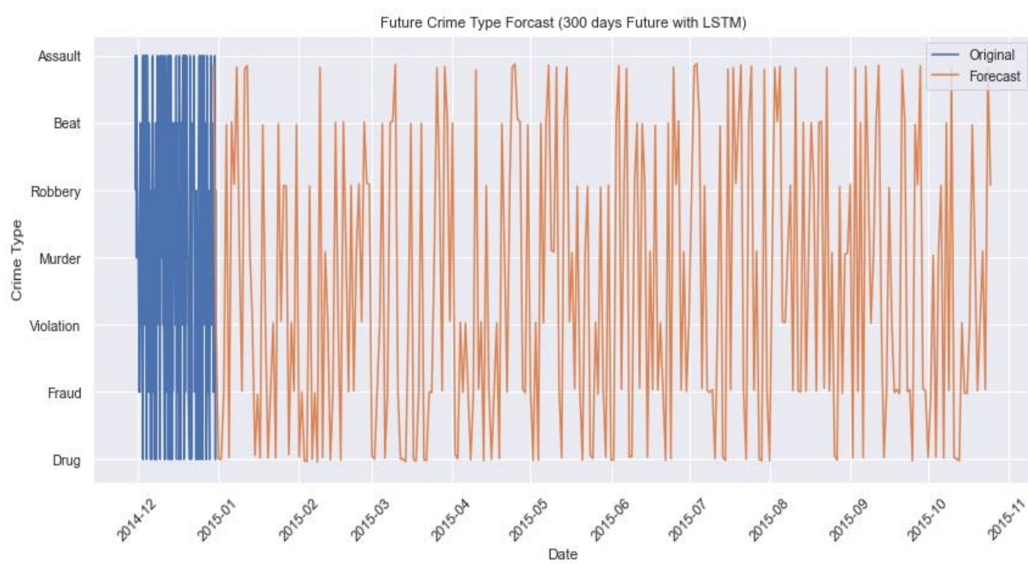


Figure 6.17. Forecasted Crime Type by LSTM

In figure 6.17, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime type. the forecasted Crime types exhibit good precision, the striking proximity between the plotted line and the line representing the actual values. The forecasted line gracefully intertwines with the actual line, showcasing a high degree of precision in the prediction.

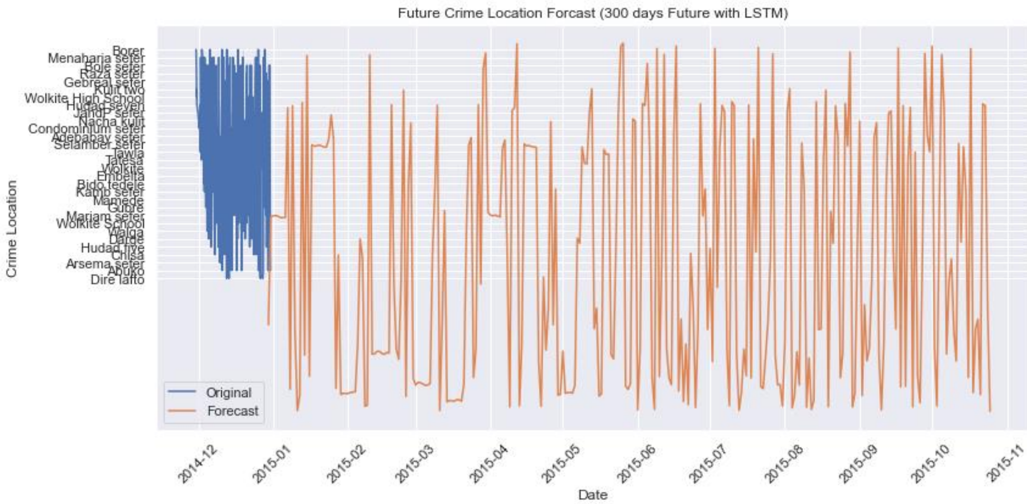


Figure 6.18. Forecasted Crime Location by LSTM

In figure 6.18, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime location. the forecasted Crime location plot line also close to the actual values, but here we plot starting from 2014-12 and continuously consistent in different months.

6.5.2. True Future vs Future Forecast of Bi-LSTM

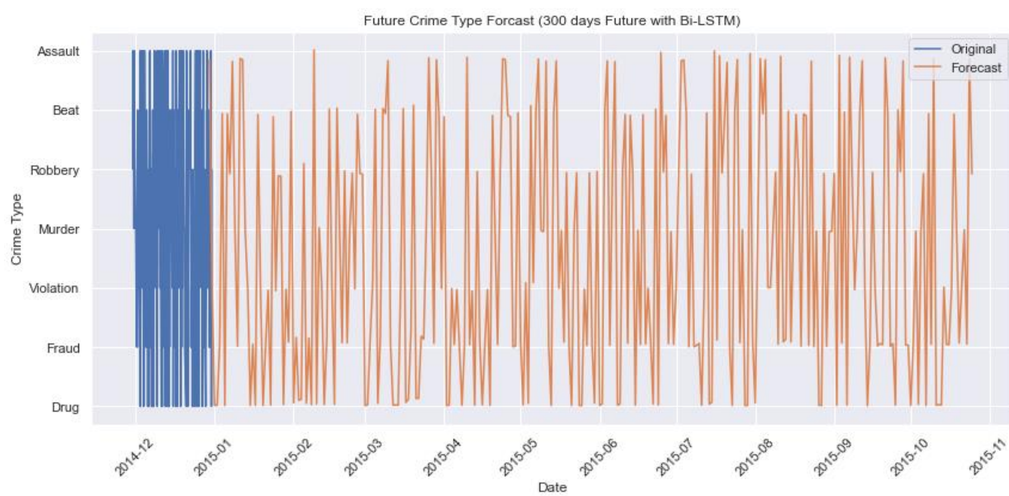


Figure 6.19. Forecasted Crime Type by Bi-LSTM

In figure 6.19, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime type. the predicted Crime types demonstrate a high level of accuracy, as indicated by the close alignment between the plotted line and the line representing the actual values. The predicted line smoothly follows the actual line, indicating a precise prediction.

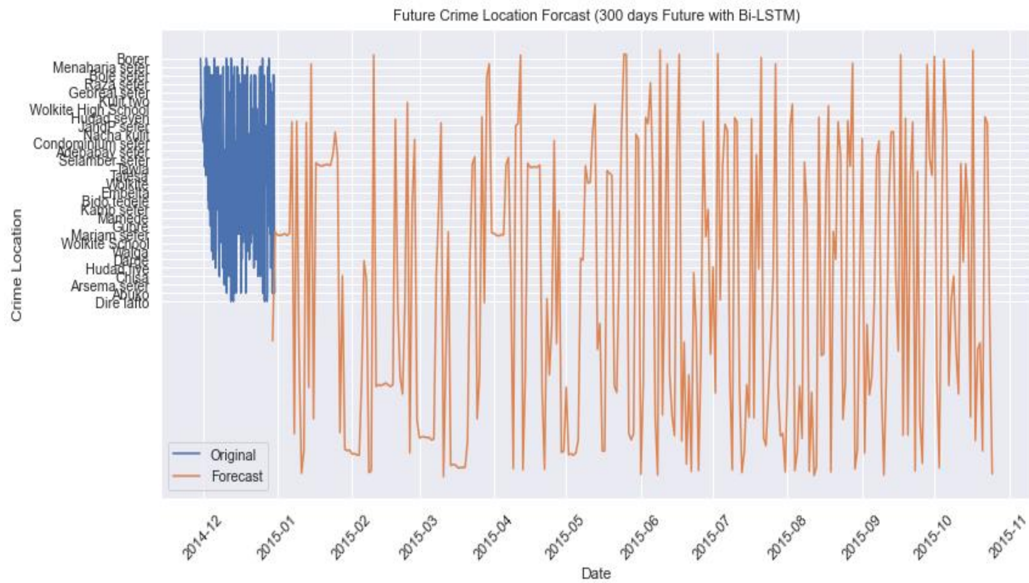


Figure 6.20. Forecasted Crime Location by Bi-LSTM

In figure 6.20, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime location. The forecasted Crime location plot line also closely matches the actual values, indicating a good level of accuracy in predicting the Crime locations, but here we plot starting from 2014-12 the original data and the forecasted values consistently varied frequency in different months.

6.5.3. True Future vs Future Forecast of GRU

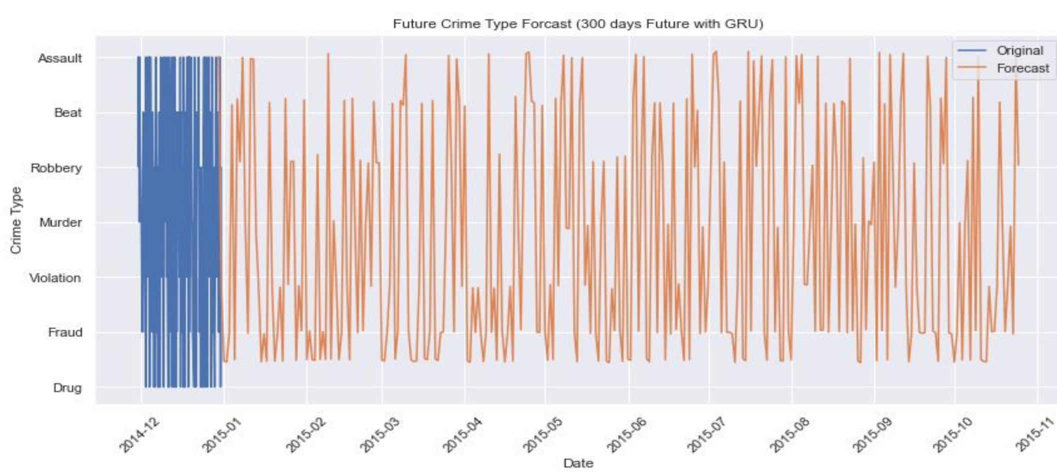


Figure 6.21. Forecasted Crime Type by GRU

In figure 6.21, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime type. The forecasted Crime types exhibit a remarkable level of precision, with the plotted line closely resembling the line representing the actual values. The predicted line seamlessly intertwines with the actual line, demonstrating a significant degree of precision in the prediction.

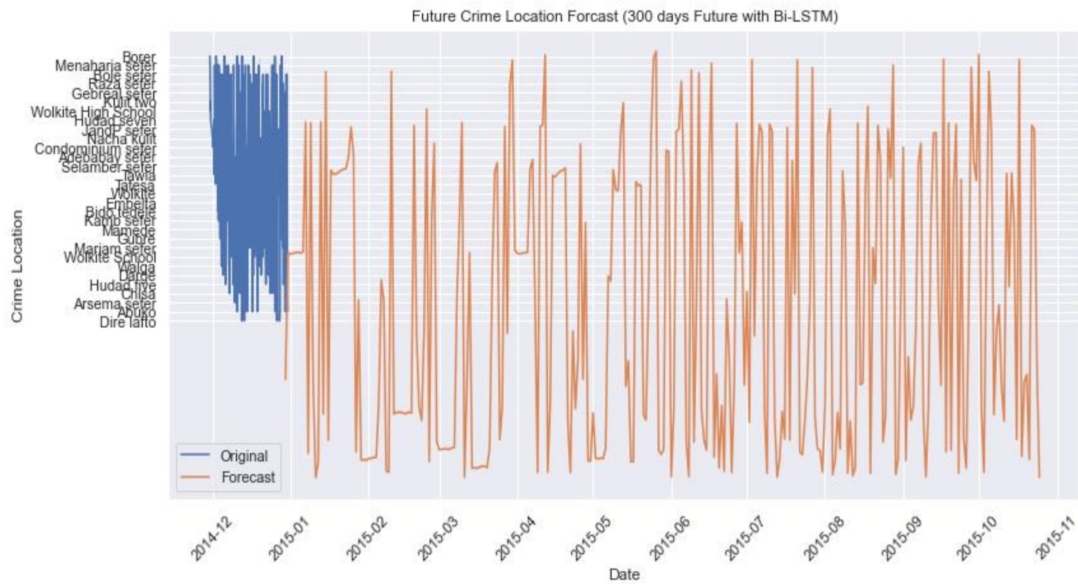


Figure 6.22. Forecasted Crime Location by GRU

In figure 6.22, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime location. The plot of the forecasted Crime locations also shows a close resemblance to the actual values, indicating good in predicting the Crime locations. But here we plot starting from 2014-12 the original data and the forecasted values consistently varied frequency in different months.

In figure 6.23, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime type. the forecasted Crime types exhibit good precision, the striking proximity between the plotted line and the line representing the actual values. The forecasted line gracefully intertwines with the actual line, showcasing a high degree of precision in the prediction.

6.5.4. True Future vs Future Forecast of Bi-GRU

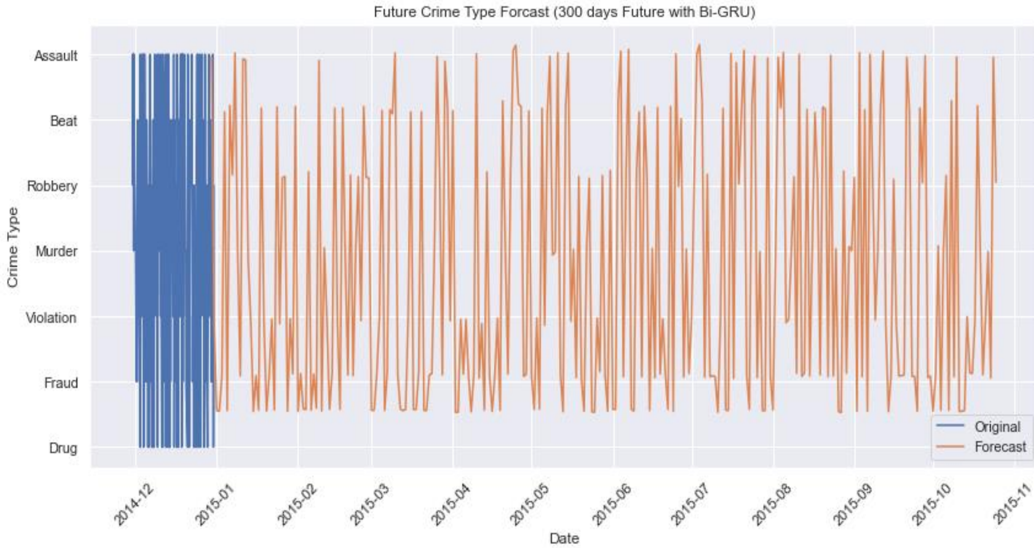


Figure 6.23. Forecasted Crime Type by Bi-GRU

The following figure shows forecasted crime location by bidirectional gated recurrent unit.

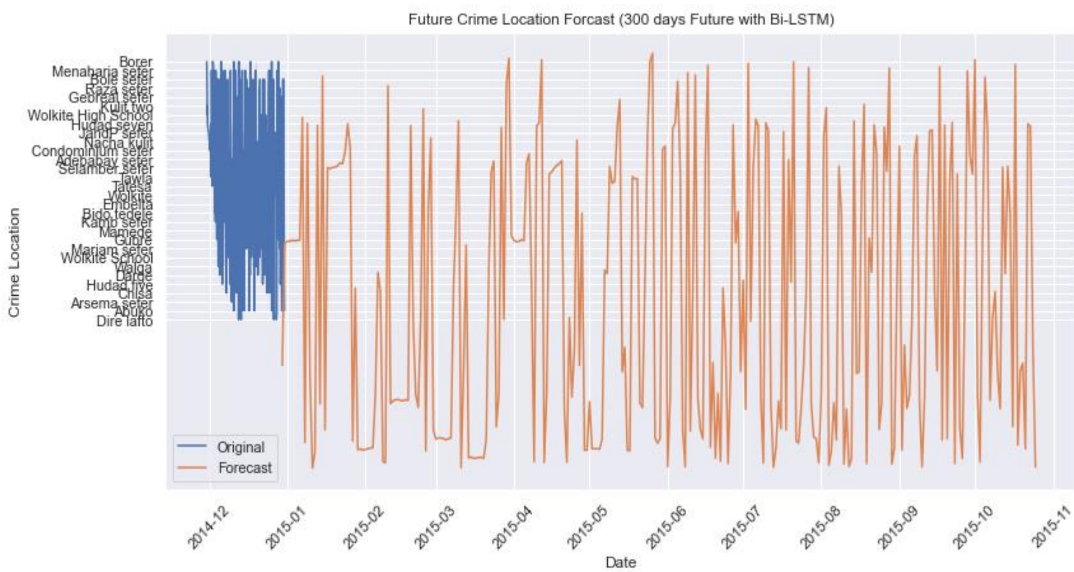


Figure 6.24. Forecasted Crime Location by Bi-GRU

In figure 6.24, x-axis represents 300 days future forecasted with a year of months and y-axis represents Crime location. The forecasted Crime location plot line also close to the actual values.

6.6. Discussions

The aim of this study is time series crime prediction using recurrent neural network (RNN) models including LSTM, BI-LSTM, GRU, and BI-GRU, the study can enable proactive crime prevention, optimize resource allocation for law enforcement agencies, enhance public safety through targeted interventions, aid in tactical deployment during critical situations, inform policy planning, and raise public awareness about crime patterns and risks. By leveraging RNN models, this research can provide valuable insights and tools to improve the effectiveness and efficiency of crime management strategies. We used nine thousand eight hundred twenty (9820) criminal records. Hyper parameters are determining the performance of the models. Such hyper parameters used for our study are: batch-size, number of epochs, activation function, optimizer, dropout rate, Early Stopping and loss function. Since all Hyper parameters help to seek enhanced the model performance score. In the existing work, some papers did not use RNN models and did not consider time series prediction. They focused on normal prediction without taking into account the temporal aspect. However, time series prediction is crucial in understanding the occurrence of crime events over time. There are related papers that incorporate time series data, but they fail to compare predictions at three granularities: hourly, daily, and monthly. Additionally, these papers do not analyze the forecasted values to identify the highest occurrence crime types in different areas or determine the locations that are prone to crime. In our study, we addressed the limitations mentioned in the existing work. We utilized RNN models for time series prediction, specifically focusing on crime events. We compared predictions at different granularities, including hourly, daily, and monthly, to gain a comprehensive understanding of crime patterns over time. Additionally, we analyzed the forecasted values to identify the highest occurrence crime types in various areas and determine locations that are more susceptible to crime. Our models result compared through different Evaluation techniques like in the above sub-section 6.1, we compared model training versus testing loss with three granularities (hour, day and month) for Crime type prediction and for Crime location prediction also illustrated in sub-section 6.2. In sub-section 6.3, we show our models prediction versus truth values result with plotting figure for Crime type prediction and for Crime location depicted in sub section 6.4. In sub-section 6.5, we compared the models future forecast versus truth values for Crime type and Crime location. In then here we have the model's comparison result using our main metrics like MSE, MAE, RMSE and R^2. These comparisons shown in Table 6.1, Table 6.2 and Table 6.3 below.

Table 6.1. Crime Type Prediction Summary

Crime Type Prediction				
Models Name	***Metrics***	***Hourly***	***Daily***	***Monthly***
LSTM	MSE	**0.0125**	0.0126	0.0468
	MAE	**0.0842**	0.0847	0.1645
	RMSE	**0.1120**	0.1122	0.2164
Bi-LSTM	MSE	0.0126	**0.0125**	**0.0466**
	MAE	0.0844	**0.0835**	**0.1541**
	RMSE	0.1124	**0.1117**	**0.2159**
GRU	MSE	0.0127	0.0128	0.0501
	MAE	0.0846	0.0847	0.1674
	RMSE	0.1128	0.1129	0.2240
Bi-GRU	MSE	0.0126	0.0126	0.0468
	MAE	0.0844	0.0844	0.1650
	RMSE	0.1122	0.1124	0.2163

As we showed in Table 6.1, the comparison of the results for Crime type prediction across the LSTM, Bi-LSTM, GRU, and Bi-GRU models reveals some interesting patterns. The MSE values varied across the models, with the Bi-LSTM, and Bi-GRU models achieving similar results in hourly granularity and LSTM and Bi-GRU perform similar result in both Daily and monthly. The GRU model exhibiting slightly higher values. The lowest MSE values were observed for the LSTM and Bi-LSTM models.

Similarly, the MAE values followed a similar pattern, with the Bi-LSTM model achieving the lowest values, closely followed by the LSTM and Bi-GRU models, while the GRU model displayed slightly higher MAE values.

The RMSE values aligned with the MSE and MAE comparisons, with the Bi-LSTM model attaining the lowest values. However, the GRU model demonstrated slightly higher RMSE values across all prediction granularities. In general, for hourly Crime type prediction LSTM is slightly better than others. For daily and monthly Crime type prediction Bi-LSTM is better than others. Next to LSTM and Bi-LSTM, Bi-GRU perform comparable results whereas GRU is slightly higher score of MSE, MAE and RMSE.

Table 6.2. Crime Location Prediction Summary

Crime Location Prediction				
Models Name	***Metrics***	***Hourly***	***Daily***	***Monthly***
LSTM	MSE	0.0108	0.0109	0.0617
	MAE	0.0526	0.0534	0.1524
	RMSE	0.1041	0.1044	0.2484
Bi-LSTM	MSE	0.0108	0.0110	0.0576
	MAE	0.0542	0.0534	0.1505
	RMSE	0.1038	0.1050	0.2399
GRU	MSE	0.0106	**0.0105**	0.0582
	MAE	**0.0524**	0.0524	0.1469
	RMSE	0.1029	**0.1025**	0.2412
Bi-GRU	MSE	**0.0105**	0.0106	**0.0513**
	MAE	0.0531	**0.0521**	**0.1405**
	RMSE	**0.1026**	0.1031	**0.2264**

As we showed in Table 6.2, in the comparison of Mean Squared Error (MSE) values for Crime location prediction, it is evident that different models and granularities yield varying results. The Bi-GRU model stands out as it achieved the lowest MSE values, with 0.0105 for hourly, 0.0106 for daily, and 0.0513 for monthly predictions. On the other hand, the LSTM and Bi-LSTM models demonstrated slightly higher MSE values.

Furthermore, the Mean Absolute Error (MAE) values followed a similar pattern to the MSE results. The Bi-GRU model performed the best, closely followed by the GRU model, while the LSTM and Bi-LSTM models exhibited slightly higher MAE values. The Root Mean Squared

Error (RMSE) values also aligned with the MSE and MAE comparisons. The Bi-GRU model outperformed the others, with the lowest RMSE values, followed by the GRU model.

The LSTM and Bi-LSTM models had slightly higher RMSE values across all granularities. Both the Bi-GRU and GRU models consistently demonstrated superior performance by achieving lower errors (MSE, MAE, RMSE) compared to the LSTM and Bi-LSTM models in Crime location prediction.

Table 6.3. RNN Models R_square Score Summary

Models Name	***Crime Type***	***Crime Location***
	R^2 Score	***R^2 Score***
LSTM	0.9899	0.9937
Bidirectional LSTM	0.9994	**0.9938**
GRU	0.9811	0.9934
Bidirectional GRU	**0.9995**	0.9937

As showed in Table 6.3, the models R^2 score is seated, R^2 scores indicate the goodness of fit of a model. Higher R^2 scores suggest that the model fits the data better and makes more accurate predictions. When we evaluate our models with R^2 metrics we are not group the data in to three granularities because the amount of data we have used is 9820, so in sufficient to get better score. So, we used total dataset to measure models by R^2 metrics and gained comparable results. For Crime type prediction Bi-GRU model performs slightly higher R^2 values than Bidirectional LSTM. Next to Bi-LSTM, LSTM performs higher R^2 values than GRU. For Crime location prediction Bidirectional LSTM gained higher R^2 values than others. Next to Bi-LSTM, Bidirectional GRU and LSTM is better. GRU is slightly less R^2 values than others for our dataset. In this summary, Bi-GRU and Bidirectional LSTM models appear to perform slightly better in Crime type prediction and For Crime location prediction Bi-LSTM performs slightly higher R^2 than others. Following to Bi-LSTM, LSTM and Bi-GRU have comparable R^2 values. Finally, GRU is slightly lowest R^2 value than others.

Table 6.4. Day of week Future Forecasted Crime Types

Day of week		***Forecasted Crime Type***						
		Assault (0)	***Beat (1)***	***Drug (2)***	***Fraud (3)***	***Murder (4)***	***Robbery (5)***	***Violation (6)***
Monday	count	7	8	1	5	3	6	11
Tuesday	count	3	5	9	4	1	9	10
Wednesday	count	1	3	3	2	7	14	11
Thursday	count	7	9	4	3	5	6	9
Friday	count	5	5	4	3	1	18	8
Saturday	count	6	6	4	3	5	11	6
Sunday	count	4	7	6	5	5	6	9

Table 6.4, presents what type of crime occur in which areas every day. Based on the table result, the crime type with the highest predicted occurrence on Monday is Violation with a count of 11 in the location Selamber, Tawla, Stadium, Adebabay, and Borer. On Tuesday, the crime type with the highest predicted occurrence is Violation with a count of 10 in Hudad seven, Tatesa, Darge, Stadium, Nacha Kulit and Alfa. Violation also has the highest predicted occurrence on Wednesday, with a count of 14 Nacha kulit, Adebabay, Darge, Tawla, Condominium, Tatesa, Dire Lafto, and Selamber. The crime type with the highest predicted occurrence on Thursday is Violation with a count of 9 in Gubre, Alfa, Mariam sefer, Darge, Nacha kulit and Stadium. Also, in this day Beat high with 9 counts in Stadium, Jayka, Gebreal, Selamber, Tatesa, Hudad five, Mazoriya, Gubre, Kera sefer. On Friday, Robbery has the highest predicted occurrence with a count of 18 in stadium, Kera, Darge, Nacha kulit, Selamber, Sebakuteba, Jayka, Arsema, Alfa, Mariam sefer, Alfa, Mariam sefer, Bole sefer, Tawla, Raza, and Menaharia. The crime type with the highest predicted occurrence on Saturday is Robbery with a count of 11 in Gunchre, Mazoriya, Jayka, Nacha kulit, Selamber, Kera sefer, Bole sefer and Tawla. On Sunday, Violation is again the crime type with the highest predicted occurrence, with a count of 9 in Stadium, Tawla, Selamber, Borer, Gibo bare, Kera sefer, Darge, Hudad five and Raza sefer. Overall, the data indicates that Violation and Robbery are the dominant crime types in terms of highest predicted occurrences on different days of the week.

Table 6.5. Day of week Future Forecasted Location

Day of Week	*Forecasted Location Value*	*Corresponding Location*	*Frequency*
Monday	46	Tawla	7
Tuesday	8	Bole sefer	4
Wednesday	39	Nacha kulit	7
Thursday	44	Stadium sefer	5
Friday	8	Bole sefer	5
Saturday	8	Bole sefer	5
Sunday	44	Stadium sefer	4

Table 6.5, presents which area is more prone to crime. Based on the table result, on Mondays, the forecasted location value is 46, indicating that it is expected to have a relatively higher occurrence of crime compared to other days of the week. The corresponding location for this forecasted value is Tawla, and the frequency of crime incidents in that location is 7. Tuesdays have a forecasted location value of 8, indicating a relatively lower expected occurrence of crime compared to other days. The corresponding location for this forecasted value is Bole sefer, and the frequency of crime incidents in that location is 4. Wednesdays have a forecasted location value of 39, suggesting a moderate level of expected crime occurrence. The corresponding location for this forecasted value is Nacha kulit, and the frequency of crime incidents in that location is 7. Thursdays have a forecasted location value of 44, indicating a relatively higher expected occurrence of crime. The corresponding location for this forecasted value is Stadium sefer, and the frequency of crime incidents in that location is 5. Fridays have a forecasted location value of 8, similar to Tuesdays, suggesting a relatively lower expected occurrence of crime. The corresponding location for this forecasted value is Bole sefer, and the frequency of crime incidents in that location is 5. Saturdays also have a forecasted location value of 8, indicating a relatively lower expected occurrence of crime. The corresponding location for this forecasted value is Bole sefer, and the frequency of crime incidents in that location is 5. Sundays have a forecasted location value of 44, similar to Thursdays, suggesting a relatively higher expected occurrence of crime. The corresponding location for this forecasted value is Stadium sefer, and the frequency of crime incidents in that location is 4.

Throughout the 10 months, Darge, Nacha kulit, Selamber, Stadium, Tawla, and Bole sefer have consistently emerged as the crime-prone locations. These areas have experienced higher crime rates compared to others. Over all fifteen locations prone to Crime based on Crime frequency, Nacha kulit (39.0) ranks first with 24 incidents, followed by Darge (13.0) with 23 incidents, Bole sefer (8.0) with 22 incidents, Tawla (46.0) also with 22 incidents, and Stadium (44.0) with 21 incidents. Selamber (43.0) has 18 reported incidents, while Tatesa (45.0) has 16 incidents. Gubre (21.0) and Sebakuteba (42.0) both have 11 reported incidents. Adebabay sefer (1.0) has 10 incidents, while Condominium (12.0) and Mazoriya (36.0) each have 9 incidents. Raza (40.0) and Alfa (2.0) both have 8 reported incidents. Lastly, Borer (9.0) rounds out the top fifteen crime-prone locations with 7 incidents.

Generally, the research questions related to predicting time series crime type, crime location, and identifying areas prone to crime in the future carry substantial significance and contribute to various aspects of crime management and public safety. By addressing these questions, the research aims to provide law enforcement agencies and policymakers with actionable insights and tools for proactive crime prevention and resource optimization. Accurate predictions of crime type and location enable authorities to allocate resources effectively, implement preventive measures, and adapt response strategies. Moreover, identifying high-risk areas in the future facilitates the development of evidence-based policies. Ultimately, this research contributes to enhancing public safety, optimizing resource allocation, and improving the overall effectiveness of crime management strategies.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. Conclusion

The aim of this study is to conduct Crime prediction analysis on time series data using Recurrent Neural Networks, which include LSTM, Bi-LSTM, GRU, and Bi-GRU. This study encompasses phases such as data collection, preprocessing, model training and testing, and ultimately, assessing and comparing the performance of the models.

This study utilized a total of 9,820 criminal records collected from the Wolkite City police department. After collecting the data, it underwent cleaning, scaling, and consideration of the past hour, day, and month. Additionally, the data was grouped based on hour, day, and month. However, a challenge arose during the grouping process due to insufficient data, as some crimes occurred simultaneously. Subsequently, the data was split into training and testing sets, and RNN-based models were trained. Finally, the models were evaluated using metrics such as MAE, MSE, and RMSE. For Crime type, hourly predictions, LSTM model has small MSE, MAE and RMSE, for daily and monthly Bi-LSTM models has slightly smaller MSE, MAE and RMSE compared to others, smaller MSE, MAE and RMSE indicating the model has good performance. So, for hourly Crime type LSTM is best score than others on the other hand for daily and monthly Bi-LSTM is best depending on our result. But, the GRU and Bi-GRU models have slightly higher MSE, MAE and RMSE values. For Crime location prediction, Bi-GRU model has the lowest MSE and RMSE, indicating better accuracy. And GRU has lowest MAE. For daily GRU model has lowest MSE and RMSE and Bi-GRU has lowest MAE. For monthly, Bi-GRU has lowest MSE, MAE and RMSE, so for Crime location prediction Bi-GRU and GRU models are slightly better than LSTM and Bi-LSTM. Also, we measure the performance of models using R_square. For Crime type prediction LSTM model performs 0.9899, Bi-LSTM model achieves 0.9994, GRU model performs 0.9811 and Bi-GRU model performs 0.9995 R^2 score. For Crime location prediction LSTM performs 0.9937, Bi-LSTM performs 0.9938, GRU gained 0.9934 and GRU models perform 0.9937 R^2 score. R_square result shows, both models perform comparable results for Crime type and Crime location prediction. Bi-GRU model slightly higher R-square score than Bi-LSTM for Crime type.

Following to Bi-LSTM, LSTM is good score and Crime location prediction Bi-LSTM models perform slightly higher R_square score and LSTM and Bi-GRU has the same result in for Crime dataset.

In general, the Bi-GRU and Bidirectional LSTM models showed slightly better performance in predicting crime types. For crime location prediction, the Bi-LSTM model had slightly higher R^2 values compared to the other models. Following the Bi-LSTM model, the LSTM and Bi-GRU models had comparable R^2 values. Finally, the GRU model exhibited slightly lower R2 values compared to the other models.

7.2. Recommendation

By deploying this method, law enforcement organizations can allocate their resources in a highly efficient and prompt manner. In terms of geographical areas, these methodologies aid in pinpointing the precise locations where the incident is likely to occur. We recommend that the Wolkite City Police Department allocate their resources strategically to address specific crime types in different areas of the city on each day of the week. Mondays: The focus should be on addressing Violation crimes in Selamber, Tawla, Stadium, Adebabay, and Borer. Tuesdays: Attention should be given to addressing Violation crimes in Hudad seven, Tatesa, Darge, Stadium, Nacha Kulit, and Alfa. Wednesdays: Priority should be given to addressing Violation crimes in Nacha kulit, Adebabay, Darge, Tawla, Condominium, Tatesa, Dire Lafto, and Selamber. Thursdays: Resources should be allocated to address Violation crimes in Gubre, Alfa, Mariam sefer, Darge, Nacha kulit, and Stadium. Additionally, beat crimes should be addressed in Stadium, Jayka, Gebreal, Selamber, Tatesa, Hudad five, Mazoriya, Gubre, and Kera sefer. Fridays: The focus should be on addressing Robbery crimes in stadium, Kera, Darge, Nacha kulit, Selamber, Sebakuteba, Jayka, Arsema, Alfa, Mariam sefer, Bole sefer, Tawla, Raza, and Menaharia. Saturdays: Priority should be given to addressing Robbery crimes in Gunchre, Mazoriya, Jayka, Nacha kulit, Selamber, Kera sefer, Bole sefer, and Tawla. Sundays: Resources should be allocated to address Violation crimes in Stadium, Tawla, Selamber, Borer, Gibo bare, Kera sefer, Darge, Hudad five, and Raza sefer. Tawla is identified as the area with the highest occurrence of crime. Therefore, it is strongly recommended for the Wolkite Police Department to prioritize their efforts and allocate resources specifically to address crime in the Tawla area.

It is evident that to delve deeper into the analysis, additional research is required.

- We used (the RNNs) methods to investigate more sophisticated methods for capturing temporal patterns. Recurrent Neural Networks (RNNs) like LSTM and GRU are good choices. But they need more data.so, we recommended extend the dataset with more Crime type and Crime location.
- We considered Crime date and time, Crime category and areas, for Crime type and location prediction for future work we recommended adding other factors like weather data, social media activity, and others. This is because weather conditions can have a significant impact on crime rates. Certain weather patterns, such as extreme temperatures, heavy rainfall, can influence people's behavior and increase the likelihood of certain types of crimes. Social media platforms generate vast amounts of data that reflect people's opinions, activities, and interactions. Analyzing social media activity can provide valuable insights into community sentiment, public events, gatherings, and potential conflicts that may contribute to criminal activities.

REFERENCES

[1] S. Walczak, "Predicting Crime and Other Uses of Neural Networks in Police Decision Making," Front Psychol, vol. 12, Oct. 2021, doi: 10.3389/fpsyg.2021.587943.

[2] Y. Du and N. Ding, "A Systematic Review of Multi-Scale Spatio-Temporal Crime Prediction Methods," ISPRS International Journal of Geo-Information, vol. 12, no. 6. MDPI, Jun. 01, 2023. doi: 10.3390/ijgi12060209.

[3] R. Rosés, C. Kadar, and N. Malleson, "A data-driven agent-based simulation to predict crime patterns in an urban environment," Comput Environ Urban Syst, vol. 89, Sep. 2021, doi: 10.1016/j.compenvurbsys.2021.101660.

[4] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," IEEE Access, vol. 9, pp. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.

[5] S. Sanjay Ghattadahalli, S. P. Naik, and K. Kumar Yadav, "Foresee Crime in Society Using Advanced ML Techniques," 2022.

[6] P. J. Brantingham, M. Valasik, and G. O. Mohler, "Does Predictive Policing Lead to Biased Arrests? Results From a Randomized Controlled Trial," Statistics and Public Policy, vol. 5, no. 1, pp. 1–6, Jan. 2018, doi: 10.1080/2330443X.2018.1438940.

[7] J. Fitterer, T. A. Nelson, and F. Nathoo, "Predictive crime mapping," Police Practice and Research, vol. 16, no. 2, pp. 121–135, Mar. 2015, doi: 10.1080/15614263.2014.972618.

[8] A. Stec and D. Klabjan, "Forecasting Crime with Deep Learning," Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.01486

[9] A. Najjar, S. Kaneko, and Y. Miyanaga, "Crime Mapping from Satellite Imagery via Deep Learning," Dec. 2018, [Online]. Available: http://arxiv.org/abs/1812.06764

[10] X. Zhang, L. Liu, L. Xiao, and J. Ji, "Comparison of machine learning algorithms for predicting crime hotspots," IEEE Access, vol. 8, pp. 181302–181310, 2020, doi: 10.1109/ACCESS.2020.3028420.

[11] X. Han, X. Hu, H. Wu, B. Shen, and J. Wu, “Risk Prediction of Theft Crimes in Urban Communities: An Integrated Model of LSTM and ST-GCN,” IEEE Access, vol. 8, pp. 217222–217230, 2020, doi: 10.1109/ACCESS.2020.3041924.

[12] V. Ceccato, T. Kahn, C. Herrmann, and A. Östlund, “Pandemic Restrictions and Spatiotemporal Crime Patterns in New York, São Paulo, and Stockholm,” J Contemp Crim Justice, vol. 38, no. 1, pp. 120–149, Feb. 2022, doi: 10.1177/10439862211038471.

[13] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM,” Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.09512

[14] N. Shah, N. Bhagat, and M. Shah, “Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention,” Visual Computing for Industry, Biomedicine, and Art, vol. 4, no. 1. Springer, Dec. 01, 2021. doi: 10.1186/s42492-021-00075-z.

[15] K. Jenga, C. Catal, and G. Kar, “Machine learning in crime prediction,” J Ambient Intell Humaniz Comput, vol. 14, no. 3, pp. 2887–2913, Mar. 2023, doi: 10.1007/s12652-023-04530-y.

[16] D. R. Bahiru Shifaw, “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,” 2019.

[17] “News: Security forces arrest several senior gov’t officials, youth, activists in Gurage zone following intensified crackdown on dissent - Addis Standard.” Accessed: Dec. 23, 2023. [Online]. Available: https://addisstandard.com/news-security-forces-arrest-several-senior-govt-officials-youth-activists-in-gurage-zone-following-intensified-crackdown-on-dissent/

[18] O. Fathi, “Time series forecasting using a hybrid ARIMA and LSTM model.”

[19] R. F. R. Forradellas, S. L. N. Alonso, M. L. Rodriguez, and J. Jorge-Vazquez, “Applied machine learning in social sciences: Neural networks and crime prediction,” Soc Sci, vol. 10, no. 1, pp. 1–20, Jan. 2021, doi: 10.3390/socsci10010004.

[20] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, "An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach," IEEE Access, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

[21] J. Wu, S. M. Abrar, N. Awasthi, E. Frias-Martinez, and V. Frias-Martinez, "Enhancing short-term crime prediction with human mobility flows and deep learning architectures," EPJ Data Sci, vol. 11, no. 1, Dec. 2022, doi: 10.1140/epjds/s13688-022-00366-2.

[22] F. T. Shohan, A. U. Akash, M. Ibrahim, and M. S. Alam, "Crime Prediction using Machine Learning with a Novel Crime Dataset," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.01551

[23] T. E. Meskela, Y. K. Afework, N. A. Ayele, M. W. Teferi, and T. B. Mengist, "Designing Time Series Crime Prediction Model using Long Short-Term Memory Recurrent Neural Network," International Journal of Recent Technology and Engineering (IJRTE), vol. 9, no. 4, pp. 402–405, Nov. 2020, doi: 10.35940/ijrte. D5025.119420.

[24] "Crime - Definition, Meaning & Synonyms | Vocabulary.com." Accessed: Feb. 19, 2023. [Online]. Available: https://www.vocabulary.com/dictionary/crime

[25] "Africa: Crime Index by City 2023." Accessed: Feb. 19, 2023. [Online]. Available: https://www.numbeo.com/crime/region_rankings.jsp?title=2023&region=002

[26] "Forecasting - Overview, Methods and Features, Steps." Accessed: Feb. 19, 2023. [Online].Available:https://corporatefinanceinstitute.com/resources/valuation/forecasting/

[27] "Time Series Analysis: The Basics." Accessed: Feb. 19, 2023. [Online]. Available: https://www.abs.gov.au/websitedbs/d3310114.nsf/home/time+series+analysis:+the+basics

[28] S. Jha, E. Yang, A. O. Almagrabi, A. K. Bashir, and G. P. Joshi, "Comparative analysis of time series model and machine testing systems for crime forecasting," Neural

Computing and Applications, vol. 33, no. 17. Springer Science and Business Media Deutschland GmbH, pp. 10621–10636, Sep. 01, 2021. doi: 10.1007/s00521-020-04998-1.

[29] “Predictive modeling, analytics and machine learning | SAS UK.” Accessed: Feb. 19, 2023. [Online]. Available: https://www.sas.com/en_gb/insights/articles/analytics/a-guide-to-predictive-analytics-and-machine-learning.html

[30] “Time series forecasting with ARMA and InfluxDB | InfoWorld.” Accessed: Feb. 21, 2023. [Online]. Available: https://www.infoworld.com/article/3681894/time-series-forecasting-with-arma-and-influxdb.html

[31] S. Aluru, Jaypee Institute of Information Technology University, University of Florida. College of Engineering, IEEE Computer Society, IEEE Computer Society. Technical Committee on Parallel Processing, and Institute of Electrical and Electronics Engineers, 2018 Eleventh International Conference on Contemporary Computing (IC3) : 2-4 August 2018, Jaypee Institute of Information Technology, Noida, India.

[32] E. P. Patulin and R. E. Talingting, “Crime prediction using autoregressive integrated moving average (ARIMA) algorithm,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 3, pp. 720–724, May 2019, doi: 10.30534/ijatcse/2019/59832019.

[33] H. Tariq, M. K. Hanif, M. U. Sarwar, S. Bari, M. S. Sarfraz, and R. J. Oskouei, “Employing deep learning and time series analysis to tackle the accuracy and robustness of the forecasting problem,” Security and Communication Networks, vol. 2021, 2021, doi: 10.1155/2021/5587511.

[34] J. Vimala Devi and K. S. Kavitha, “Automating Time Series Forecasting on Crime Data using RNN-LSTM.” [Online]. Available: www.ijacsa.thesai.org

[35] “Time Series Forecasting and Analysis: Arima and Seasonal-Arima | by Subham Sarkar | Analytics Vidhya | Medium.” Accessed: Nov. 11, 2023. [Online]. Available: https://medium.com/analytics-vidhya/time-series-forecasting-and-analysis-arima-and-seasonal-arima-cacaf61ae863

[36] J. Nicholls, A. Kuppa, and N. A. Le-Khac, “Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape,” IEEE Access, vol. 9, pp. 163965–163986, 2021, doi: 10.1109/ACCESS.2021.3134076.

[37] Y. Ning, S. He, Z. Wu, C. Xing, and L. J. Zhang, “Review of deep learning based speech synthesis,” Applied Sciences (Switzerland), vol. 9, no. 19. MDPI AG, Oct. 01, 2019. doi: 10.3390/app9194050.

[38] P. Stalidis, T. Semertzidis, and P. Daras, “Examining Deep Learning Architectures for Crime Classification and Prediction,” Forecasting, vol. 3, no. 4, pp. 741–762, Dec. 2021, doi: 10.3390/forecast3040046.

[39] Suhartono, S. P. Rahayu, D. D. Prastyo, D. G. P. Wijayanti, and Juliyanto, “Hybrid model for forecasting time series with trend seasonal and salendar variation patterns,” in Journal of Physics: Conference Series, Institute of Physics Publishing, Sep. 2017. doi: 10.1088/1742-6596/890/1/012160.

[40] G. M. Misganew, “Semantic-Aware Amharic Text Classification Using Deep Learning Approach,” 2020.

[41] I. Sutskever, J. Martens, and G. Hinton, “Generating Text with Recurrent Neural Networks,” 2011.

[42] S. University of Technology and Institute of Electrical and Electronics Engineers, ICSEng 2018 : 26th International Conference on Systems Engineering : conference proceedings : December 18th - 20th, 2018, University of Technology Sydney, Australia.

[43] J. Yin, “Crime Prediction Methods Based on Machine Learning: A Survey,” Computers, Materials and Continua, vol. 74, no. 2, pp. 4601–4629, 2023, doi: 10.32604/cmc.2023.034190.

[44] N. Tasnim, I. T. Imam, and M. M. A. Hashem, “A Novel Multi-Module Approach to Predict Crime Based on Multivariate Spatio-Temporal Data Using Attention and Sequential Fusion Model,” IEEE Access, vol. 10, pp. 48009–48030, 2022, doi: 10.1109/ACCESS.2022.3171843.

[45] M. J. Hamayel and A. Y. Owda, “A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms,” AI, vol. 2, no. 4, pp. 477–496, Oct. 2021, doi: 10.3390/ai2040030.

[46] G. H. Kwak, C. W. Park, H. Y. Ahn, S. Il Na, K. Do Lee, and N. W. Park, “Potential of bidirectional long short-term memory networks for crop classification with multitemporal remote sensing images,” Korean Journal of Remote Sensing, vol. 36, no. 4, pp. 515–525, 2020, doi: 10.7780/kjrs.2020.36.4.2.

[47] Jordan University of Science & Technology, Institute of Electrical and Electronics Engineers. Jordan Section, and Institute of Electrical and Electronics Engineers, 2018 9th International Conference on Information and Communication Systems (ICICS) : 3-5 April, 2018, Jordan University of Science and Technology, Irbid, Jordan.

[48] IEEE Staff, 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2019.

[49] M. Alkaff, N. Fatanah Mustamin, G. Aditya, and A. Firdaus, “International Journal of Intelligent Systems and Applications in Engineering Prediction of Crime Rate in Banjarmasin City Using RNN-GRU Model.” [Online]. Available: www.ijisae.org

[50] “Gated Recurrent Unit (GRU) With PyTorch.” Accessed: Nov. 11, 2023. [Online]. Available: https://blog.floydhub.com/gru-with-pytorch/

[51] Tufts University and IEEE Circuits and Systems Society, 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) : August 6-9, 2017, Boston, MA, USA.

[52] “Bidirectional Gated Recurrent Unit Neural Network (BGRU). The general... | Download Scientific Diagram.” Accessed: Sep. 30, 2023. [Online]. Available: https://www.researchgate.net/figure/Bidirectional-Gated-Recurrent-Unit-Neural-Network-BGRU-The-general-architecture-of-a_fig2_349634872

[53] “Bidirectional Gated Recurrent Unit Neural Network (BGRU). The general... | Download Scientific Diagram.” Accessed: Nov. 11, 2023. [Online]. Available:

https://www.researchgate.net/figure/Bidirectional-Gated-Recurrent-Unit-Neural-Network-BGRU-The-general-architecture-of-a_fig2_349634872

[54] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science." Accessed: Mar. 22, 2023. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[55] H. Verma, S. Lotia, and A. Singh, "Convolutional neural network based criminal detection," in IEEE Region 10 Annual International Conference, Proceedings/TENCON, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 1124–1129. doi: 10.1109/TENCON50793.2020.9293926.

[56] H. Jaeger, "Echo state network," Scholarpedia, vol. 2, no. 9, p. 2330, 2007, doi: 10.4249/SCHOLARPEDIA.2330.

[57] U. M. Butt, S. Letchmunan, F. H. Hassan, and T. W. Koh, "Hybrid of deep learning and exponential smoothing for enhancing crime forecasting accuracy," PLoS One, vol. 17, no. 9 September, Sep. 2022, doi: 10.1371/journal.pone.0274172.

[58] F. Pan, H. Zhang, and M. Xia, "A hybrid time-series forecasting model using extreme learning machines," in 2009 2nd International Conference on Intelligent Computing Technology and Automation, ICICTA 2009, 2009, pp. 933–936. doi: 10.1109/ICICTA.2009.232.

[59] "Welkite - Wikipedia." Accessed: Dec. 23, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Welkite

[60] "Gurage Zone - Wikipedia." Accessed: Dec. 26, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Gurage_Zone

[61] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," IEEE Access, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[62] N. Tasnim, I. T. Imam, and M. M. A. Hashem, "A Novel Multi-Module Approach to Predict Crime Based on Multivariate Spatio-Temporal Data Using Attention and

Sequential Fusion Model," IEEE Access, vol. 10, pp. 48009–48030, 2022, doi: 10.1109/ACCESS.2022.3171843.

[63] N. Tasnim, I. T. Imam, and M. M. A. Hashem, "A Novel Multi-Module Approach to Predict Crime Based on Multivariate Spatio-Temporal Data Using Attention and Sequential Fusion Model," IEEE Access, vol. 10, pp. 48009–48030, 2022, doi: 10.1109/ACCESS.2022.3171843.

APPENDICES

Appendix A. Sample Code

Appendix A.1. Data Loading and Grouping

```python
# Load and preprocess the dataset
data = pd.read_csv('crime_dataset.csv')
data['Crime_date'] = pd.to_datetime(data['Crime_date'])
data = data.set_index('Crime_date')
data['Crime_time'] = pd.to_datetime(data['Crime_time']).dt.time
# Hourly prediction
data_hourly = data.groupby([data.index.date, data.index.hour, 'Crime_type']).size().unstack(fill_value=0)
ct =data_hourly.astype('float32')
hourly_scaler = MinMaxScaler()
scaled_hourly_data = hourly_scaler.fit_transform(data_hourly)
# Daily prediction
daily_data = data.groupby([data.index.date, 'Crime_type']).size().unstack(fill_value=0)
daily_scaler = MinMaxScaler()
scaled_daily_data = daily_scaler.fit_transform(daily_data)
# Monthly prediction
monthly_data = data.groupby([data.index.year, data.index.month, 'Crime_type']).size().unstack(fill_value=0)
monthly_scaler = MinMaxScaler()
scaled_monthly_data = monthly_scaler.fit_transform(monthly_data)
# Number of previous hours/days/months to consider
sequence_length = 30

def prepare_sequences(data, sequence_length):
    X = []
    y = []
    for i in range(sequence_length, len(data)):
        X.append(data[i - sequence_length:i])
        y.append(data[i])
    return np.array(X), np.array(y)

# Now use the defined function to prepare sequences for each granularity
X_hourly, y_hourly = prepare_sequences(scaled_hourly_data, sequence_length)
X_daily, y_daily = prepare_sequences(scaled_daily_data, sequence_length)
X_monthly, y_monthly = prepare_sequences(scaled_monthly_data, sequence_length)

# Placeholder for the split_train_test function
def split_train_test(X, y):
    train_size = int(0.8 * len(X))
    X_train, X_test = X[:train_size], X[train_size:]
    y_train, y_test = y[:train_size], y[train_size:]
    return X_train, X_test, y_train, y_test
```

Data Loading and Grouping Sample Code

Appendix A.2. LSTM Model Building

```python
# Define hyperparameters
epochs = 30
batch_size = 64
lstm_units = 100
dropout_rate = 0.2
# Build model for hourly prediction
model_hourly = Sequential()
model_hourly.add(LSTM(units=lstm_units, input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_hourly.add(Dropout(rate=dropout_rate))
model_hourly.add(Dense(units=y_train_hourly.shape[1], activation='tanh'))
model_hourly.compile(loss='mse', optimizer='adam')
early_stopping_hourly = EarlyStopping(patience=5, restore_best_weights=True)
lstmhistory_hourly = model_hourly.fit(X_train_hourly, y_train_hourly, epochs=epochs, batch_size=batch_size,
                                  validation_data=(X_val_hourly, y_val_hourly), callbacks=[early_stopping_hourly])
lstmtest_loss_hourly = model_hourly.evaluate(X_test_hourly, y_test_hourly)
print(f'Hourly Test Loss: {lstmtest_loss_hourly}')
model_hourly.summary()
# Build model for daily prediction
model_daily = Sequential()
model_daily.add(LSTM(units=lstm_units, input_shape=(X_train_daily.shape[1], X_train_daily.shape[2])))
model_daily.add(Dropout(rate=dropout_rate))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_daily.compile(loss='mse', optimizer='adam')
early_stopping_daily = EarlyStopping(patience=5, restore_best_weights=True)
lstmhistory_daily = model_daily.fit(X_train_daily, y_train_daily, epochs=epochs, batch_size=batch_size,
                                validation_data=(X_val_daily, y_val_daily), callbacks=[early_stopping_daily])
lstmtest_loss_daily = model_daily.evaluate(X_test_daily, y_test_daily)
print(f'Daily Test Loss: {lstmtest_loss_daily}')
model_daily.summary()
# Build model for monthly prediction
model_monthly = Sequential()
model_monthly.add(LSTM(units=lstm_units, input_shape=(X_train_monthly.shape[1], X_train_monthly.shape[2])))
model_monthly.add(Dropout(rate=dropout_rate))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
model_monthly.compile(loss='mse', optimizer='adam')
early_stopping_monthly = EarlyStopping(patience=5, restore_best_weights=True)
model_monthly.summary()
lstmhistory_monthly = model_monthly.fit(X_train_monthly, y_train_monthly, epochs=epochs, batch_size=batch_size,
                                    validation_data=(X_val_monthly, y_val_monthly), callbacks=[early_stopping_monthly])
lstmtest_loss_monthly = model_monthly.evaluate(X_test_monthly, y_test_monthly)
```

LSTM Model Building Sample Code for Crime Type Prediction

Appendix A.3. Bi-LSTM Model Building

```
# Build model for hourly prediction
model_hourly = Sequential()
model_hourly.add(Bidirectional(LSTM(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_hourly.add(Dropout(rate=dropout_rate))
model_hourly.add(Dense(units=y_train_hourly.shape[1], activation='tanh'))
model_hourly.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_hourly = EarlyStopping(patience=5, restore_best_weights=True)
bilstmhistory_hourly = model_hourly.fit(X_train_hourly, y_train_hourly, epochs=epochs, batch_size=batch_size,
                                  validation_data=(X_val_hourly, y_val_hourly), callbacks=[early_stopping_hourly])
bilstmtest_loss_hourly = model_hourly.evaluate(X_test_hourly, y_test_hourly)
print(f'Hourly Test Loss: {bilstmtest_loss_hourly}')
print(" ")
print("Bi-LSTM-Hourly Prediction Summary ")
model_hourly.summary()
# Build model for daily prediction
model_daily = Sequential()
model_daily.add(Bidirectional(LSTM(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_daily.add(Dropout(rate=dropout_rate))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_daily.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_daily = EarlyStopping(patience=5, restore_best_weights=True)
bilstmhistory_daily = model_daily.fit(X_train_daily, y_train_daily, epochs=epochs, batch_size=batch_size,
                                  validation_data=(X_val_daily, y_val_daily), callbacks=[early_stopping_daily])
bilstmtest_loss_daily = model_daily.evaluate(X_test_daily, y_test_daily)
print(f'Daily Test Loss: {bilstmtest_loss_daily}')
print(" ")
print("Bi-LSTM-Daily Prediction Summary ")
model_daily.summary()
# Build model for monthly prediction
model_monthly = Sequential()
model_monthly.add(Bidirectional(LSTM(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_monthly.add(Dropout(rate=dropout_rate))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
model_monthly.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_monthly = EarlyStopping(patience=5, restore_best_weights=True)
bilstmhistory_monthly = model_monthly.fit(X_train_monthly, y_train_monthly, epochs=epochs, batch_size=batch_size,
                                  validation_data=(X_val_monthly, y_val_monthly), callbacks=[early_stopping_monthly])
bilstmtest_loss_monthly = model_monthly.evaluate(X_test_monthly, y_test_monthly)
print(f'Monthly Test Loss: {bilstmtest_loss_monthly}')
```

Bi-LSTM Model Building Sample Code for Crime Type Prediction

Appendix A.4. GRU Model Building

```
# Build model for hourly prediction
model_hourly = Sequential()
model_hourly.add(GRU(units=lstm_units, input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_hourly.add(Dropout(rate=dropout_rate))
model_hourly.add(Dense(units=y_train_hourly.shape[1], activation='tanh'))
model_hourly.compile(loss='mse', optimizer='adam')
early_stopping_hourly = EarlyStopping(patience=5, restore_best_weights=True)
gruhistory_hourly = model_hourly.fit(X_train_hourly, y_train_hourly, epochs=epochs, batch_size=batch_size,
                                  validation_data=(X_val_hourly, y_val_hourly), callbacks=[early_stopping_hourly])
grutest_loss_hourly = model_hourly.evaluate(X_test_hourly, y_test_hourly)
print(f'Hourly Test Loss: {grutest_loss_hourly}')
print(" ")
print("GRU-Hourly Prediction Summary ")
model_hourly.summary()
# Build model for daily prediction
model_daily = Sequential()
model_daily.add(GRU(units=lstm_units, input_shape=(X_train_daily.shape[1], X_train_daily.shape[2])))
model_daily.add(Dropout(rate=dropout_rate))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_daily.compile(loss='mse', optimizer='adam')
early_stopping_daily = EarlyStopping(patience=5, restore_best_weights=True)
gruhistory_daily = model_daily.fit(X_train_daily, y_train_daily, epochs=epochs, batch_size=batch_size,
                                validation_data=(X_val_daily, y_val_daily), callbacks=[early_stopping_daily])
grutest_loss_daily = model_daily.evaluate(X_test_daily, y_test_daily)
print(f'Daily Test Loss: {grutest_loss_daily}')
print(" ")
print("GRU-Daily Prediction Summary ")
model_daily.summary()
# Build model for monthly prediction
model_monthly = Sequential()
model_monthly.add(GRU(units=lstm_units, input_shape=(X_train_monthly.shape[1], X_train_monthly.shape[2])))
model_monthly.add(Dropout(rate=dropout_rate))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
model_monthly.compile(loss='mse', optimizer='adam')
early_stopping_monthly = EarlyStopping(patience=5, restore_best_weights=True)
gruhistory_monthly = model_monthly.fit(X_train_monthly, y_train_monthly, epochs=epochs, batch_size=batch_size,
                                    validation_data=(X_val_monthly, y_val_monthly), callbacks=[early_stopping_monthly])
grutest_loss_monthly = model_monthly.evaluate(X_test_monthly, y_test_monthly)
print(f'Monthly Test Loss: {grutest_loss_monthly}')
```

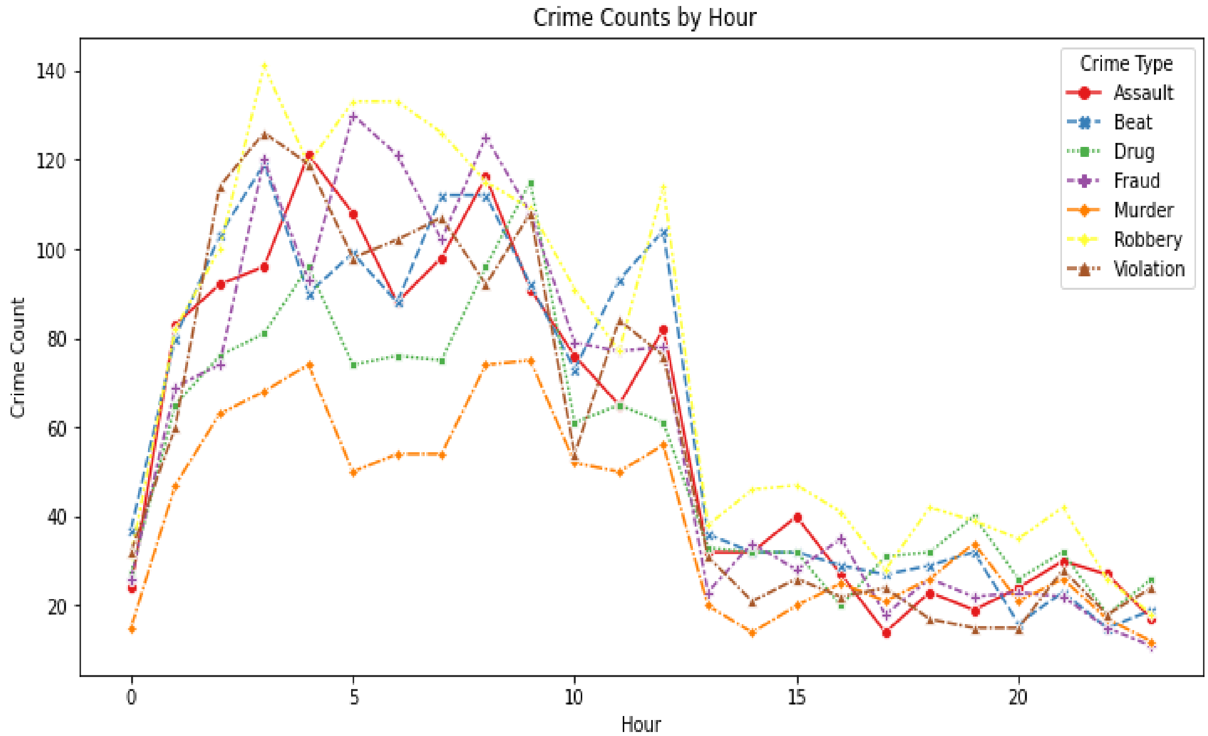
GRU Model Building Sample Code for Crime Type Prediction

Appendix A.5. Bi-GRU Model Building

```python
# Build model for hourly prediction
model_hourly = Sequential()
model_hourly.add(Bidirectional(GRU(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_hourly.add(Dropout(rate=dropout_rate))
model_hourly.add(Dense(units=y_train_hourly.shape[1], activation='tanh'))
model_hourly.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_hourly = EarlyStopping(patience=5, restore_best_weights=True)
bigruhistory_hourly = model_hourly.fit(X_train_hourly, y_train_hourly, epochs=epochs, batch_size=batch_size,
                                validation_data=(X_val_hourly, y_val_hourly), callbacks=[early_stopping_hourly])
bigrutest_loss_hourly = model_hourly.evaluate(X_test_hourly, y_test_hourly)
print(f'Hourly Test Loss: {bigrutest_loss_hourly}')
print(" ")
print("Bi-GRU-Hourly Prediction Summary ")
model_hourly.summary()
# Build model for daily prediction
model_daily = Sequential()
model_daily.add(Bidirectional(GRU(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_daily.add(Dropout(rate=dropout_rate))
model_daily.add(Dense(units=y_train_daily.shape[1], activation='tanh'))
model_daily.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_daily = EarlyStopping(patience=5, restore_best_weights=True)
bigruhistory_daily = model_daily.fit(X_train_daily, y_train_daily, epochs=epochs, batch_size=batch_size,
                                validation_data=(X_val_daily, y_val_daily), callbacks=[early_stopping_daily])
bigrutest_loss_daily = model_daily.evaluate(X_test_daily, y_test_daily)
print(f'Daily Test Loss: {bigrutest_loss_daily}')
print(" ")
print("Bi-GRU-Hourly Prediction Summary ")
model_daily.summary()
# Build model for monthly prediction
model_monthly = Sequential()
model_monthly.add(Bidirectional(GRU(100), input_shape=(X_train_hourly.shape[1], X_train_hourly.shape[2])))
model_monthly.add(Dropout(rate=dropout_rate))
model_monthly.add(Dense(units=y_train_monthly.shape[1], activation='tanh'))
model_monthly.compile(loss='mean_squared_error', optimizer='adam')
early_stopping_monthly = EarlyStopping(patience=5, restore_best_weights=True)
bigruhistory_monthly = model_monthly.fit(X_train_monthly, y_train_monthly, epochs=epochs, batch_size=batch_size,
                                validation_data=(X_val_monthly, y_val_monthly), callbacks=[early_stopping_monthly])
bigrutest_loss_monthly = model_monthly.evaluate(X_test_monthly, y_test_monthly)
print(f'Monthly Test Loss: {bigrutest_loss_monthly}')
```

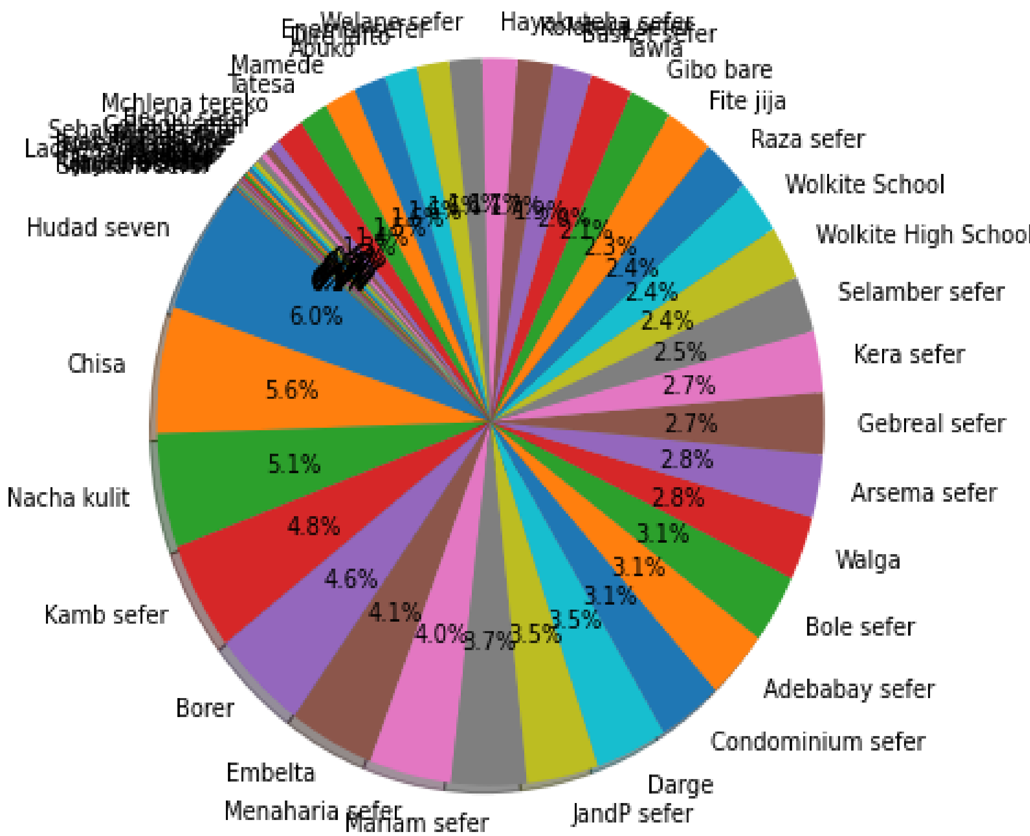
Bi-GRU Model Building Sample Code for Crime Type Prediction

Appendix B. Each Crime Type Count over Hours Graph



Crime Type Counts Per Hour Plotting

Appendix C. Crime Location Percentage Graph



Crime Location Percentage Plotting

Appendix D. Wolkite Police Department Sample Crime Dataset

Crime Dataset

Crime_date	*Crime_time*	*Crime_type*	*Crime_location*
1/1/2008	12:25:00 PM	Assault	Embelta
1/3/2008	6:00:00 PM	Murder	Darge
1/1/2008	4:50:00 AM	Fraud	Chisa
1/11/2008	4:55:00 AM	Violation	Hudad seven
1/15/2008	9:18:00 AM	Drug	Embelta
...	...	...	...
12/15/2014	1:40 PM	Beat	JandP sefer
12/16/2014	9:30 PM	Assault	Kamb sefer
12/17/2014	11:20 AM	Robbery	Menaharia sefer
12/18/2014	12:45 PM	Robbery	Bole sefer
12/19/2014	1:15 AM	Fraud	Adebabay sefer

Sample Collected Crime Dataset