CRIME PREDICTION ANALYSIS

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF **BACHELOR OF ENGINEERING** IN **COMPUTER SCIENCE AND ENGINEERING** OF THE ANNA UNIVERSITY

PROJECT WORK

2022

Submitted by

DIVYA A

1817116

HAFEEZA BEGAM M

1817118

KEERTHANA K

1817123

PAVITHRA T

1817137

Under the Guidance of

Dr.K.KUMAR M.E.,Ph.D

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University)

COIMBATORE - 641 013

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University)

COIMBATORE - 641 013

PROJECT WORK

JUNE 2022

This is to certify that this project work entitled

CRIME PREDICTION ANALYSIS

is the bonafide record of project work done by

DIVYA A

1817116

HAFEEZA BEGUM M

1817118

KEERTHANA K

1817123

PAVITHRA T

1817137

of B.E.(COMPUTER SCIENCE AND ENGINEERING) during the year 2021 - 2022

Project Guide	Head of Department
Dr.K.KUMAR M.E.,Ph.D	**Dr.J.C.MIRACLIN JOYCE PAMILA M.E.,Ph.D**

Submitted for the Project Viva-Voce examination held on _____

Internal Examiner	External Examiner

ACKNOWLEDGEMENT

We express our sincere gratitude to **Dr.P.THAMARAI, Ph.D.,** Principal, Government College of Technology, Coimbatore for providing us all facilities that we needed for the completion of this project.

We whole-heartedly express our thankfulness and gratitude to **Dr.J.C.MIRACLIN JOYCE PAMILA, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering, Government College of Technology, for helping us to successfully carry out this project.

Our thankfulness and gratitude to our respectable project guide **DR.K.KUMAR, M.E., Ph.D.,** Associate Professor(CAS) of the Department of Computer Science and Engineering who has been an immense help through the various phases of the project. With his potent ideas and excellent guidance, we were able to comprehend the essential aspects involved.

We extend our sincere thanks to **Dr.S.RATHI, M.E., Ph.D.,** Professor (CAS), **Prof.T.RAJASENBAGAM, M.E.,** Assistant Professor, **Dr.A.MEENA KOWSHALYA, M.E., Ph.D.,** Assistant Professor, **Dr.R.BHAVANI, M.E., Ph.D.,** Assistant Professor, **Dr.R.MUTHURAM,M.E., Ph.D.,** Assistant Professor, **Prof.L.SUMATHI, M.E.,** Assistant Professor and **Prof.G.POOVILA, M.E,** Assistant Professor for all their valuable suggestions to the completion of the project.

We thank all the non-teaching staff and our friends for their cooperation towards the successful completion of the project.

We would like to dedicate the work to our parents for their constant encouragement throughout the project.

SYNOPSIS

Crime is one of the major issues is continuing to grow in intensity and complexity. In the recent years, crime is one of the social problems influencing the nature of life and economic development in a community. Crime can be divided into a few types such as crime against properties (theft, burglary, and robbery) and crime of aggression (homicides, assaults and rape). The availability of information technologies has enabled law enforcement to collect detailed information of crime data. With the increasing numbers of crimes nowadays, crime analysis is needed which comprises measure and procedure that intend to reduce the risk of crime. Crime analysis can be done through both quantitative and qualitative methods. Qualitative approaches in predicting crime such as scenario writing or environmental scanning are valuable in identifying the future of criminal activity. Meanwhile, quantitative method is used to predict the crime rates in future. Moreover, crime analysis is a practical approach to analyze and identify the pattern of crimes.

A crime is a deliberate act that can cause physical or psychological harm, as well as property damage or loss, and can lead to punishment by a state or other authority according to the severity of the crime. The number and forms of criminal activities are increasing at an alarming rate, forcing agencies to develop efficient methods to take preventive measures. In the current scenario of rapidly increasing crime, traditional crime-solving techniques are unable to deliver results, being slow paced and less efficient. Thus, if we can come up with ways to predict crime in detail before it occurs that can assist police officers, it would lift the burden of police and help in preventing crimes. To achieve this, we suggest including machine learning (ML) algorithms and deep learning algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Safety and security lie at the heart of the prosperity of any nation. Citizens want to feel safe and secure. But today security is challenged in all aspects of our daily lives. In this new reality, national, regional and local governments need to view citizen safety in a holistic light and work across borders to achieve it.

In this era of modern world, our popularity is increasing and citification carries enormous general, financial and environmental, while presenting challenges in urban management issues such as traffic resource planning, environment and safe water quality, public policy and public safety services. In addition, represent the most crime rates in larger cities, crime reducing is becoming one of the most important social issues in enormous metropolitan areas as it affects people security, youngster growth and person socio-economic status. For daily purposes people have to go many places every day and many times in the daily lives they face numerous security issues such as hijacking, kidnapping, harassment etc. Some people not aware of visiting new places is really safe or not that's why they endup with many unpleasant circumstances.

Crime is one of the biggest and dominating problem in our society and its prevention is an important task. Daily there are huge numbers of crimes committed frequently. This require keeping track of all the crimes and maintaining a database for same which may be used for future reference. Crime in a way, influences organizations and institutions when occurred frequently in a society. Thus, it is necessary to study the factors and relations between different crimes and to find a way to accurately predict and avoid these crimes. Recently law enforcement agencies have been moving towards a more empirical, data driven approach to predictive policing. However, even with new data-driven approaches to predict crime, the fundamental job of crime analysts still remains difficult and often manual. specific patterns of crime are not very easy to find by way of automated tools, whereas larger-scale density-based trends comprised mainly of background crime levels are much easier for data-driven approaches and software to estimate. Crime is naturally unpredictable. It is not necessarily

random neither does it take place persistently in space or time. A Good theoretical understanding is needed to provide practical crime revention solutions that equivalent to specific places and times. Crime analysis takes past crime data to predict future crime. Crime prediction for future crime is a process that finds out crime rate change from one year to the next. The crime prediction approach involves relating past crime trends with factors that will influence the future scope of crime.

The proposed method is to predict the severity of the crime and predict the crime type depends on the past crime dataset in that location with a specified time block of the day. Machine Learning techniques can be used to predict the crime patterns and thus may assist in further necessary actions based on the historical data before prediction, the data collected was pre processed in order to enhance the performance.

To find hotspots of criminal activities based on the time of day, Binary classification techniques were used and based on those results whether the crime is severe or not severe was decided.

For finding the Crime type ,Multi Class classification techniques were used based on them predicting which crime type occur is done.

CHAPTER 2

LITERATURE REVIEW

In literature various techniques has been proposed by researchers to predict the occurrence of the crime.

[1] **Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques**. This study analyze crime prediction in the Chicago and Los Angeles datasets implementing the Logistic Regression, SVM, Naïve Bayes, KNN, Decision Tree, MLP, Random Forest, XGBoost algorithms, time-series analysis by LSTM, creating a visual summary through exploratory data analysis and crime forecasting for the crime rate and high intensity crime areas for subsequent years by using an ARIMA model.The performance of LSTM for time series analysis was reasonably adequate in order of magnitude of root mean square error (RMSE) and mean absolute error (MAE), on both data sets.

[2] **Prediction of Crime Hotspots based on Spatial Factors of Random Forest.** The spatiotemporal clustering modeled through self-excited point process in seismology is applicable to the application of criminology, and proposed theoretical models for hot spot understanding, such as random walking process and selfexcited point process. Random forest machine learning method used in the research of crime hot spot prediction. Various types of micro-space elements have a certain effect on various types of crime. For prediction of crime hot spot consider not only historical crime data but also surrounding environmental factors and other relevant factors. In the research of crime hotspot prediction, this paper adopts the random forest algorithm to construct the model, and adds covariates to observe the overall prediction accuracy of the research area, observed the changes in the prediction results. Changes in the overall prediction effect of the model by adding different data type features to the prediction model.

[3] **An Overview on Crime Prediction Methods.** This study was conducted to review and examine the current methods used in crime prediction analysis. Support Vector Machine method used in hot-spots prediction to predefine a level of crime rate and given the percentage of data. Fuzzy time series was applied in

orders to discover a crime pattern in community. Prediction crime using artificial neural network. ANN educated using geographical clusters of crime data to ease predictive modeling. The scanning algorithm based on geographical crime incidents used to identify clusters with relatively high level of crime hotspots. Logistic regression model to examine the relationship between the predicting factors of crimes and burglary occurrence probability. The factors used including time of day (category data), day of the week (discrete), barriers (physical structures that will interfere an individual's egress from a targeted residence), connectors (the amount of access bridges, streets and pathways relative to the targeted residence) and repeat victimization (the occurrence of the offense at the same residence following the initial offense over the calendar year, same category data).

[4] **Crime Analysis and Prediction Using Machine Learning.** This paper has employed several mathematical and statistical tools in building effective models for the prevention and patrolling of crime. The mathematical results can be used by the administration in crime prevention departments to work in a systematic and coordinated manner to identify the reason of increasing crime in particular location. This paper has attempted to use techniques such as Exploratory Data Analysis, Data Analysis and Modeling techniques which includes systematic steps like the collection and compilation of primary/secondary data, analysis of the relevant historical and contemporary data using the development of a linear regression model, creating a model with the most correlated features.

[5] **A knowledge centric hybridized approach for crime classification incorporating deep bi-LSTM neural network.** In this paper, with the help of a Bi-LSTM model , a method of classification is proposed that classifies the crime-related news into types, and based on those, an ontological representation has been created that records frequency and spatial data. This ensures that in the future, crime statistics can be updated in a fast and efficient manner, and also statistical methods of analytics can be performed. There are a variety of techniques for data mining in crime data such as sequential pattern mining, clustering techniques, string comparison, association rule mining, and classification

Table 2.1 Summary of the crime Detection Methods

Ref	Technique	Advantage	Disadvantage
[1]	Fuzzy Theory	Provides High degree of robustness and accuracy in crime forecasting together with lesser computational complexity	prediction system can be affected by increasing the number of inputs
[3]	Multivariate Time Series	Consider more than one factor for analysis and looks at various independent variables that influence dependent variables	statistical modeling outputs are not always easy to interpret
[5]	Dempster-Shafer Theory	Adding more information reduces the uncertainity interval has lower level of ignorance	Computation effort is High
[6]	AdaBoost	Combines several weak learners to produce a stronger model.	Sensitive to Noisy data and outliers . Needs a quality dataset.
[8]	Long Short-Term Memory	Well-suited to classify, process and predict time series given time lags of unknown duration	Require a lot of resources and time to get trained and become ready for real-world applications.
[9]	Artificial Neural Networks	Able to manage abundant no.of data and input variables.It has great capacity in predicting models	Takes long time to process of large neural network. Quality predictions need large amount of data

According to Table 2.1 the crime detection for the project needs better crime classification model to predict the crime with more accuracy and low computation time .The proposed system has developed with Binary classification, Multi class classfication and Bi-LSTM Model for the crime classification problem.

CHAPTER 3

PROJECT DESIGN

3.1 PROBLEM STATEMENT

Number of crimes in Cities increase, more and more people care about their safety, and the government leader want to build a good environment to the citizens. The education in a country and the rate of poverty, unemployment has an impact in society's safety. Crimes are the significant threat to the humankind. There are many crimes that happens regular interval of time. Perhaps it is increasing and spreading at a fast and vast rate. Crimes happen from small village, town to big cities. Crimes are of different type – robbery, murder, rape, assault, battery, false imprisonment, kidnapping, homicide. Since crimes are increasing there is a need to solve the cases in a much faster way. The crime activities have been increased at a faster rate and it is the responsibility of police department to control and reduce the crime activities. Crime prediction and criminal identification are the major problems to the police department as there are tremendous amount of crime data that exist. There is a need of technology through which the crime can be predicted before they occur.

3.2 OBJECTIVE

Objective of this project is to predict the occurrence of a crime at a location at a specific time of a day and to anticipate if a particular neighbourhood in the city, at any given duration of the day will be a crime hotspot or not, with an acceptable rate of accuracy. To incorporate the impact of housing and inhabitation, literacy rate, employment and socioeconomic status on the crime occurrence rate. Visualization of dataset is done to analyze the crimes which may have occurred in the country. Methodology would be to train a model for prediction. The training would be done using the training data set which will be validated using the test dataset. Selective machine learning algorithms and Deep learning techniques used to build models which predict the crime occurrence based on the past crime records. This work helps the law enforcement agencies to predict and detect crimes thereby reduces the crime rate.

3.3 PROPOSED METHOD

Proposed Method is to predict the crime occurrence in the city at specified location with specified time block of the day. To predict the crimes which are sever or not is done using Binary classification model. Binary classification model developed using different classification techniques which are Decision Tree, Logistic Regression, k-Nearest Neighbour, Random Forest Methods to find hotspots of criminal activities based on the time of day. To predict which crime type occur is done using Multi Class Classification model. Multi Class Classification model developed using different Multi Class classification techniques which are Decision Tree, Logistic Regression, Random Forest , Bi-LSTM Methods to find the Crime type using the given test data.

3.4 PROJECT WORKFLOW

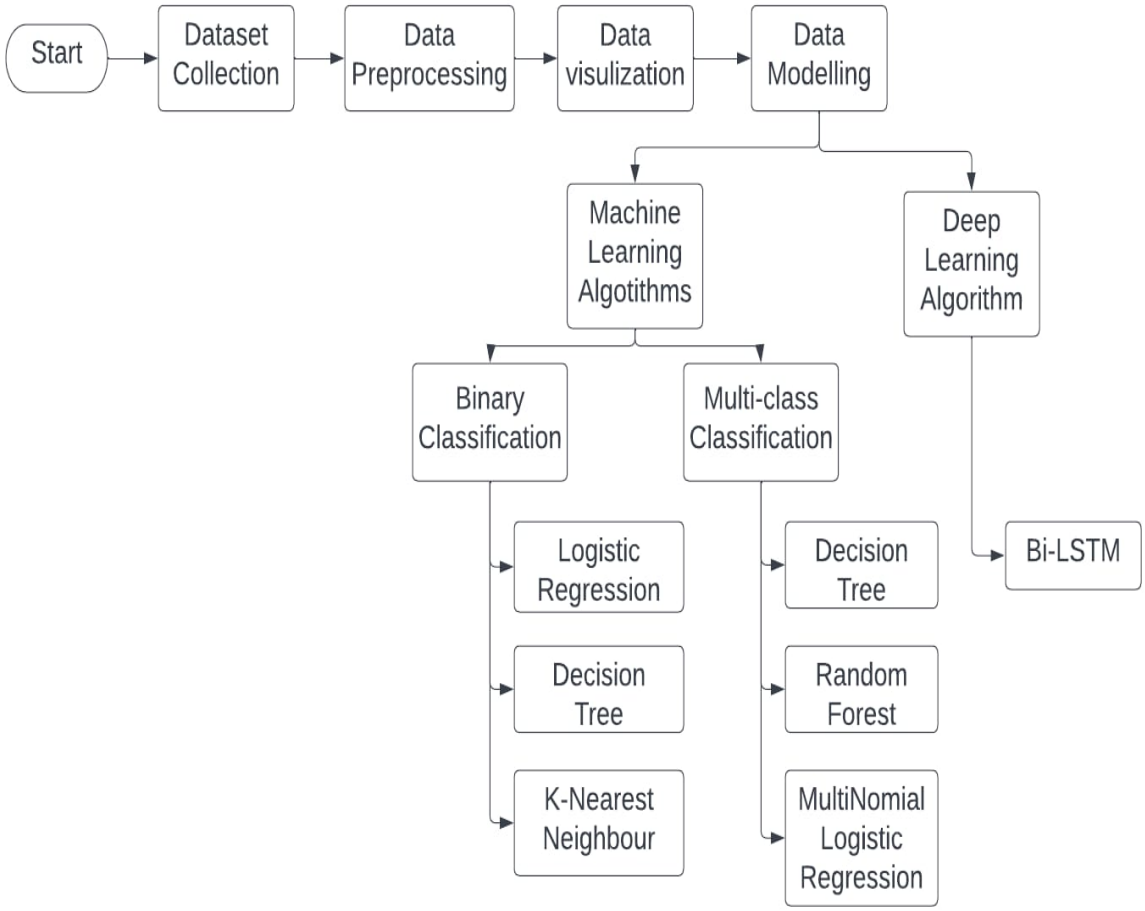


Fig.3.1 Project Work Flow

CHAPTER 4

PROPOSED SYSTEM

4.1 BI-DIRECTIONAL LONG SHORT TERM MEMORY WITH ATTENTION ARCHITECTURE

Bidirectional long short term memory is a type of LSTM model which processes the data in both forward and backward direction. LSTM are a special kind of recurrent neural networks Which are very useful when dealing with sequential data like time series data and NLP data. Long Short Term Memory Network is an advanced RNN a sequential network that allows information to persist. A recurrent neural network is also known as RNN is used for persistent memory.

4.1.1 RECURRENT NEURAL NETWORK

A recurrent neural network (RNN) is a special type of an artificial neural network adapted to work for time series data or data that involves sequences. Ordinary feed forward neural networks are only meant for data points which are independent of each other. Data in a sequence such that one data point depends upon the previous data point to modify the neural network to incorporate the dependencies between these data points. RNNs have the concept of ‘memory’ that helps them store the states or information of previous inputs to generate the next output of the sequence. RNN Structure shown in fig.4.1

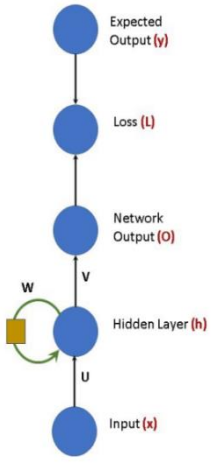


Fig.4.1 RNN Structure

4.1.2 LONG SHORT-TERM MEMORY

Recurrent neural networks (RNN) are a kind of feedforward neural networks which have a recurrent hidden state and the hidden state is activated by the previous states at a certain time. Therefore, RNNs can model the contextual information dynamically and can handle the variable-length sequences. LSTM is a kind of RNNs architecture and addresses the problem of vanishing gradient by replacing the self-connected hidden units with memory blocks. The memory block uses purpose-built memory cell to store information, and it is better at finding and exploiting long range context. The memory units enable the network to be aware of when to learn new information and when to forget old information. A LSTM unit consists of the four components shown in Fig.4.2 . The i is an input gate and it controls the size of the new memory content added to the memory. The f is a forget gate and it determines the amount of the memory that needs to be forgotten. The o is an output gate and it modulates the amount of the output memory content. The c is the cell activation vector and it consists of two components, namely partially forgotten previous memory and modulated new memory. t nominates the t − th moment.

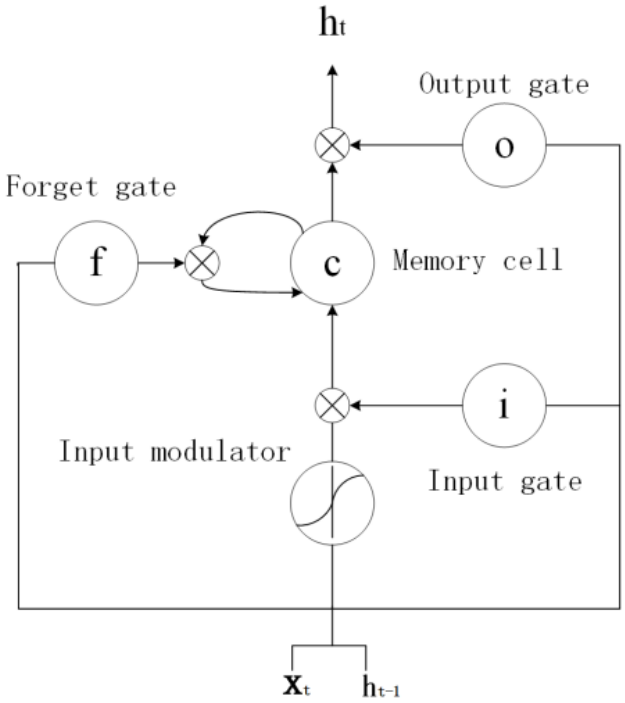


Fig.4.2 LSTM Unit

LSTM has a hidden state where h(t-1) represents the hidden state of the previous timestamp and h(t) is the hidden state of the current timestamp. LSTM has a cell state represented by C(t-1) and C(t) for previous and current timestamp respectively. The hidden state is known as Short term memory and the cell state is known as Long term memory.

i) Forget Gate

In a cell of the LSTM network, the first step is to decide whether to keep the information from the previous timestamp or forget it. Sigmoid function is applied over it. Forget gate (f_t) a number between 0 and 1. If ft is 0 then the network will forget everything and if the value of ft is 1 it will forget nothing.

$$f_t = \sigma\left(x_t * U_f + H_{t-1} * W_f\right)$$

X_t is the input to the current timestamp.

U_f is the weight associated with the input

H_{t-1} is the hidden state of the previous timestamp

W_f is the weight matrix associated with hidden state

ii) Input Gate

Input gate is used to quantify the importance of the new information carried by the input. Sigmoid function applied over I_t .The value of Input (I_t) will be between 0 and 1.

$$i_t = \sigma\left(x_t * U_i + H_{t-1} * W_i\right)$$

iii) New information

The New information that needed to be passed to the cell state is a function of a hidden state at the previous timestamp t-1 and input x at timestamp t. The activation function here is tanh. Due to the tanh function, the value of new information will between -1 and 1. If the value is of New information (N_t) is

negative the information is subtracted from the cell state and if the value is positive the information is added to the cell state at the current timestamp.

$$N_t = \tanh(x_t * U_c + H_{t-1} * W_c)$$

iv) Output Gate

The output gate determines the value of the next hidden state. This state contains information on previous inputs. Its value will also lie between 0 and 1.

$$o_t = \sigma(x_t * U_o + H_{t-1} * W_o)$$

Current hidden state is a function of Long term memory (C_t) and the current output.

$$H_t = o_t * \tanh(C_t)$$

Output of the current timestamp is SoftMax activation on hidden state (h_t).

$$\text{Output} = \text{Softmax}(H_t)$$

LSTM MODEL

LSTM is mainly used to process the sequence data. LSTM Model is shown in fig.4.3

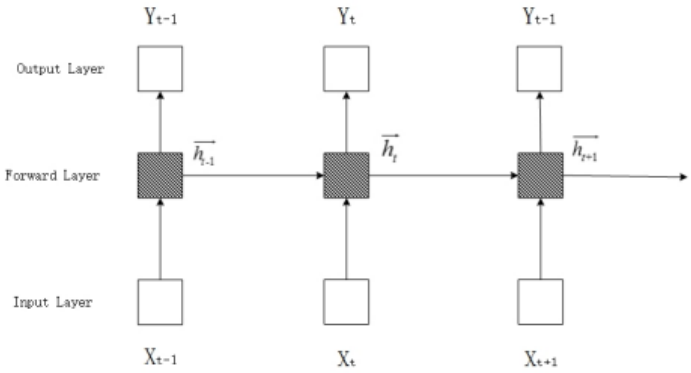


Fig.4.3 Structure of LSTM Model

BI-LSTM WITH ATTENTION

Deep Bi-Directional LSTM Recurrent Neural Network was constructed for the Multi-Class classification task. Bi-LSTM structure shown in fig.4.4.

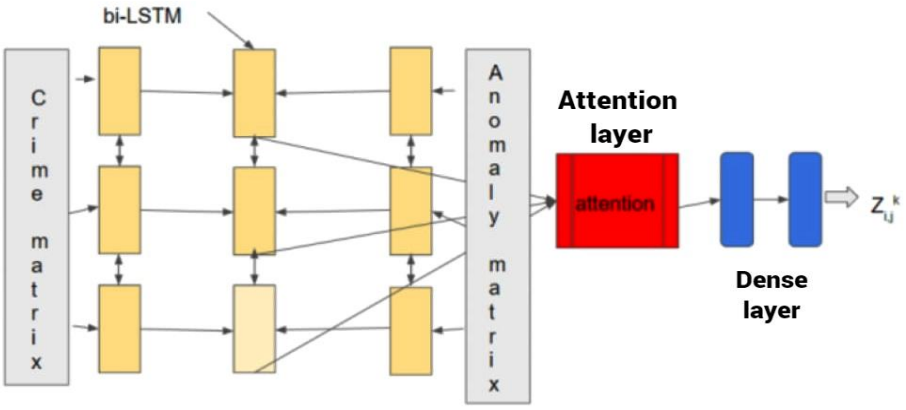


Fig.4.4 BI-LSTM STRUCTURE

Bi-directional LSTM was used, as by using bi-directional networks the inputs are fed one by one from the direction left to right and one input from the direction right to left.

Attention Layer

Attention is a technique that mimics cognitive attention. The effect enhances some parts of the input data while diminishing other parts thought being that the network should devote more focus to that small but important part of the data.

LSTM Layer

A bidirectional LSTM layer learns bidirectional long-term dependencies between time steps of time series or sequence data. These dependencies can be useful when the network want to learn from the complete time series at each time step.

PROPOSED ARCHITECTURE

Table.4.1 Description of Bi-LSTM Architecture

Layer	Description
BI-LSTM Input Layer 1	[(None,10,308)]
BI-LSTM Input Layer 2	[(None,10,308)]
BI-LSTM Layer 3	[(None,10,64)]
Attention Layer	[(None , 128)]
Dense Layer 1	[(64,308)]
Dropout Layer	To prevent overfitting
Dense Layer 2	[(308,308)]

Features of layers

Table 4.1 shows the Bi-LSTM model will take 2 inputs, the Crime Matrix(a) and the anamoly Matrix (b). In this model,

- Optimizer: Adam optimiser
- Learning Rate: 1e-4
- Batch Size: 4
- Context window: 10
- MLP layers: 2
- loss = sigmoid cross entropy with logits
- Hidden Size: 64
- Attention Size: 64

Training task

Given crime records for a period of T days, predict the occurrence/non-occurence of crime category j, in region i, at day T+1. This is done for all crime classes (total 4), for all regions (total 77). Therefore, a total of 308 predictions for each step.

EVALUATION

Metrics used for evaluation are:

1. Macro-F1 score

Macro F1-score (macro-averaged F1 score) is used to assess the quality of problems with multiple classes.

$$\text{Macro F1-score} \quad = \quad \frac{1}{N} \sum_{i=0}^{N} \text{F1-score}_i$$

2. Micro-F1 score

Micro F1-score (micro-averaged F1 score) is used to assess the quality of multi-label binary problems. It measures the F1-score of the aggregated contributions of all classes.

$$\text{Micro F1-score} \quad = \quad 2 * \frac{\text{Micro-precision} * \text{Micro-recall}}{\text{Micro-precision} + \text{Micro-recall}}$$

The implementation for this architecture is explained in chapter 5.1.7.3

4.2 Linear SVM

Support Vector Machine (SVM) is one of the most popular Supervised Learning algorithms used for Classification as well as Regression problems. Primarily used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors.

Support vector machine segregate the given data in the best possible way. When the segregation is done, the distance between the nearest points is known as the margin. The approach is to select a hyperplane with the maximum possible margin between the support vectors in the given data-sets. To select the maximum hyperplane in the given sets, the support vector machine follows the following sets Generate hyperplanes which segregates the classes in the best possible way , Select the right hyperplane with the maximum segregation from either nearest data points. SVM Algorithm is Effective in high dimensional spaces, Still effective in cases where the number of dimensions is greater than the number of samples, Used a subset of training points in the decision function that makes it memory efficient, Different kernel functions can be specified for the decision function that also makes it versatile, When the number of features is much larger than the number of samples, avoid over-fitting in choosing kernel functions and regularization term is crucial. SVM do not directly provide probability estimates, these are calculated using five-fold cross-validation. The implementation for this algorithm is explained in chapter 5.1.7

4.3 Decision Tree

Decision Tree is a Supervised learning technique that can be used for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions is shown in fig.4.6. It is called a decision tree because similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree the CART algorithm is used which stands for Classification and Regression Tree algorithm**.**

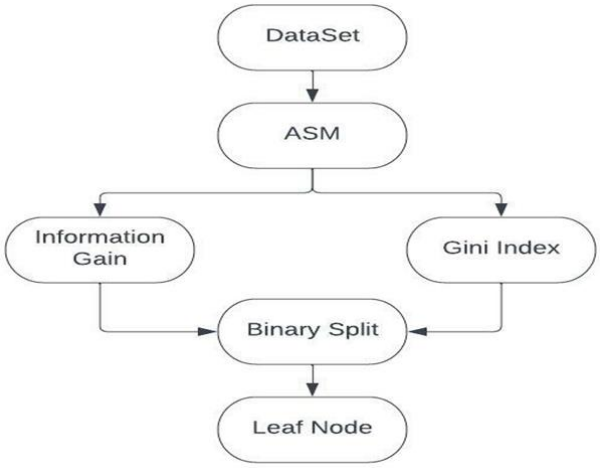


Fig.4.6 Decision Tree

In a decision tree for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and based on the comparison follows the branch and jumps to the next node. For the next node the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm.

PROCESS

Step-1: Begin the tree with the root node S which contains the complete dataset.

Step-2: Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where cannot further classify the nodes and called the final node as a leaf node.

Attribute Selection Measures

Implementing a Decision tree select the best attribute for the root node and for sub-nodes. By **Attribute selection measure** this measurement select the best attribute for the nodes of the tree. Techniques for ASM which are

- **Information Gain**
- **Gini Index**

Information Gain

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides about a class. According to the value of information gain, split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula.

Information Gain = Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as.

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

Gini Index

Gini index is a measure of impurity or purity used while creating a decision tree in the CART algorithm. An attribute with the low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits. Gini index can be calculated using the below formula:

$$\text{Gini Index} = 1 - \sum_j P_j^2$$

Decision Tree is simple to understand as it follows the same process which a human follow while making any decision in real-life. It can be very useful for solving decision-related problems. It helps to think about all the possible outcomes for a problem. There is less requirement of data cleaning compared to other algorithms. The decision tree contains lots of layers, which makes it complex. It may have an overfitting issue, which can be resolved using the Random Forest algorithm**.** For more class labels, the computational complexity of the decision tree may increase. The implementation for this algorithm is explained in chapter 5.1.7

4.4 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for Classification problems in ML. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output shown in fig.4.7.

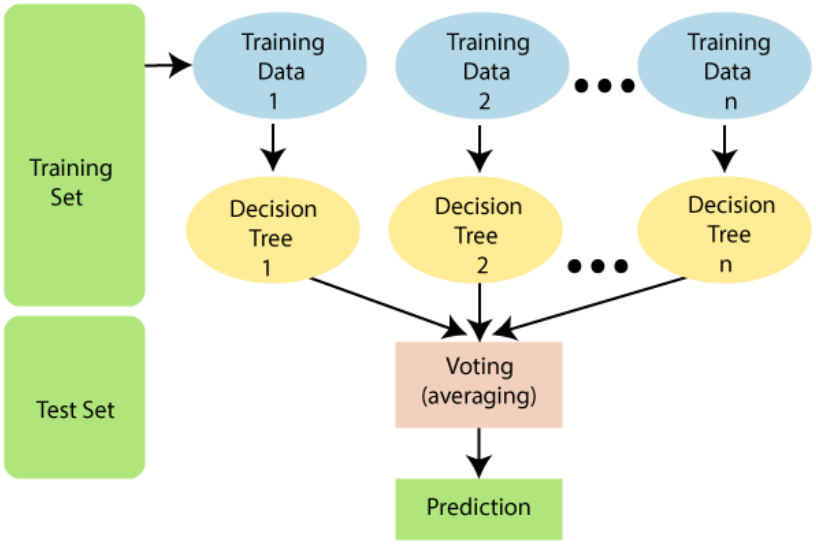


Fig.4.7 Random Forest

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Random Forest is a classifier consisting of a collection of tree-structured classifiers {h(x, Θk) k=1, 2, ….}, where the {Θk } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x. Random Forest generates an ensemble of decision trees. To achieve diversity among base decision trees, The randomization approach which works well with bagging or random subspace methods. To generate each single tree in Random Forest following steps are done, If the number of records in the training set is N, then N records are sampled at random but with replacement from the original data this is bootstrap sample. This sample will be the training set for growing the tree. If there are M input variables, a number m << M is selected such that at each node m variables are selected at random out of M and the best split on these m attributes is used to split the node. The value of m is held constant during forest growing. Each tree is grown to the largest extent possible.

Accuracy of Random Forest

The Generalization error (PE*) of Random Forest

$$PE^* = P_{x,y}\,(mg(X,Y)) < 0$$

- mg(X,Y) is Margin function.

The Margin function measures the extent to which the average number of votes at (X, Y) for the right class exceeds the average vote for any other class. If X is the predictor vector and Y is the classification. The generalization error of ensemble classifier is bounded by a function of mean correlation between base classifiers and their average strength (s. If ρ is mean value of correlation, an upper bound for generalization error is

$$PE^* \leq \rho\,(1 - s^2) / s2$$

If the number of cases in the training set is N, then sample of N cases is taken at random but with replacement. This sample will be the training set for growing the tree. If there are M input variables, a number m<<M is specified such that at each

node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing. Each tree is grown to the largest extent possible. There is no pruning.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process of Random Forest Algorithm.

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes**.**

Random Forest is capable of handling large datasets with high dimensionality . It enhances the accuracy of the model and prevents the overfitting issue. The implementation for this algorithm is explained in chapter 5.1.7

4.5 KNN Algorithm

K nearest neighbors stores all available cases and classifies new cases by a majority vote of its k neighbors shown in fig.4.8. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function. These distance functions can be Euclidean and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbour. At times, choosing K turns out to be a challenge while performing KNN modeling.

WORKING K-NN ALGORITHM

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Model is ready.

A new data point need to put it in the required category. Choose the number of neighbors, k=5. Calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points. By calculating the Euclidean distance the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. The 3 nearest neighbors are from category A, this new data point must belong to category A.

KNN algorithm is robust to the noisy training data and can be more effective if the training data is large. Determine the value of K which may be complex some time. The computation cost is high because of calculating the distance between the data points for all the training samples. The implementation for this algorithm is explained in chapter 5.1.7

4.6 LOGISTIC REGRESSION

Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables. The independent variables can be nominal, ordinal, or of interval type. The logistic function is also known as the sigmoid function. The value of this logistic function lies between zero and one.

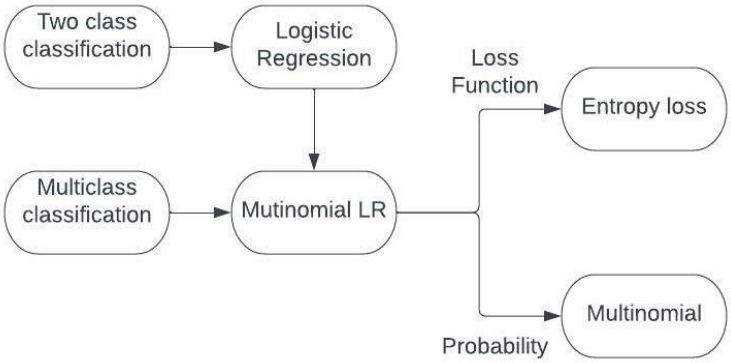


Fig.4.8 Logistic Regression

4.6.1 LOGISTIC FUNCTION

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function shown in fig.4.9. In logistic regression, the threshold value which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

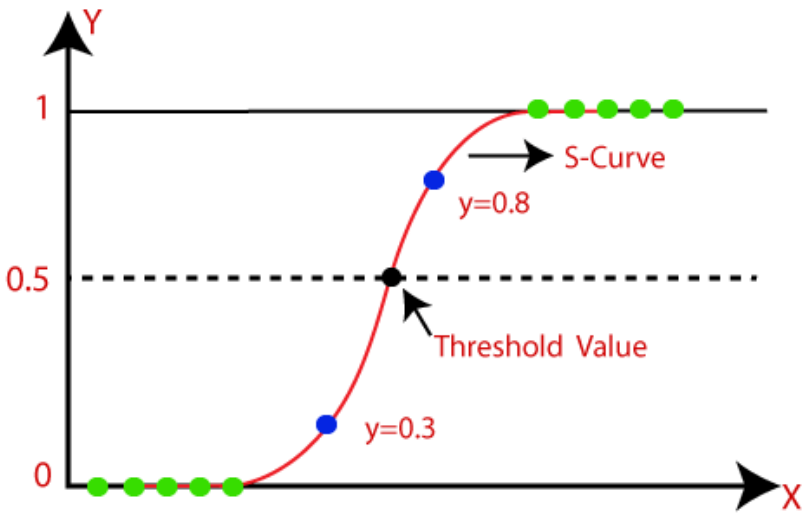


Fig.4.9 S-FORM CURVE

Logistic Regression Equation

The Logistic regression equation can be obtained from the Linear Regression equation.

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

Equation Of The Straight Line:

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

In Logistic Regression y can be between 0 and 1.

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

Range between -[infinity] to +[infinity], Take logarithm of the equation it will become

Logistic Regression is less prone to over-fitting. Logistic Regression proves to be very efficient when the dataset has features that are linearly separable. The presence of data values that deviate from the expected range in the dataset may lead to incorrect results as this algorithm is sensitive to outliers. Logistic Regression requires a large dataset and also sufficient training needed for all the categories it needs to identify. Only important and relevant features should be used to build a model otherwise the probabilistic predictions made by the model may be incorrect and the model's predictive value may degrade. It is difficult to capture complex relationships using logistic regression. The implementation for this algorithm is explained in chapter 5.1.7

CHAPTER 5

IMPLEMENTATION

5.1 SOFTWARE ENVIRONMENT

5.1.1 GOOGLE COLABORATORY

Google Colaboratory is a free Jupyter notebook environment running wholly in the cloud to write and execute code in Python. Colab notebooks execute code on Google's cloud servers, meaning we can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of our machine. Google Colab is a very simple and powerful platform to run your Machine Learning and Deep Learning models without worrying about GPU power. Google Colab gives us about 13 GB of RAM and 25 GB of storage for free. Save notebooks, datasets from/to Google Drive.

5.1.2 PROGRAMMING LANGUAGE USED

- Python

5.1.3 LIBRARIES USED

- NumPy
- Pandas
- Scipy
- Seaborn
- Matplotlib
- Sklearn
- GeoPandas

5.1.4 DATASET USED

Crime Data provided by Chicago Police Department

Table.5.1 Chicago Crime Dataset

Variable	Description
ID	Unique identifier for the record.
Case Number	The Chicago Police Department RD Number (Records Division Number), which is unique to the incident.
Date	Date when the incident occurred. this is sometimes a best estimate.
Block	The partially redacted address where the incident occurred, placing it on the same block as the actual address.
IUCR	The Illinois Uniform Crime Reporting code. This is directly linked to the Primary Type and Description.
Primary Type	The primary description of the IUCR code.
Description	The secondary description of the IUCR code, a subcategory of the primary description.
Location Description	Description of the location where the incident occurred
Arrest	Indicates whether an arrest was made.
Domestic	Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act.
Beat	Indicates the beat where the incident occurred. A beat is the smallest police geographic area – each beat has a dedicated police beat car. Three to five beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts.
District	Indicates the police district where the incident occurred.
Ward	The ward (City Council district) where the incident occurred.
Community Area	Indicates the community area where the incident occurred. Chicago has 77 community areas.
FBI Code	Indicates the crime classification as outlined in the FBI's National Incident-Based Reporting System (NIBRS).
X Coordinate	The x coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
Y Coordinate	The y coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
Year	Year the incident occurred.
Updated On	Date and time the record was last updated.
Latitude	The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Longitude	The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Location	The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block.

Chicago crime dataset attributes information is shown in Table.5.1

5.1.5 DATA PREPROCESSING

In this part,clean up the crime dataset obtained from the city of Chicago data portal. The most recent data from year of 2016 to 2022 from the Chicago crime dataset, instead of using the whole dataset which contains 6 million records back to year 2001. Not to use earlier data because the distribution of crimes can change over time due to various reasons, and the most recent 5 to 7 years will be more relevant to the current situation, view the problem as a classification problem.

Step 1. Excluded records with vague information: For example "others" or typo or NA. some unnecessary columns are also removed.

Step 2. Removed extremely rare crime types and locations : Classify every single type of crime, it is not really practical since some crimes are happening at extremely low frequency. Based on the sorted list of crimes, some crimes are really rare, for example human trafficking. Though these rare records won't greatly affect our model, Picked top 25 types of crime for simplicity. Build a classification (yes-or-no) model to determine whether a crime is going to be severe or not, Classified the following types of crimes as severe and give them an indicator variable 1 : "Arson", "Assault", "Battery", "Crime Sexual Assault", "Criminal Damage", "Criminal Trespass", "Homicide", "Robbery". In this case, severe means only the crime that involves direct violence, and everyone near the crime scene needs to use caution immediately. Another important feature for classification is "Location Description". Imagine robbery is more likely to happen on a street compared with a restaurant, while deceptive practice is probably going to happen at a store instead of at home. To avoid the curse of dimensionality , dropped crimes that happen at extremely unlikely locations, and only kept crimes that happen at the top 25 locations. Even after two rounds of filtering, still kept around 85% of the crimes.

Step 3. Converted categorical features to dummy (indicator) variables for classification : Checked some other features that can be used for the classifier. Police district and community area are the two features that can be important, expect bad neighborhood may have higher chance to have crimes that are more violent. 77 community areas, and 22 districts. This is reasonable because one

police district can cover multiple areas. In order to do classification, it is preferable to convert the categorical data into dummy variable (0 and 1). Converted crime type, police district, and location description into dummy variables.

Step 4. Extracted the time information and converted categorical features to dummy variables : For each crime time record , Segmented the time into eight blocks of 3-hour, for example 0-3am, 3-6am, etc and created indicator variable based on that. In the data cleanup step, saved both the exact information and the "binned" information to use it later.

Step 5. Added an extra feature "distance to closest police station" : Calculated an additional feature using the coordinate info the distance of the location to the closest police station. it is not very likely that a severe crime like robbery is going to happen right in front of a police station. Since the longitude / latitude data of each crime, calculated distances between the crime scene and all the police stations in Chicago and determined the distance of the crime to the closest police station.

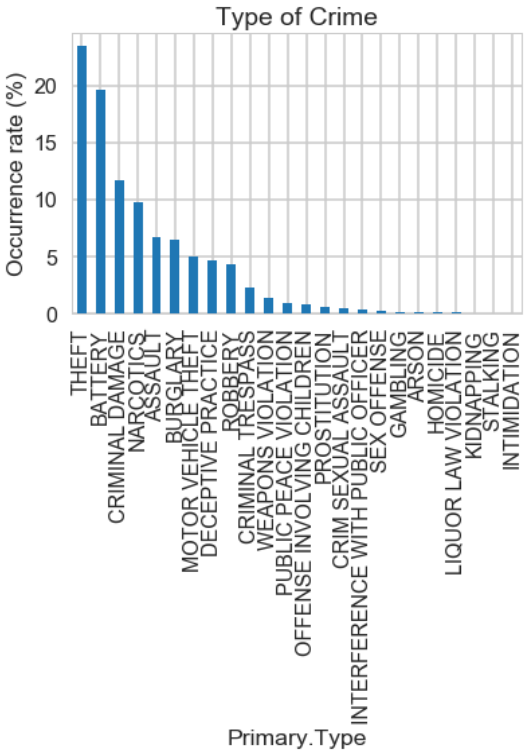
Step 6. Integrated socioeconomic status of the neighborhood where the crime happened : Given a location, there are other relevant information such as income, education level and population structure of the neighborhood.Included several columns of continuous variables describing income, housing condition, education level, population structure based on the "Community Area". The data is also provided by the government of Chicago. To avoid the duplication, police district and not the community area as a categorical feature.

5.1.6 DATA EXPLORATION

Visualizing the data using different ways. The analyzed trend of crime occurrence for each year Then, the plotted crime occurrence rates of the following crime type, scene of crime, hour-day-month of crime .This gives a better understanding of the major crimes that occur. The interpret that theft has the highest percentage and is the crime type with the highest crime rate. This also identified locations that are more prone to crimes, street being the scene with

highest crime rate. The crime rate with respect to time of day, day of week and month, depicts which crime happens the most and at what time. Observation says that higher crime intensity is between the duration 12:00PM – 18:00PM. To dive deeper into the data, The focused on only the 4 major types of crimes and grouped the rest into others, that gave us 5 broad categories of crime types. Breakdown of the crimes by location of crime scene, hour of the day, weekday and month as depicted ,to see if the can spot any trends. The plotted normalized crime types for each category.

The graph shows the geographical distribution of the crimes in the city of Chicago. The intensity of the color shows the number of crimes. The higher the intensity and higher is the number of crime in that particular area of City of Chicago. This could give a quick insight on which region in the City of Chicago ranks the highest in Crime Rates. The graph is plotted based on the latitude and longitude of the crimes in ward wise is shown in fig.5.10 , crimes in community area wise is shown in fig.5.11, crimes in District wise is shown in fig.5.12 .



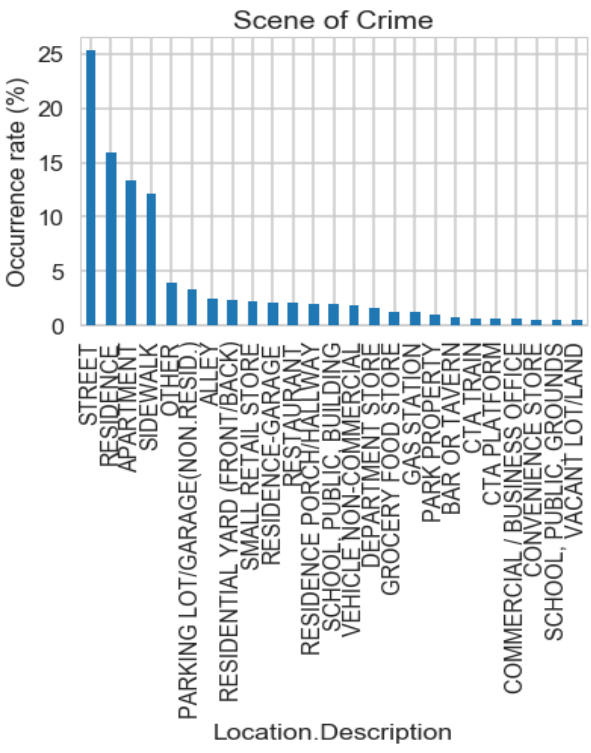


Fig.5.1.Crime Occurrence Rate V/s Crime Type

Fig.5.2.Crime Occurrence Rate V/s Scene of Crime

The bar graph shown in fig.5.1 is plotted based on the Crime Occurrence rate and Crime Type. The Primary type attribute in the dataset holds the crime type occurred in that crime . This graph is plotted to interpret the highest occurrence of crime types. The bar graph shown in fig.5.2 is plotted based on the Crime Occurrence rate and Crime Location. The Location Description attribute in the dataset holds the Locations where occurred in that crime. This graph is plotted to interpret the highest occurrence of crime in Location.

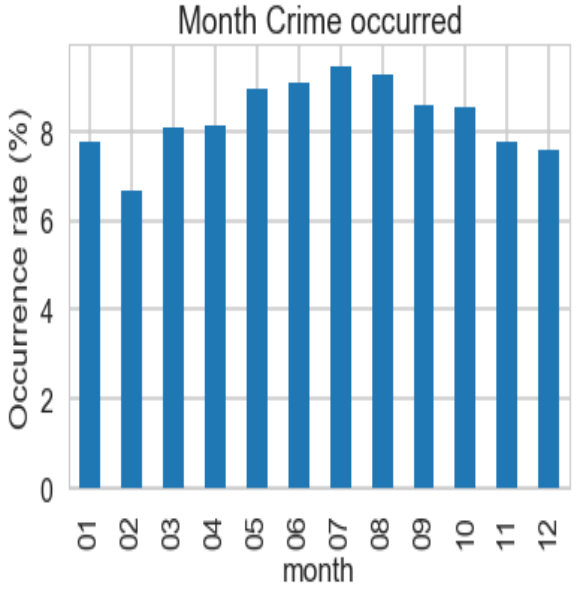


Fig.5.3. Crime occurrence rate V/s Time of Crime

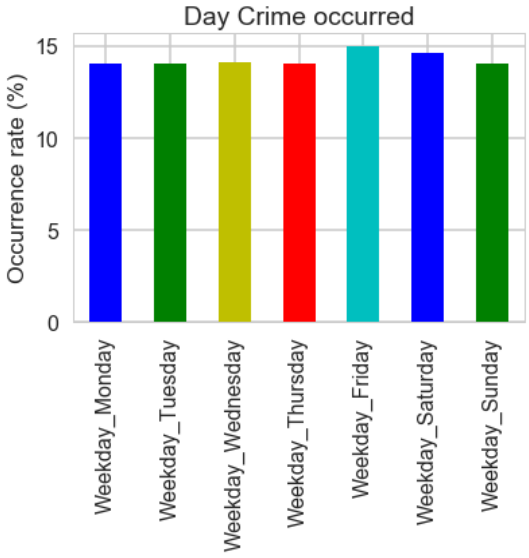


Fig.5.4. Crime Occurrence Rate V/s Day of Crime

The bar graph shown in fig.5.3 is plotted based on the Crime Occurrence rate and Crime occurred over the months of the year. This graph is plotted to interpret the highest occurrence of crime happened during which month. The bar graph shown in fig.5.4 is plotted based on the Crime Occurrence rate and Crime occurrence over the days of the week. This graph is plotted to interpret the highest occurrence of crime in Days.

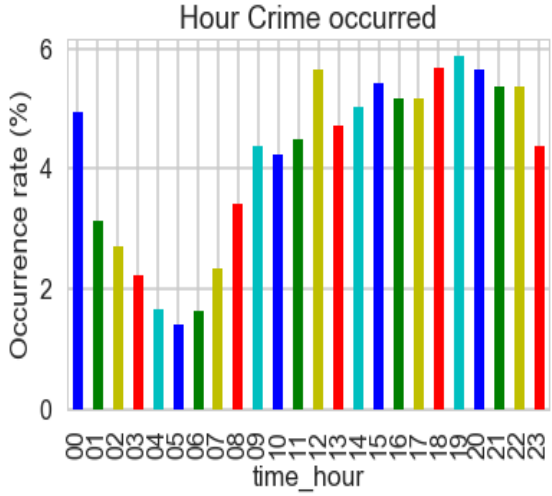


Fig.5.5. Crime occurrence rate V/s Month of Crime

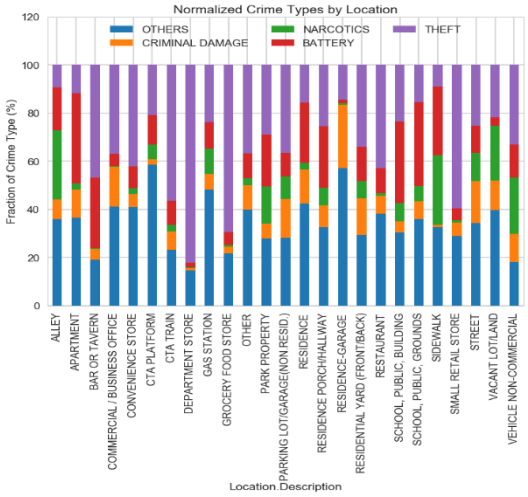


Fig.5.6 Normalized Crime Types by Location

The bar graph shown in fig.5.5 is plotted based on the Crime Occurrence rate and Crime occurred over the hours of the day. This graph is plotted to interpret the highest occurrence of crime happened during which hours of the day. The bar graph shown in fig.5.6 is plotted based on the Top four highest occurred Crime and Crime occurrence over the Locations. This graph is plotted to interpret the highest occurrence of crime type happened during that specific location among the four crimes. The bar graph shown in fig.5.7 is plotted based on the Top four highest occurred Crime and Crime occurrence over the hours of the day. This graph is plotted to interpret the highest occurrence of crime type happened during that specific hour among the four crimes. The bar graph shown in fig.5.8 is plotted based on the Top four highest occurred Crime and Crime occurrence over the days of the week. The bar graph shown in fig.5.9 is plotted based on the Top four highest occurred Crime and Crime occurrence over the months of the year.

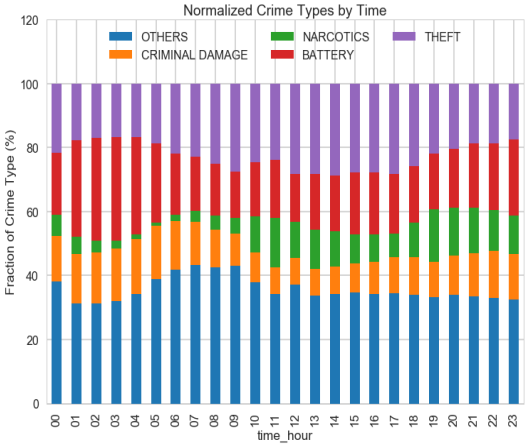


Fig.5.7 Normalized Crime Types by Time

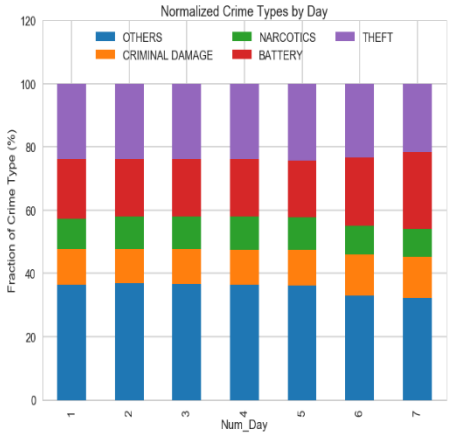


Fig.5.8. Normalized Crime Types by Day

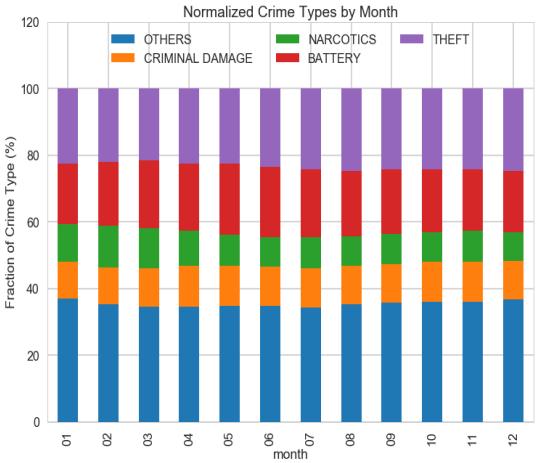


Fig.5.9.Normalized Crime Types by Month

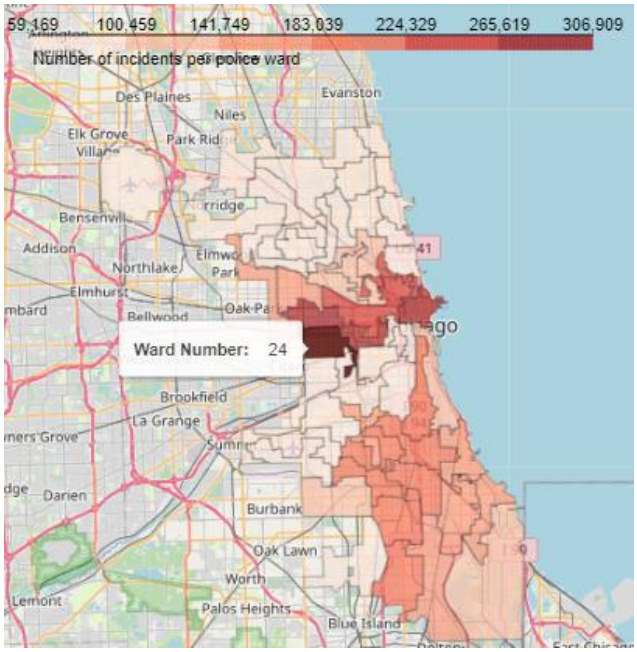


Fig.5.10. Ward Wise Crime Intensity

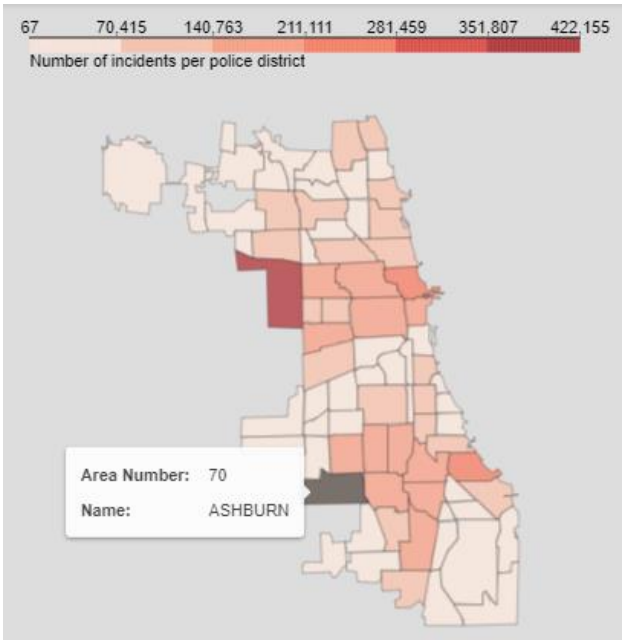


Fig.5.11. Community Area Wise Crime Intensity

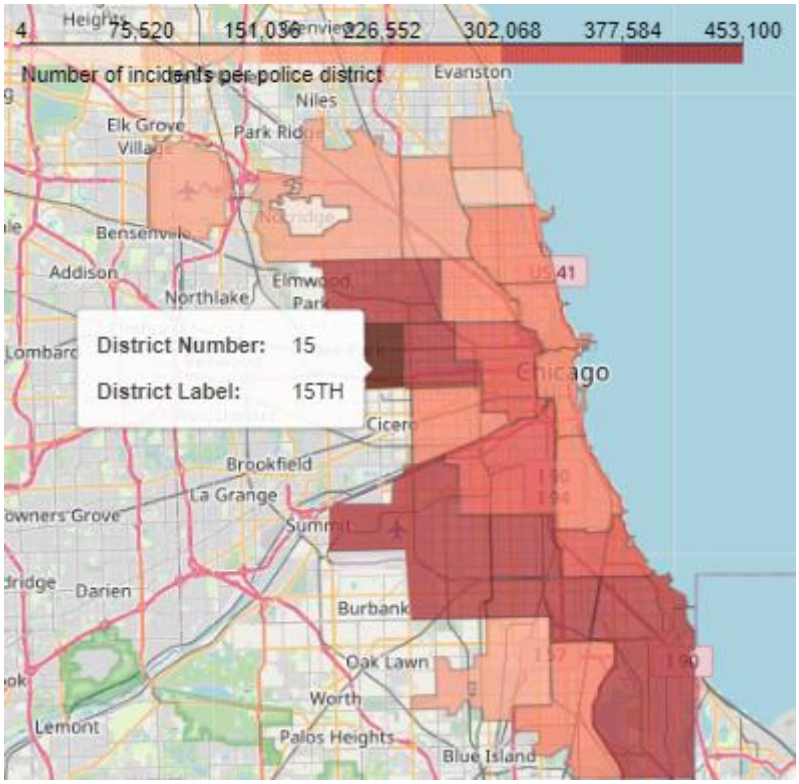


Fig.5.12.District Wise Crime Intensity

5.1.7 DATA MODELING

In data modeling tried to build two types of model, **Binary Classification Model** for predicting whether a crime is severe or not, **Multi-Class Classification Model** to determine the exact type of crime.

5.1.7.1 BINARY CLASSIFICATION

In binary classification to predict severity of the crime, so dropped the exact crime type for that point. These are the features that required normalization Latitude, Longitude, closest_station. Other features for our dataset are all categorical data and have all been converted to dummy variables (District, Time_block, Weekday, Location description). The performed split on the 80000 sample records, into training set and test set, and normalized by themselves. Then the performed exploratory data analysis to see what features can be important.

The latitude and longitude information may be helpful for classification. For example, if the latitude is low, it's more likely that severe crime will happen. This is consistent with the fact that the safety at south Chicago is notoriously bad. The economic, unemployment, age status do provide expected result. For example, for regions with lower income and higher unemployment rate, the crimes are going to be more severe. The explored all 60 indicator variables then have to see whether any of them will be good to judge whether a crime is severe or not. From all of those, some indicator variables are good to for classification.

For example, if the time block is 3am to 6am, the proportion of severe crime is obviously higher, which is not the case in time block 9am to 12pm. If the crime takes place in an apartment or house, it's more likely to be a severe crime, while if the crime takes place on a street where everyone can see, it's less likely going to be severe.

MODEL GENERATION

LOGISTIC REGRESSION CLASSIFIER

Binary classifier model which can predict the severity of the given instance using Logistic Regression . the dataset is divided into two portion testing dataset and training dataset. Training dataset contains all features along with the target

label. Testing dataset only contains the features from which a machine learning model predicts the target label. The ‘liblinear’ solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty. C is a parameter passed for Inverse of regularization strength, must be a positive float, smaller values specify stronger regularization.

```
from sklearn.linear_model import LogisticRegression
clflog = LogisticRegression(C=1, penalty='l1', solver='liblinear')
clflog, Xtrain, ytrain, Xtest, ytest, confclflog, training_accuracy, test_accuracy=do_classify(clflog, {"C": [0.001, 0.01, 0.1, 1.0, 10.0, 20.0, 40.0,
confusion_dict["Logistic"]=confclflog
model_dict["Logistic"]=clflog
accuracy_dict["Logistic"]=training_accuracy
accuracy_dict1["Logistic"]=test_accuracy
```

```
Training accuracy: 0.62
Test accuracy:     0.62
[[7508 4790]
 [4404 7298]]
LogisticRegression(penalty='l1', solver='liblinear')
```

Fig.5.13 Logistic regression binary classifier for Binary Class

Using the Testing data trained the binary classifier for the target class and the created binary classifier model is tested with test data for measuring the performance of the Logistic regression binary classifier model. Training and testing results of the model is shown in the table.5.2

Table.5.2 Logistic regression binary classifier for Binary Class

Training Accuracy	Testing Accuracy
62%	62%

SUPPORT VECTOR MACHINE CLASSIFIER

LinearSVC class capable of performing binary classification on a dataset. Use of the hinge loss , directly optimized by LinearSVC. SVM models use C as regularization parameter. The exact equivalence between the amount of regularization of two models depends on the exact objective function optimized by the model.

```
from sklearn.svm import LinearSVC
clfsvm=LinearSVC(loss="hinge")
clfsvm, Xtrain, ytrain, Xtest, ytest, confclfsvm, training_accuracy, test_accuracy= do_classify(clfsvm, {"C": [0.001, 0.01, 0.1, 1.0, 10.0, 50, 100.0
confusion_dict["svm"]=confclfsvm
model_dict["svm"]=clfsvm
accuracy_dict["svm"]=training_accuracy
accuracy_dict1["svm"]=test_accuracy
```

```
Training accuracy: 0.59
Test accuracy:     0.59
[[7139 5089]
 [4713 7059]]
LinearSVC(C=0.1, loss='hinge')
```

Fig.5.14 Support vector machine classifier for Binary Class

The Binary classifier SVM model is trained and tested with the dataset results of the training and testing dataset is shown in the table.5.3

Table.5.3 Support vector machine classifier for Binary Class

Training Accuracy	Testing Accuracy
59%	59%

DECISION TREE CLASSIFIER

A Binary Decision Tree is a structure based on a sequential decision process.Starting from the root, an input is evaluated and one of the two branches is selected.The root node is selected by calculating the entropy and information gain of the given attributes in the dataset.The attribute having maximum information gain was chosen as root node.For binary classification the given case number is identified the severity of the crime.Fig.5.15 shows the Decision Tree Classifier for binary classification.

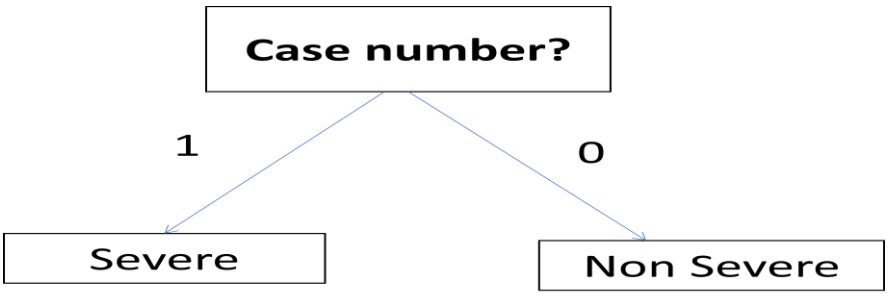


Fig.5.15 Decision Tree for Binary Classification

The hyperparameter max_depth controls the overall complexity of a decision tree. This hyperparameter allows to get a trade-off between an under-fitted and over-fitted decision tree. This parameter is adequate under the assumption that a tree is built symmetrically. Tree will be symmetrical and sometimes it will result in unsymmetrical trees. Optimal generalization performance could be reached by growing some of the branches deeper than some others. The max_depth is the hyperparameters that should optimize via cross-validation and grid-search.

```
from sklearn.tree import DecisionTreeClassifier
clfdt=DecisionTreeClassifier()
clfdt, Xtrain, ytrain, Xtest, ytest, confclfdt, training_accuracy, test_accuracy = do_classify(clfdt, {"max_depth":np.arange(1,20,2)}, data, total_fe
confusion_dict["decision tree"]=confclfdt
model_dict["decision tree"]=clfdt
accuracy_dict["decision tree"]=training_accuracy
accuracy_dict1["decision tree"]=test_accuracy
```

```
Training accuracy: 0.62
Test accuracy:     0.62
[[9455 2843]
 [6317 5385]]
DecisionTreeClassifier(max_depth=7)
```

Fig.5.16 Decision Tree Classifier for Binary Class

. The Decision Tree Binary classifier for the severe crime area , trained and tested the results is shown in the table .5.4

Table.5.4 Decision Tree Classifier for Binary Class

Training Accuracy	Testing Accuracy
62%	62%

Prediction Of Binary Classification

By given the case number as input, binary classification uses the Decision Tree algorithm to predict whether the case is severe or not severe. Output of the prediction came out to be 0 or 1 . The Output with 0 represent the prediction of the case which is non severe . The Output with 1 represent the prediction of the case which is severe .

For the given case number 102438 the output prediction is severe.

```
features = [[102438]]
prediction = clfdt.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [1]
```

Fig.5.17 Severe crime prediction

For the given case number 105797 the output prediction is non severe.

```
features = [[411648]]
prediction = clfdt.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [0]
```

Fig.5.18 Non Severe crime prediction

RANDOM FOREST CLASSIFIER

Random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
from sklearn.ensemble import RandomForestClassifier
randf=RandomForestClassifier()
clfrdf, Xtrain, ytrain, Xtest, ytest, confrdf, training_accuracy, test_accuracy=do_classify(randf, {"n_estimators":[10, 20, 30, 40, 100]}, data, tota
confusion_dict["Random forest"]=confrdf
model_dict["Random forest"]=clfrdf
accuracy_dict["Random forest"]=training_accuracy
accuracy_dict1["Random forest"]=test_accuracy
```

```
Training accuracy: 1.00
Test accuracy:     0.60
[[7614 4684]
 [4805 6897]]
RandomForestClassifier()
```

Fig.5.19 Random Forest Classifier for Binary Class

The training and testing dataset is passed to the classifer and the result of Random Forest classifier is shown in Table.5.5.

Table.5.5 Random Forest Classifier for Binary Class

Training Accuracy	Testing Accuracy
100%	60%

K-NEAREST NEIGHBOUR

K is the number of voters that the algorithm consult to make a decision about to which class a given data point it belongs to. Cross-validation is when the dataset is randomly split up into 'k' groups. One of the groups is used as the test set and the rest are used as the training set. The model is trained on the training set and scored on the test set. Then the process is repeated until each unique group as been used as the test set.

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier()
neigh, Xtrain1, ytrain1, Xtest1, ytest1, confknn, training_accuracy, test_accuracy=do_classify(neigh, {"n_neighbors":[5, 10, 20, 40]}, data, total_fea
confusion_dict["KNN"]=confknn
model_dict["KNN"]=neigh
accuracy_dict["KNN"]=training_accuracy
accuracy_dict1["KNN"]=test_accuracy
```

```
Training accuracy: 0.64
Test accuracy:     0.61
[[7741 4557]
 [4878 6824]]
KNeighborsClassifier(n_neighbors=40)
```

Fig.5.20 KNN Classifier for Binary Class

The K-Nearest Neighbor for the severe crime area , trained and tested the results of KNN Classifier is shown in the table.5.6.

Table.5.6 KNN Classifier for Binary Class

Training Accuracy	Testing Accuracy
64%	61%

To predict severity of the crime, so dropping the exact crime type for that point. These are the features that required normalization: Latitude, Longitude, closest_station. Other features for our dataset are all categorical data and have all been converted to dummy variables (District, Time_block, Weekday, Location

description). The performed split on the 80000 sample records, into training set and test set, and normalized by themselves. Then performed exploratory data analysis to see what features can be important.

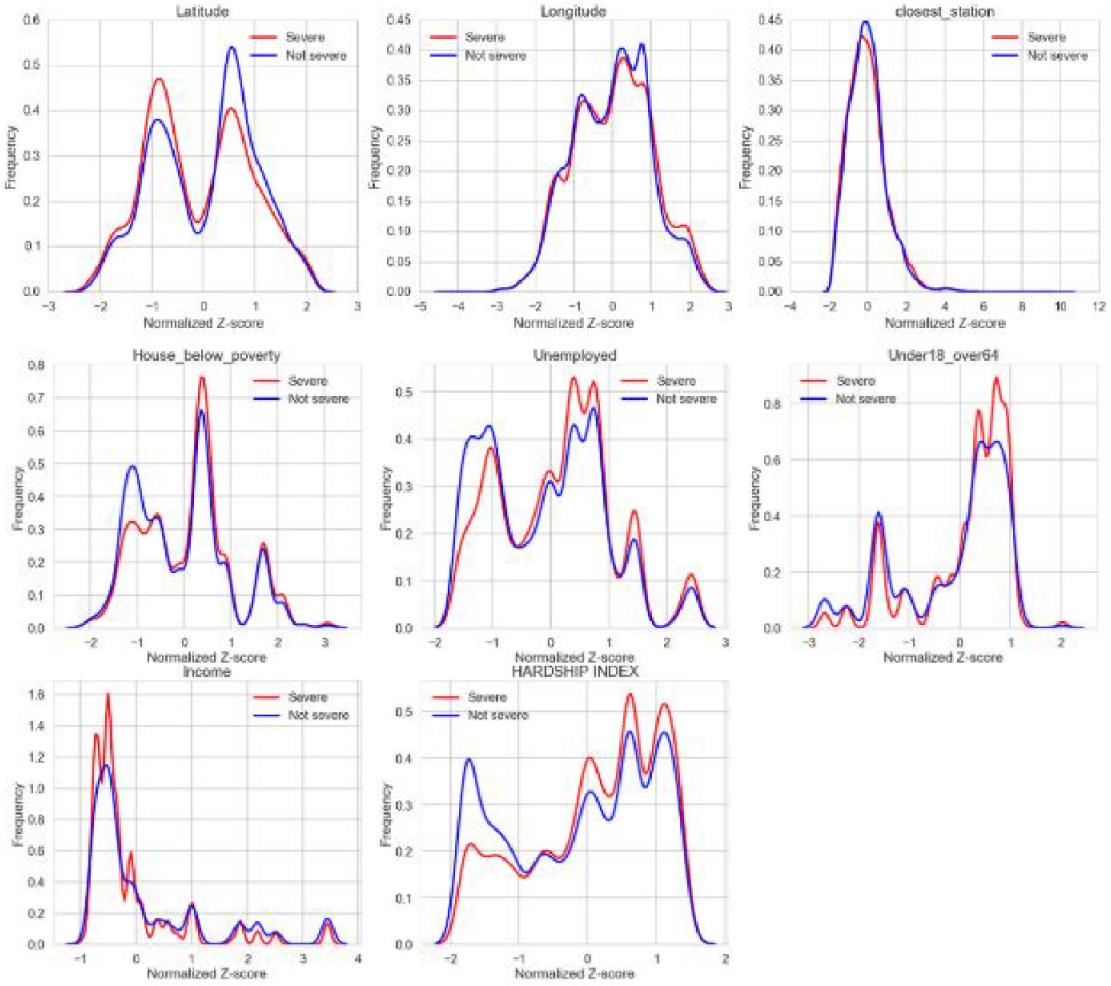


Fig.5.21 Analysis of Continuous Features

The latitude and longitude info may be helpful for classification. For example, if the latitude is low, it's more likely that severe crime will happen. This is consistent with the fact that the safety at south Chicago is notoriously bad. The economic, unemployment, age status do provide expected result. For example, for regions with lower income and higher unemployment rate, the crimes are going to be more severe. Then explored all 60 indicator variables that have to see whether any of them will be good to judge whether a crime is severe or not. From all of those, that could see some indicator variables are good to for classification.

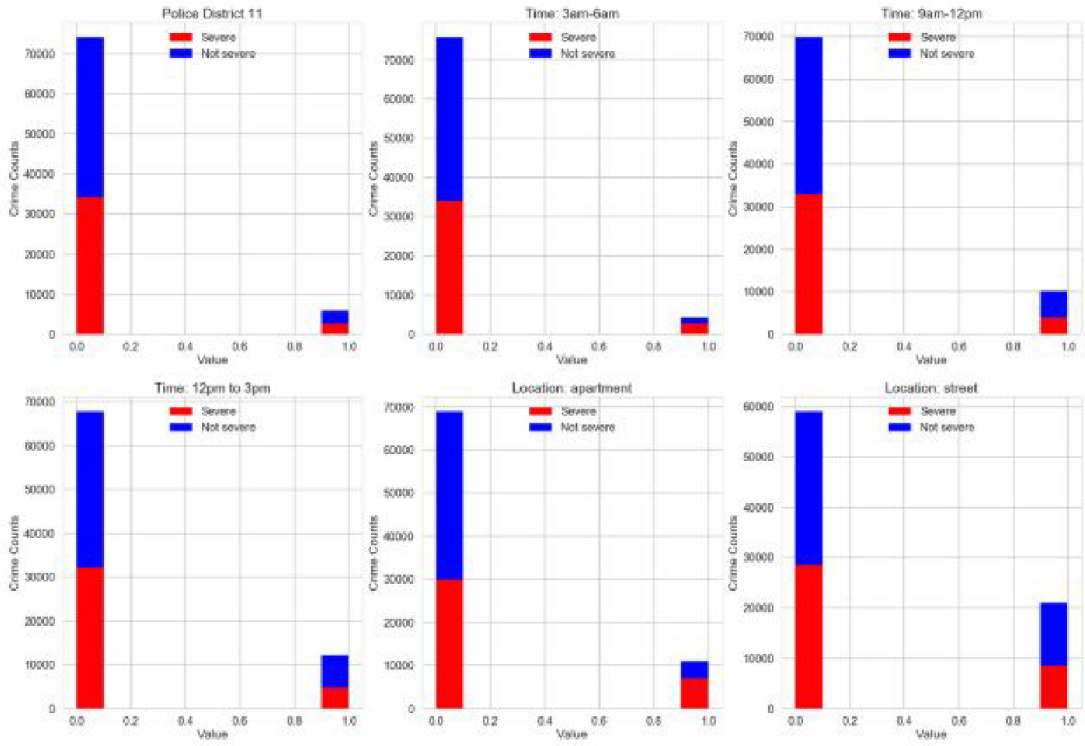


Fig.5.22 Analysis of Indicator Variables

For example, if the time block is 3am to 6am, the proportion of severe crime is obviously higher, which is not the case in time block 9am to 12pm. If the crime takes place in an apartment or house, it's more likely to be a severe crime, while if the crime takes place on a street where everyone can see, it's less likely going to be severe.

5.1.7.2 MULTI CLASS CLASSIFICATION

The goal of multi class classification is to classify each record into specific type of crimes. The first step is to clean up the data as the response variable specifically needs to be transformed into numpy array. Another very important thing is that have 25 classes of crimes, but the computational model does not allow us to do multiclass classification at this scale. Therefore, here only picked up top four types of crimes. The only maintained these types of crimes: THEFT, BATTERY, CRIMINAL DAMAGE, NARCOTICS, Splitting the data into training and testing sets and build the models for multiclass classification.

LOGISTIC REGRESSION CLASSIFIER

Multi classifier model which can predict the severity of the given instance using Multinomial Regression. Using the Testing data trained the multi classifier for the target class and the created multi classifier model is tested with test data for measuring the performance of the Logistic regression multi classifier model**.**

```
from sklearn.linear_model import LogisticRegression
clflogmulti=LogisticRegression(penalty="l2",multi_class='multinomial',solver="newton-cg",max_iter=100)
clflogmulti, Xtrain, ytrain, Xtest, ytest, training_accuracy, test_accuracy=do_classify2(clflogmulti, {"C": [0.001, 0.01, 0.1, 1.0, 10.0, 100.0]}, da
accuracy_multi_train["Logistic - newton cg"]=training_accuracy
accuracy_multi_test["Logistic - newton cg"]=test_accuracy
```

```
Training accuracy: 0.55
Test accuracy:     0.56
[[3961 1233  398   81]
 [1191 3470  272  189]
 [1239 1102  493   79]
 [ 363  350  167  438]]
LogisticRegression(C=0.1, multi_class='multinomial', solver='newton-cg')
```

Fig.5.23 Logistic regression classifier for MultiClass.

Training and testing results of the model is shown in the table.5.7.

Table.5.7 Logistic regression classifier for MultiClass

Training Accuracy	Testing Accuracy
55%	56%

Prediction Of Multi Class Classification

By given the case number as input, multi classification uses the Logistic Regression algorithm to predict whether the case is which type of crime.

For the given case number 102012 the output prediction is Theft.

```
features = [[102012]]
prediction = clflogmult.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [0.]
```

Fig.5.24 Theft Crime Prediction

For the given case number 102438 the output prediction is Battery.

```
features = [[102438]]
prediction = clflogmult.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [1.]
```

Fig.5.25 Battery Crime Prediction

For the given case number 102509 the output prediction is Criminal damage.

```
features = [[102509]]
prediction = clflogmult.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [2.]
```

Fig.5.26 Criminal damage Crime Prediction

For the given case number 102355 the output prediction is Narcotics.

```
features = [[102355]]
prediction = clflogmult.predict(features)
print("Prediction: {}".format(prediction))
```

```
Prediction: [3.]
```

Fig.5.27 Narcotics Crime Prediction

DECISION TREE CLASSIFIER

The root node is selected by calculating the entropy and information gain of the given attributes in the dataset.The attribute having maximum information gain was chosen as root node.For multiclass classification the given case number is identified among the four crime types.

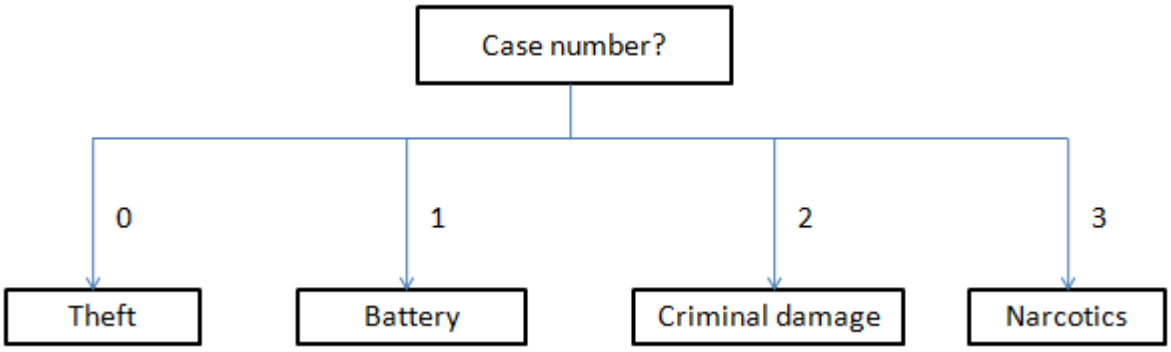


Fig.5.28 Decision Tree Classifier for Multiclass Classification

The hyperparameter max_depth controls the overall complexity of a decision tree. This hyperparameter allows to get a trade-off between an under-fitted and over-fitted decision tree. This parameter is adequate under the assumption that a tree is built symmetrically. Tree will be symmetrical and sometimes it will result in unsymmetrical trees. Optimal generalization performance could be reached by growing some of the branches deeper than some others. The max_depth is the hyperparameters that should optimize via cross-validation and grid-search.

```
from sklearn.tree import DecisionTreeClassifier
clfdt_multi=DecisionTreeClassifier()
clfdt_multi, Xtrain, ytrain, Xtest, ytest, training_accuracy, test_accuracy=do_classify2(clfdt_multi, {"max_depth":np.arange(1,20,2)}, data2, total_f
accuracy_multi_train["Decision tree multiclass"]=training_accuracy
accuracy_multi_test["Decision tree multiclass"]=test_accuracy
```

```
Training accuracy: 0.55
Test accuracy:     0.54
[[3782 1142  667   82]
 [1221 3205  488  208]
 [1069 1067  700   77]
 [ 298  299  316  405]]
DecisionTreeClassifier(max_depth=7)
```

Fig.5.29 Decision Tree Classifier for MultiClass

The Decision Tree for Multiclass classification trained and tested the results is shown in the table .5.8

Table.5.8. Decision Tree Classifier for MultiClass

Training Accuracy	Testing Accuracy
55%	54%

RANDOM FOREST CLASSIFIER

Random forests combine the predictions of multiple decision trees.Random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
from sklearn.ensemble import RandomForestClassifier
randfmulti=RandomForestClassifier()
randfmulti, Xtrain, ytrain, Xtest, ytest, training_accuracy, test_accuracy=do_classify2(randfmulti, {"n_estimators":[10, 20, 30, 40, 100]}, data2, to
accuracy_multi_train["Random forest multiclass"]=training_accuracy
accuracy_multi_test["Random forest multiclass"]=test_accuracy
```

```
Training accuracy: 1.00
Test accuracy:     0.52
[[3771 1232  555  115]
 [1279 3047  555  241]
 [1172 1049  590  102]
 [ 289  411  156  462]]
RandomForestClassifier()
```

Fig.5.30 Random Forest Classifier for MultiClass

The training and testing dataset is passed to the classifer and the result is shown in Table.5.9.Random Forest Result.

Table.5.9 Random Forest Classifier for MultiClass

Training Accuracy	Testing Accuracy
100%	52%

5.1.7.3 BI-DIRECTIONAL LONG SHORT TERM MEMORY WITH ATTENTION

After the Dataset preprocessing, the required important features which are contributing lot in predicting the crime occurrence is taken . Multi class classification of crime is done with BI-LSTM with Attention. Model takes two inputs(one for each bi-LSTM) crime matrix and anomaly matrix, Both are feed to Bi-LSTMs, whose outputs are combined(based on weights, which themselves are learned, to assign importance to these inputs) and passed to another Bi-LSTM layer. The output from this Bi-LSTM is fed to an attention layer, which basically captures and assigns importance to the features of this output. This is fed to a 2-layer MLP, which predicts the occurrence of crime class j, for region i at day T+1, given the historical data for the previous T days. The ground truth here is the actual crime occurence data for that day. The loss function is sigmoid cross-entropy with logits.

BUILDING THE MODEL

For building the model, Bi-LSTM a base model, For LSTM, added a dense layer for classification. In the dense layer we use Softmax as the activation function. The Softmax activation function gives probability value as output, based on the probability value classifies the crime instance among the four crime classes.

TRAINING THE MODEL

Once the model is built, train the model using the pre processed dataset .To train the model randomly initialize the weight and then each, the weights using the loss value. Loss is Calculated from the predicted output and true output. Update the weights until the loss is minimized. Once the training is done save the model into disk.

EVALUATING THE MODEL

To evaluate the model calculate F1 macro and micro values from y_preds and prediction based on number of correct predictions and total number of predict . Testing is the process where the performance of a fully trained model is evaluated based on F1 Score.

After building the BI-LSTM Network model, we then trained and validated the model with Adam Optimizer. Then feed the whole graph to both training and testing .Then seperate the training , validation and testing data using Boolean mask that constructed before. These mask will be passed to sample_weight argument set the batch size to be 4 and epoch value 50.

Fig.5.31 Model Summary of Bi-LSTM

Fig.5.32 Training the Bi-LSTM Model

```
    x_test3[min(len(x_test)-1,timeSize+(i-3))]=np.array(oht[0]).T

preds = np.array(preds)
print(preds.shape)
#print("\npreds = ",preds)
trues = np.array(trues)
#print("\ntrues = ",trues)
f1 = f1_score(y_true=np.where(trues>0,1,0), y_pred=np.where(preds>0,1,0), average='micro')
f2 = f1_score(y_true=np.where(trues>0,1,0), y_pred=np.where(preds>0,1,0), average='macro')
ts.append([f1,f2])
print(np.mean(err)," : micro ",f1," : macro",f2," : ")
print(weightSupport)
```

```
0.6767241379310345  :  0.6685404665073442
0.6991525423728814  :  0.690638473828129
0.6973777462792345  :  0.6881352937998514
0.6999468932554435  :  0.69207648058419 21
0.7067287346593313  :  0.6975355899330099
0.7016300496102054  :  0.6944336411543011
0.6867506916692284  :  0.6798628306539825
0.6893959731543624  :  0.6818630784141612
0.6995027231825717  :  0.6911654218755464
0.6976942783945346  :  0.6889808659807264
0.6960746210649047  :  0.6876406769449125
0.6942561541205852  :  0.6849591993385232
0.6903385631709332  :  0.6810823032936354
0.6924961715160797  :  0.6834656722923619
0.6842180981153791  :  0.6759189347380683
(1155, 4)
691.4469  : micro  0.6842180981153791  : macro 0.6759189347380683  :
[0.27768    0.22052966 0.2594949  0.2422954 ]
```

Fig.5.33 Testing the Bi-LSTM Model

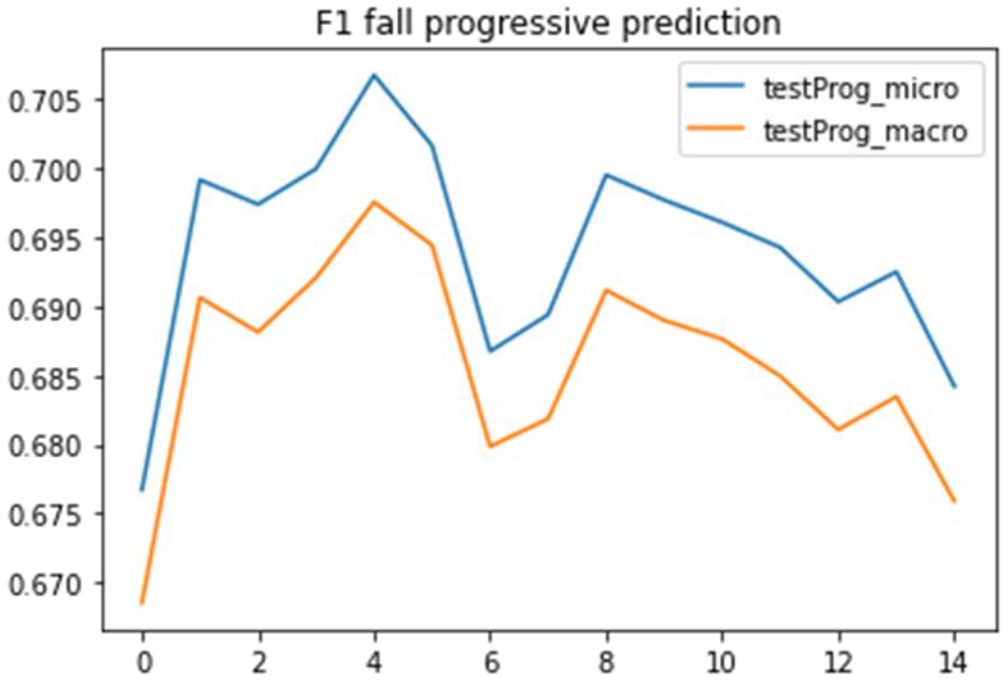


Fig.5.34 F1 progression Bi-LSTM Model

CHAPTER 7

CONCLUSION

5.1 CONCLUSION

We have used Bidirectional Long Short-Term Memory (BI-LSTM) and Machine learning Algorithm to train the model. The dataset is shuffled and split up into training and testing data sets. 80% of the data is for training and the remaining 20% for testing. Comparing the accuracy obtained on testing the model, Bidirectional-LSTM performs good with 70% of F1 score .

We conclude that BI-LSTM model has training accuracy of 70% for Crime Type classification.

5.2 FUTURE WORK

We aim to enhance the performance of the model for accurate prediction by collecting more data from various source to develop model and also classify crime type into precised sub category. We expected to develop the model that can be used to help people in real- world.

CHAPTER 7
REFERENCES

[1]N. H. M. Shamsuddin, N. A. Ali and R. Alwee, 2017, "An overview on crime prediction methods," 2017 6th ICT International Student Project Conference (ICT-ISPC), pp. 1-5.

[2]S. Yao et al., 2020, "Prediction of Crime Hotspots based on Spatial Factors of Random Forest", 15th International Conference on Computer Science & Education (ICCSE), pp. 811-815.

[3]B. Chandra, M. Gupta and M. P. Gupta, 2008, "A multivariate time series clustering approach for crime trends prediction," 2008 IEEE International Conference on Systems, Man and Cybernetics,pp. 892-896.

[4]A. Shukla, A. Katal, S. Raghuvanshi and S. Sharma, 2021, "Criminal Combat: Crime Analysis and Prediction Using Machine Learning," 2021 International Conference on Intelligent Technologies (CONIT),pp. 1-5.

[5]N. Baloian et al., 2017, "Crime prediction using patterns and context," 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 2-9.

[6]Hossain, Sohrab & Abtahee, Ahmed & Kashem, Imran & Hoque, Moshiul & Sarker, Iqbal. (2020). Crime Prediction Using Spatio-Temporal Data, pp. 277-289.

[7] Deepak, Gerard & Rooban, S. & Santhanavijayan, A.. (2021). “A knowledge centric hybridized approach for crime classification incorporating deep bi-LSTM neural network”, Multimedia Tools and Applications,No.80.

[8]W. Safat, S. Asghar and S. A. Gillani, 2021,"Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," in IEEE Access, vol. 9, pp. 70080-70094.

[9]C. Rajapakshe, S. Balasooriya, H. Dayarathna, N. Ranaweera, N. Walgampaya and N. Pemadasa, 2019, "Using CNNs RNNs and Machine Learning Algorithms for Real-time Crime Prediction," 2019 International Conference on Advancements in Computing (ICAC), pp. 310-316.

[10] S. Kim, P. Joshi, P. S. Kalsi and P. Taheri, 2018, "Crime Analysis Through Machine Learning," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 415-420.

[11] S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma and N. Yadav, 2017, "Crime pattern detection, analysis & prediction," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), pp. 225-230