VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018

Project Report
on
“Analysis and Prediction of Crime Hotspots”

Submitted in partial fulfillment of the requirement
for the award of the degree of

Bachelor of Engineering
in
Information Science & Engineering
by

Ayushi Lodha (1BG19IS009)

Under the Guidance Of

Mrs. Divyashree S R
Assistant Professor, Dept. of ISE
B.N.M. Institute of Technology

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Information Science and Engineering
2022 – 2023

B.N.M. Institute of Technology

An Autonomous Institution under VTU

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the project work entitled **Analysis and Prediction of Crime Hotspots** is carried out by **Ayushi Lodha (1BG19IS009),** the bonafide student of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2022-2023. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Mrs. Divyashree S R
Assistant Professor,
Dept. of ISE,
BNMIT

Dr. S. Srividhya
Prof. & Head,
Dept. of ISE,
BNMIT

Dr. Krishnamurthy G N
Principal,
BNMIT

Name of the Examiners **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

We consider it a privilege to express through the pages of this report, a few words of gratitude to all those distinguished personalities who guided and inspired me in the completion of this project work.

We would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMEI, Bengaluru for providing an excellent academic environment in college.

We would like to thank **Prof. T.J. Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of work.

We would like to thank **Dr. S.Y. Kulkarni**, Additional Director, BNMIT, Bengaluru for his extended support and encouragement during the course of work.

We would like to express my gratitude to **Prof. Eishwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance, and encouragement.

We would like to thank **Dr. Krishnamurthy G.N**., Principal, BNMIT, Bengaluru for his constant encouragement.

We would like to thank **Dr. S. Srividhya**, Professor and Head of the Department of Information Science and Engineering, BNMIT, Bengaluru, for her support and encouragement towards the completion of our project work.

We would like to express my gratitude to my guide **Mrs. Divyashree S R**, Assistant Professor, Department of Information Science and Engineering, BNMIT, Bengaluru, who has given us all the support and guidance in completing our project work successfully.

We would like to thank our project coordinator **Mrs. Laxmi V**, Assistant Professor, Department of Information Science and Engineering, BNMIT, for being the guiding force towards the successful completion of our project work.

Ayushi Lodha
(1BG19IS009)

ABSTRACT

Ensemble learning method is a collaborative decision-making mechanism that implements to aggregate the predictions of learned classifiers in order to produce new instances. Early analysis has shown that the ensemble classifiers are more reliable than any single part classifier, both empirically and logically. While several ensemble methods are presented, it is still not an easy task to find an appropriate configuration for a particular dataset. Several prediction-based theories have been proposed to handle machine learning crime prediction problem in India. It becomes a challenging problem to identify the dynamic nature of crimes. Crime prediction is an attempt to reduce crime rate and deter criminal activities. This work proposes an efficient authentic method called assemble-stacking based crime prediction method (SBCPM) based on algorithms for identifying the appropriate predictions of crime by implementing learning-based methods applied to achieve domain-specific configurations compared with another machine learning model. The result implies that a model of a performer does not generally work well. In certain cases, the ensemble model outperforms the others with the highest coefficient of correlation, which has the lowest average and absolute errors. The proposed method achieved classification accuracy on the testing data. The model is found to produce more predictive effect than the previous researches taken as baselines, focusing solely on crime dataset based on violence. The results also proved that any empirical data on crime, is compatible with criminological theories. The proposed approach also found to be useful for predicting possible crime predictions. And suggest that the prediction accuracy of the ensemble model is higher than that of the individual classifier.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them. Machine Learning algorithms are the programs that can learn the hidden patterns fromthe data, predict the output, and improve the performance from experiences on their own. Different algorithms can be used in machine learning for different tasks, such as simple linearregression that can be used for prediction problems like stock market prediction, and the KNNalgorithm can be used for classification problems.

Skillset of human fails to keep track of criminal records, if handled manually. So, there is need for identifying in a novel way, which will help in analysing crime related information. Analysis on crime prediction is currently based on two significant aspects, prediction of crime risk field, and crime hotspot forecast. Data processing techniques are applied to facilitate this task. The expanded accessibility of computers and data innovations have empowered law authorization offices to incorporate broad database. Typically, violence crimes include murders, robberies, rapes, attempt to murder, kidnapping, thefts, riots, dowry death, dowry atrocity etc. The rate of violent crimes is very high in districts of whereas the number of states respectively for the states concerning the crimes against women.

Commenting on the factual data in terms of the number of murders, it is reported that 17 districts listed among the poorest counties. The crimes generally fall in the areas of violence criminal activities mostly against women. Similarly, defying public order as a crime. The crime predictions are generally suggested by using machine learning techniques with respect to what percentage of future violence is possible in crimes. This research has been done for many years, but with some limited algorithms and small dataset. This research claims its novelty with the help of empirical analysis of machine learning and other contributions listed in this section. Though, machine learning models are widely used in crime prediction, but still despite of its expanding application and its gigantic potential, there are numerous regions, where the new procedures created in the zone of artificial intelligence have not been completely explored and has major drawbacks.

1.1 Motivation

To solve a case based upon a particular data there should be a thorough investigation and analysis that is to be done internally. With the amount of crime data that is present in India currently the analysis and decision making of these criminal cases is too difficult for the officials. Identifying this a major problem this paper concentrates on creating a solution for the decision making of crime thatis committed. the vehicle starts driving on its own. An autonomous driving vehicle performs various actions to arrive at its destination, repeating the steps of recognition, judgment and control on its own.

1.2 Problem Statement

Crimes now a days are increasing day by day and with different levels of Intensity and versatility. Result of increased crimes is a great loss to society in terms of monetary loss, social loss and further it enhances the level of threat against the smooth livelihoodin the society. It is a major challenge to understand the versatile data available with us,then model it to predict the future incidence with acceptable accuracy and further to reduce the crime rate.

The main problem is that day to day the population is going to be increased andby that the crimes are also going to be increased in different areas by this the crime ratecannot be accurately predicted by the officials. The officials as they focus on many issues may not predict the crimes to be happened in the future. The officials/police officers although they try to reduce the crime rate they may not reduce in full-fledged manner. The crime rate prediction in future may be difficult for them.

1.3 Objectives

- Extraction of crime patterns by analysis of available crime and criminal data.
- Predict the crime rate and analyze the crime rate to be happened in future. Based on this information the officials can take charge and try to reduce the crime rate.
- Identify precise crime prone zones through crime analysis.
- Displaying the pictorial representation of crime hotspots.
- Facilitate and suggest surveillance plan and scheduling with respect to crime prone zones identified.
- The aim of this project is to perform analysis and prediction of crimes in states

using machine learning models. It focuses on creating a model that can help to detect the number of crimes by its type in a particular state.

1.4 Summary

The focus of this chapter has been to briefly discuss the advances in machine learning and its applications, in particular application of Ensemble learning methods, namely, bagging, boosting and stacking. Ensemble methods combine several base models in order to produce one optimal predictive model. The chapter then discusses the need fora system that can extract, analyze the available criminal data and using the analysis report to predict the crime rate as well as crime prone zones and give an indication or alert to users.

The proposed system will help in reducing number of crimes and will display crime hotspots. After having highlighted the need for the proposed system the approach is discussed for such a system which provides a better and efficient method to predict this project mainly revolves around predicting the type of crime and crime per capita which may happen in future. Using the concept of machine learning we have built a model using training data set that have undergone data cleaning and data transformation using Multi Linear Regression Algorithm. The model predicts the type of crime and Data visualization helps in analysis of data set and prediction of crimes.

The objective of crime mapping and identifying high crime prone zones using various parameters has been met. The objective of facilitating better surveillance planning so as to control crime effectively by police authorities is achieved well within the scope set. The research has generated the crime hot spots distance from the police stations so as police force can have a plan to put a third eye on criminals. As the crime spots have a strong association along roads, the scheduled patrolling can definitely check the crime occurrence in the region The study can be extended further by forecasting the shift in hot spots with respect to various crime investigation using machine learning or the other techniques. It would definitely ensure safety and security in the area and work towards social welfare.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

A literature review is a comprehensive summary of previous research on a topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research. The review enumerates, describes, summarizes, objectively evaluates and clarify the previous research. It gives a theoretical base for the research and help you (the author) determine the nature of research. The literature review acknowledges the work of previous researchers, and in so doing, assures the reader that your work has been well conceived. It is assumed that by mentioning a previous work in the field of study, that the author has read, evaluated, and assimilated that work into the work at hand.

Crime is a complex social phenomenon that has grown due to major changes in society. Law enforcement agencies need to learn the factors that lead to an increase in crime tendency. To curb this, there is always a need for strategies and policies to prevent crime. As a result of technology development, science and information, data mining and artificial intelligence tools are increasingly prevalent in the law enforcement community.

Crime analysis, as part of criminology, is tasks with exploring and discovering crime and its relationship with criminals. Law enforcement is a process that aims to identify the characteristics of crime. Identifying crime characteristics is the first step in developing further analysis. The high volume of crime data and the complexity of the relationships between them have made criminology an appropriate field for applying data mining and machine learning techniques.

2.1 Overview

The survey is a recognized and accepted part of the modern society. It is one of the means by which society keeps it informed, a way of bringing under central situations of increasing size and complexity of obtaining perceptive and standard of comparison. A survey gives an oversight of a field and is thus distinguishing from a sort of study which consists of a microscopic examination of a turf; it is a map rather than a detailed plan.

The survey must be planned before a start is made. The literature review plays a very important role in the project. It is a source from where project ideas are drawn and developed into concepts and finally theories. It also provides a bird’s eye view about the research done in that area so far. Depending on what is observed in the literature review, one will understand where his/her work stands. Here in this literature survey, all primary, secondary and tertiary sources of information were searched.

2.2 Literature Survey

XU ZHANG et al. [1] has found that the model with built environment covariates has better prediction effect compared with the original model that is based on historical crime data alone. This paper takes the historical data of public property crime from 2015 to 2018 from a section of a large coastal city in the southeast of China as research data to assess the predictive power between several ML algorithms. The prediction accuracies of LSTM model are better than those of the other models. The addition of urban built environment covariates further improves the prediction accuracies of the LSTM model.

In this paper, random forest algorithm, KNN algorithm, SVM algorithm and LSTM algorithm are used for crime prediction. First, historical crime data alone are used as input to calibrate the models. Comparison would identify the most effective model. Second, built environment data such as road network density and poi are added to the predictive model as covariates, to see if prediction accuracy can be further improved. By comparing the prediction results of different machine learning models before and after adding covariates, the following indicators are used for evaluation. The Hit Rate mainly includes Grid Hit Rate and Case Hit Rate.

S. Mahmud, M. et al. [2] used different clustering approaches of data mining to analyse the crime rate of Bangladesh and had also use K-nearest neighbour (KNN) algorithm to train our dataset [9]. Finally, to find out our safe route, had used the forecast rate. This job will assist individuals to become aware of the crime area and discover their secure way to the destination. The sparsity of crime in many areas complicates the application of the prediction rate area-specific modelling use three types ML algorithms Linear regression, Naïve Bayes and K-nearest neighbour among which we discover distinct precision in different instances some linear operates good and provides better precision but the general situation K-nearest neighbour provides the appreciated accuracy.

The author used broken window theory, deep learning algorithm, random forest and naïve Bayes to reduce criminal activity and detect the crime zone Prepare the data frame to train the model for recognition of images, pre-processing of information and detection of crime hotspot. The model tuned with deep learning provides 0.87% of the best accuracy. Machine learning offers methods of regression and classification used to predict rates of crimes.

WAJIHA SAFAT et al. [3] applied different, ML algorithms namely, the logistic regression, support vector machine (SVM), Naïve Bayes, k-nearest neighbors (KNN), decision tree, multilayer perceptron (MLP), random forest, and extreme Gradient Boosting (XGBoost), and time series analysis by long-short term memory (LSTM) and autoregressive integrated moving average (ARIMA) model to better fit the crime data. They achieved an improved predictive accuracy for crimes by implementing different machine learning algorithms on Chicago and Los Angeles crime datasets. Among the different algorithms, XGBoost achieves the maximum accuracy on Chicago datasets and KNN achieves the maximum accuracy on Los Angeles.

This study reports an improved efficiency for accurate crime prediction as compared with previously achieved with further analysis based on different machine learning algorithms. Besides crime prediction accuracy, the LSTM for time series analysis was reported using different performance metrics. Moreover, the study also provides a visual summary through exploratory data analysis to portray crime types and count. Finally, the future crime rate and crime density areas for the next five years were examined through ARIMA.

SAPNA SINGH KSHATRI et al. [4] used Ensemble learning method that implements to aggregate the predictions of learned classifiers in order to produce new instances. This work proposes an efficient authentic method called assemble-stacking based crime prediction method (SBCPM) based on SVM algorithms for identifying the appropriate predictions of crime by implementing learning-based methods, using MATLAB. In the present study, ML models with machine learning algorithms (ensemble and simile), i.e., SMO, SVM bagging, SVM-Random Forest, SVM-stacking J48, and Naive Bias, were designed and were implemented. Babakura et al. presented a comparison between Naïve Bayesian and Back Propagation to predict the crime data and classified on various levels of crime rate such as low, medium and high. The accuracy, recall and precision were also calculated. The Naïve Bayesian was found performing better for data classification tested over crime dataset using WEKA. Yadav et al. employed different types of machine learning methods supervised as well as unsupervised. Clustering, k-means clustering, Naïve Bayes, Regression methods are studied for analysis of crime rates and their impact based on criminal data. Zhe Li et al. presented prediction of crimes of China tested over different season’s data. Sivaranjani et al. used K-means clustering, hierarchical clustering and DBSCAN clustering for analysis of crime data of cities in Tamil Nādu state of India.

Liao et al. suggested a novel method of prediction of crimes using Bayesian learning based on different geographic data. Hazwani et al. presented a comparative study between different machine learning methods such as SVM, fuzzy theory, artificial neural network. The multivariate time series report was presented as a result of an extensive comparison of crime prediction methods. The future scope still explained the limitation of current methods for obtaining better accurate results and good performance by optimizing and tuning the parameters.

Hitesh Kumar Reddy et al. [5] used raw datasets that were processed and visualized based on the need. Afterwards, machine learning algorithms were used to extract the knowledge out of these large datasets and discover. The hidden relationships among the data which is further used to report and discover. Accurate real-time crime predictions help to reduce the crime rate. The crime patterns are valuable for crime analysts to analyze these crime networks by the means of various interactive visualizations for crime prediction and hence is supportive in prevention of crimes. For optimum analysis and prediction of crime incidents, a Crime Prediction & Monitoring Framework Based on Spatial Analysis is introduced. In this framework, various visualization techniques are used to analyze the data in a better way. This framework is implemented in a GUI based tool using R programming and its various libraries.

The type of crime is also an important factor as safety measures are majorly taken based on the type of crime. This module helps visualize the crimes that had happened based on category over different areas as shown in Fig.3. This helps the law enforcement to analyze what The project helps the crime analysts to analyze these crime networks by means of various interactive visualizations. The interactive and visual feature applications will be helpful in reporting and discovering the crime patterns. Many classification models can be considered and compared in the Analysis. It is evident that law enforcing agencies can take a great advantage of using machine learning algorithms to fight against the crimes and saving humanity. For better results, we need to update data as early as possible by using current trends such as web and Apps.

BO Yang et al. [6] used Mathematical formulation of ST- Cokriging Spatio-temporal covariance model, accuracy evaluation using Statistical scores of Pearson Correlation Coefficient (PCC), Root Mean Squared Error (RMSE), Predictive Accuracy Index (PAI) and Predictive Efficiency Index (PEI). Crime data were aggregated for (half-) monthly basis in previous research (Levitt 2002, Mares 2013). However, the monthly aggregated data were not evenly distributed in the temporal domain because of the inequality of the number of days in each month. The aggregated crime data in 2012 are further processed to spatial continuous crime risk maps using the kernel density function (Okabe et al. 2009). The spatio-temporal covariance was estimated through the calculation of empirical semi–variograms. Predictions of regular ST-Kriging without the input of co-variable were calculated with the same source of historical crime data. Statistical tests were also performed on the prediction without transitional zones as the control group. PAI and PEI were also calculated to evaluate crime prediction accuracy. Hotspots maps were calculated by varying the threshold on the crime risk map.

Simon Kojo Appiah et al. [7] used number of clusters and model of the clustering process. They identified the centroids of the K-means and combination of the classical E-M and K means algorithms for efficient identification of crime hotspots. The optimal segmentation value (number of clusters) for modelling of the criminal activities was determined by the total within- cluster variation of the K-means, BIC value, and the LRT. The volume of the covariance structure identified in Coventry was equally classified by the classical E-M algorithm and E-M algorithm initialized by K-means clustering process. The study explored a combined use of model-based clustering procedures, widely applied in data mining studies, and efficiently utilized them to unmask the concentration of criminal activities by identifying and parameterizing hotspots of violent crime activities, which are governed by the place-based theories. The study explored a combined use of model-based clustering procedures, widely applied in data mining studies, and efficiently utilized them to unmask the concentration of criminal activities by identifying and parameterizing hotspots of violent crime activities, which are governed by the place-based theories. The study has proven that hotspot analysis of criminal activities is an iterative process, starting from the lowest level to avoid overlooking low-level pattern in crime data. Using large point pattern crime data, two initialized model-based clustering approaches were explored to evaluate the concentrations of these criminal activities. Crime events were modelled as arising from 12 GMMs, whose parameters were estimated by a model-based clustering of E-M algorithm with two different initializations. The proposed E-M algorithm with semi supervised K -clustering initialization proved efficient in identifying crime hotspots, a vital information for crime combating initiatives to be implemented in such identified hotspot areas of criminal activities.

J. Vimala Devi et al. [8] used the adaptive DRQN model to improve the efficiency of the prediction. The adaptive DRQN model is applied with GRU instead of LSTM unit to store the relevant features for a long time. The MDP is applied in an adaptive agent to learn the instances effectively. Reinforcement learning is applied to identify the optimal state value and to update the reward function. The reward function is applied in the proposed adaptive DRQN model to improve the prediction performance of the model. The memory of Deep Q Network (DQN) is used to analyze the observation set of the last collected information that provides more insights about gradient decent observation, bias, and the rate of change.

The reinforcement learning method is applied to the training process. The dual- component and single-component radar signals are applied to give a reward value as feedback to the classification network. The reward function is used to update the network parameter. The reinforcement reward and punishment methods are applied to select the output order of recognition results. The network label of hard restriction in the traditional classification is autonomously avoided and this improves the network adaptability in the pulse overlap ping to form complex dual-component signals. Reinforcement learning is applied in the DQN to train GRU for improvement in the accuracy of the classification network. The crime prediction model helps to predict crime hotspots and helps the police force to prevent crimes from occurring in a particular area. Various existing crime prediction models focused on predicting the crimes by using machine learning methods.

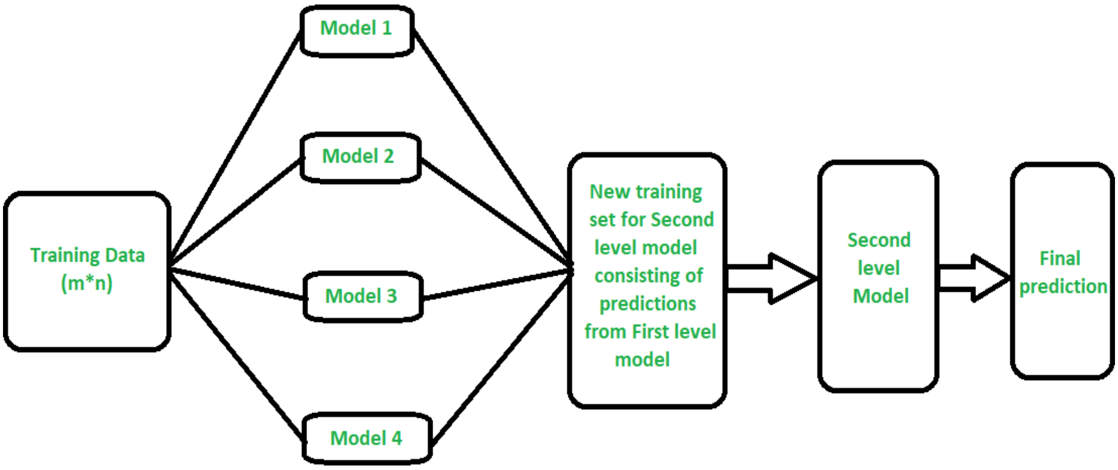
Existing models have the limitations of data imbalance and overfitting problems which affected the performance of these models. In this research, the DRQN method with reward function is proposed to provide a robust crime prediction model. The proposed adaptive DRQN model involves applying the multi-agent-based Markov Decision Process (MDP) to update the state. The reinforcement learning method is applied to update the state of the DRQN model based on the input data. The reward function is introduced in the model to improve learning efficiency.

2.3 Approach Towards the Problem

Stacking, another ensemble method, is often referred to as stacked generalization. This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions. Stacking has been successfully implemented in regression, density estimations, distance learning, and classifications. It can also be used to measure the error rate involved during bagging.

Using Ensemble learning method to aggregate the predictions of learned classifiers in order to produce new instances. Crime prediction is an attempt to reduce crime rate and determine criminal activities. The Decision tree algorithm is applied to achieve domain-specific configurations compared with another machine learning model Gradient Boosting and Random Forest.

The proposed approach will be useful for predicting possible crime predictions. And the prediction accuracy of the stacking ensemble model will be higher than that of the individual classifier. In stacking, an algorithm takes the outputs of sub-models as input and attempts to learn how to best combine the input predictions to make a better output prediction.


Model 1
Model 2
Training Data (m*n)
Model 3
Model 4
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

Model 1: Decision Tree
Model 2: Random Forest
Model 3: Gradient Boosting

Fig 2.1 Stack Generalization Method: Ensembeling Approach

Stacking is one of the most popular ensemble machine learning techniques used to predict multiple nodes to build a new model and improve model performance. Stacking enables us to train multiple models to solve similar problems, and based on their combined output, it builds a new model with improved performance.

This ensemble technique works by applying input of combined multiple weak learners' predictions and Meta learners so that a better output prediction model can be achieved. In stacking, an algorithm takes the outputs of sub-models as input and attempts to learn how to best combine the input predictions to make a better output prediction.

Stacking is also known as a stacked generalization and is an extended form of the Model Averaging Ensemble technique in which all sub-models equally participate as per their performance weights and build a new model with better predictions. This new model is stacked up on top of the others; this is the reason why it is named stacking.

The architecture of the stacking model is designed in such a way that it consists of two or more base/learner's models and a meta-model that combines the predictions of the base models. These base models are called level 0 models, and the meta-model is known as the level 1 model. So, the Stacking ensemble method includes original (training) data, primary level models, primary level prediction, secondary level model, and final prediction. The basic architecture of stacking can be represented as shown below the image.

2.4 Summary

In this chapter the various solutions to the sub problems of scene analysis as discussed in the academia have been presented. These include a survey found that using efficient data collection and data mining techniques to create a better crime prediction using knowledgeable learning to develop multiple models for single problem solving will improve crime forecasting.

The type of methods and the purpose behind the different crime prediction studies and applications varied in the papers we collected. Many of the crime prediction methods were developed for generic crimes and situations, where different models were used and tested in crime prediction to determine the most effective one relative to the provided dataset.

Prediction output of a single classification that aids in predicting what the next crime may be in a specific district within a given time period and identifies the season and Crime has a time dimension in which it occurs more often. The chapter also discusses the proposed methodology to scene analysis and the various potential approaches that may be used in implementing the methodology. The chapter also discusses as to how the individual approaches shall be used in order to suit the real time constraint.

The authors took into account several different representations of criminal depictions and conducted a comparative study. The authors believe that many ideas and procedures have been established for crime prediction, but that field testing is necessary for the usability of those approaches. They provided the approaches of ML and data mining in hotspot detection, in addition to their effectiveness, and outlined the challenges of building a spatiotemporal crime prediction model.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

AND

COST ESTIMATION OF THE PROJECT

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION AND COST ESTIMATION OF THE PROJECT

Requirements analysis is a critical process that helps to ensure the success of a system or software project. Requirements are typically divided into two types: functional and non-functional requirements. Both types of requirements are important for ensuring that a system or software project meets the needs of its users and stakeholders. It's critical to carefully analyse and document requirements before beginning development, as this can help to prevent problems and ensure that the project is completed on time and within budget.

3.1 Hardware Requirements

- Processor - I5/Intel Processor or more
- Hard Disk - 160 GB
- RAM - 8 GB or more

3.2 Software Requirements

- Operating System - Windows 7/8/10
- Server-side Script - HTML, CSS & JS.
- IDE - Pycharm.
- Libraries Used - Numpy, IO, OS, Django, keras.
- Technology - Python 3.6+.

3.3 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

- Creating dataset
- Data Preprocessing
- Real-time monitoring
- Visualization

- Prediction and display of hotspots

3.4 Non-Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

- **Performance**: The system should be able to process large amounts of data in real-time to predict crime hotspots accurately.
- **Scalability**: The system should be scalable to accommodate the increasing amounts of data and the number of users.
- **Reliability**: The system should be reliable, with minimal downtime, to ensure that law enforcement agencies can depend on the system to provide accurate and timely information.
- **Security**: The system should be secure, with appropriate access controls and encryption to protect sensitive information from unauthorized access.
- **Usability**: The system should be user-friendly and easy to use, with a simple and intuitive interface that allows users to quickly access and analyze crime hotspot data.
- **Accessibility**: The system should be accessible to all users, including those with disabilities, and should comply with accessibility standards.
- **Maintainability**: The system should be designed in a way that makes it easy to maintain and update, with clear documentation and well-structured code.
- **Compatibility**: The system should be compatible with a range of devices and operating systems to ensure that it can be accessed and used by law enforcement agencies using different technology platforms.

3.5 Description of COCOMO Model

COCOMO (Constructive Cost Model) is a software cost estimation model developed by Barry Boehm in 1981. The model is used to estimate the effort, time, and cost required to develop a software product. COCOMO is a widely used model in the software industry and provides a systematic approach to estimating software development costs.

COCOMO is based on the assumption that there is a relationship between the software product's size, complexity, and development effort. The model takes into account three levels of software product complexity, which are Basic, Intermediate, and Advanced. These levels are defined based on the product's functional requirements, performance, and usability. It uses a formulaic approach to estimate the software development effort. The model takes into account several input parameters, such as the product size, development team's experience, and the programming language used. The model then calculates the effort required for each development phase, including design, coding, testing, and maintenance.

The COCOMO model consists of three variations:

1. **COCOMO 1**: This model is used for small and simple software projects. It estimates the development effort based on the number of lines of code (LOC) required to develop the software product.
2. **COCOMO 2**: This model is used for medium to large software projects. It estimates the development effort based on the size of the software product, complexity, and the development team's experience.
3. **COCOMO 3**: This model is used for very large and complex software projects. It estimates the development effort based on several factors, such as software size, development team's experience, and the product's complexity.

The COCOMO model provides a valuable tool for software development teams to estimate the effort required for a software project accurately. By using COCOMO, software development teams can plan their resources and budgets better and minimize the risk of cost and time overruns.

3.6 Cost Estimation

- The type of project that is being implemented is a medium sized project so, COCOMO 2 model can be used to estimate the development efforts.
- It involves the analysis of large amounts of data and the development of a predictive model.
- Effort applied (E) is found to be 38- man months.
- Development time is found to be 11 months.
- Minimum number of people required is found to be 2.

We can estimate the development effort as follows:

Effort = 2.94 * (KLOC) ^ 1.09 * (Cf)^1.01 * (sum of E)

Where:

KLOC = Estimated size of the project in thousands of lines of code (10 KLOC in this case)

Cf = Complexity factor (assumed to be 1.0 for a moderately complex project)

E = Effort multipliers based on various factors such as team experience, software tools, and project constraints. The sum of E values for our assumptions is 0.91.

Substituting the values in the formula, we get:

Effort = 2.94 * (10) ^ 1.09 * (1.0) ^1.01 * (0.91) = 38 person-months (PM)

This estimation assumes that the project can be completed in about 38 person-months. The effort applied and development time as shown in the circulation above is in compliance with the actual effort applied and development time taken.

3.7 Summary

This chapter contains the functional, non-functional requirements, Software, Hardware requirements and cost estimation that are needed for efficient working of the proposed system. This chapter provides a clear understanding of the proposed project, helps in reducing risks, improves the quality of the proposed methodology, etc. In summary, a well-defined system requirements specification is critical to the success of any software development project and estimating the cost of the project helps in proper project planning, budgeting, risk management, resource allocation and decision making.

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 Architectural Design

Requirements of the software should be transformed into an architectural that describes the software's top-level structure and identifies its components. This is accomplished through architectural design (also called system design), which acts as a preliminary blueprint from which software can be developed. IEEE architectural design as the process of defining a collection of hardware and software components and their interface to establish the framework for the development of a computer system. This framework is established by examining the software requirement document and designing a model for providing implementation details. These details are used to specify the components of the system along with their inputs, outputs, functions, and the interaction between them. An architectural design performs several functions.

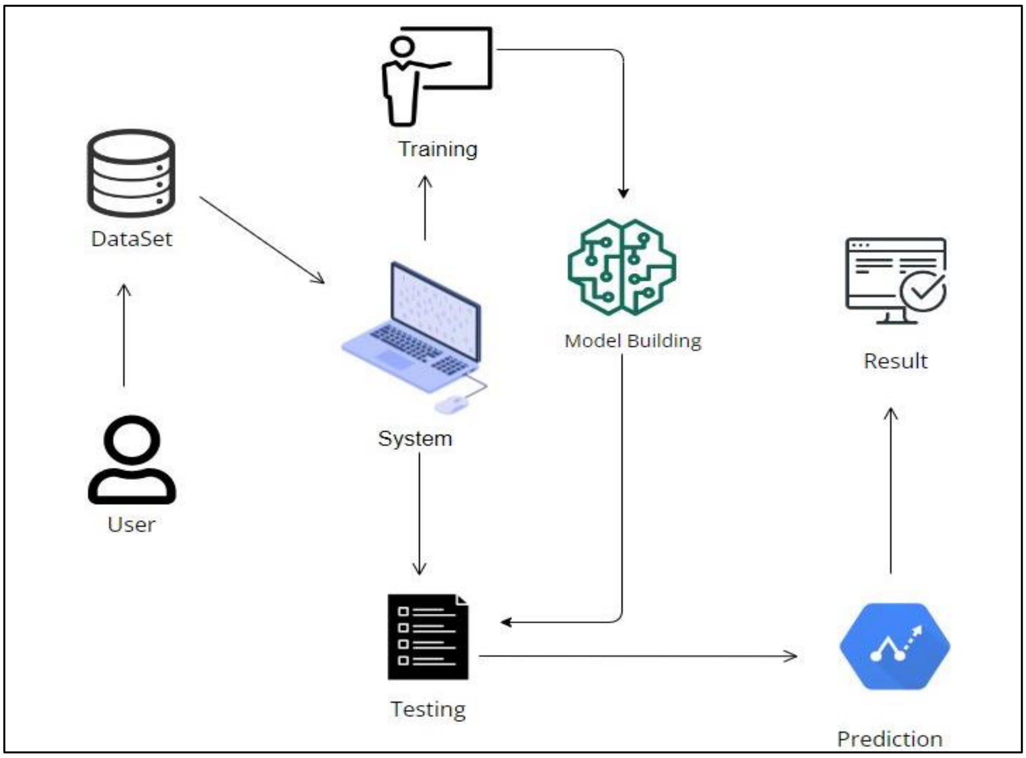


Fig 4.1: Architectural Design

Fig 4.1 shows the system design for crime hotspots involves defining the architecture, components, modules, interfaces, and data of a system to accurately predict the locations with high probability of crime occurrences. This process is based on analysing crime data from various sources, such as police reports, social media, and CCTV footage. The collected data undergoes pre-processing, which involves extracting frames from CCTV footage and resizing them to a standardized format. This pre-processed data is then used to train and test models using various algorithms, such as Random Forest, Decision Tree, and Gradient Boosting, to accurately predict crime hotspots. The system's accuracy is evaluated by comparing the predicted hotspots with actual crime incidents.

4.2 Input/Output Design

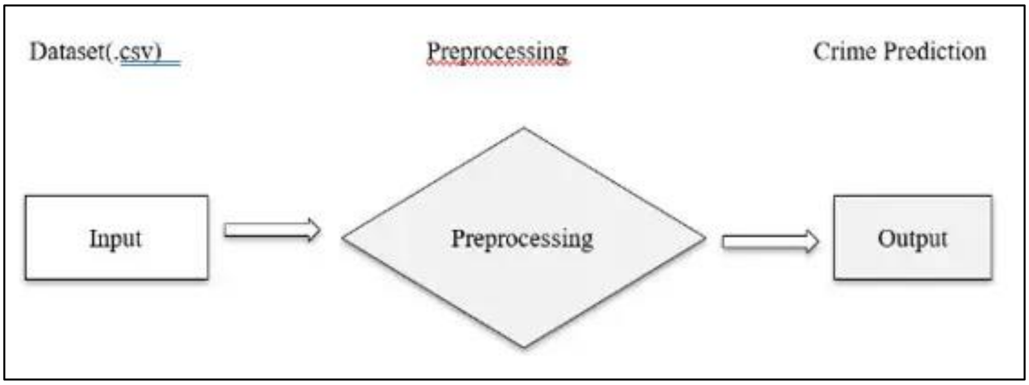


Fig 4.2: Input/Output Design

Fig 4.2 shows the input/output design of the proposed work. Following are the components of the design system:

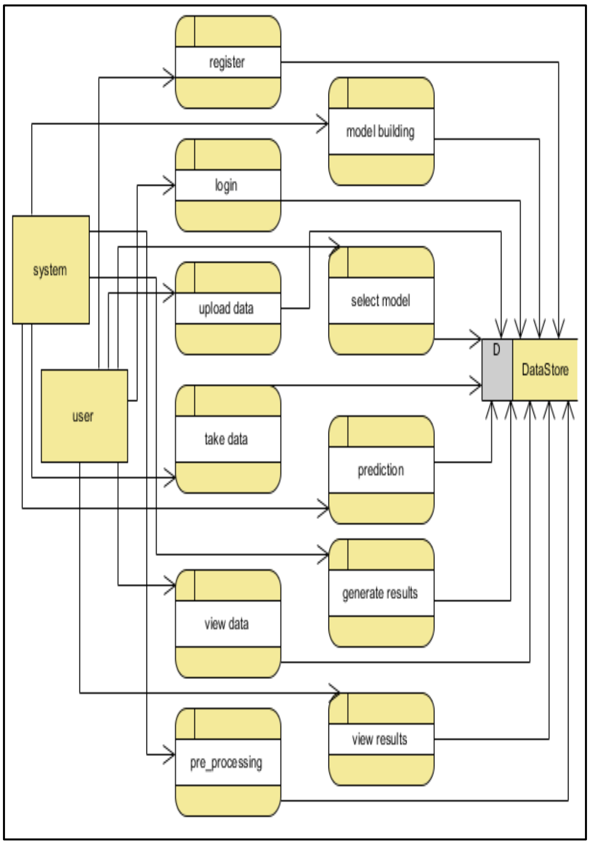
- **Input:** We import the crime dataset. The raw dataset is fed into our machine. A .csv file contains the dataset. After loading the data, it proceeds to pre-process

- **Pre-processing:** Pre-processing is done on raw datasets. The dataset pre-processing is done to convert raw data into clean data.

- **Output:** We apply Random Forest, Decision Tree, and Gradient Boosting algorithm. Finally, crime hotspots prediction is done.

4.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system. Fig 4.3.1 and 4.3.2 shows the level 1 and level 2 of the data flow diagram.



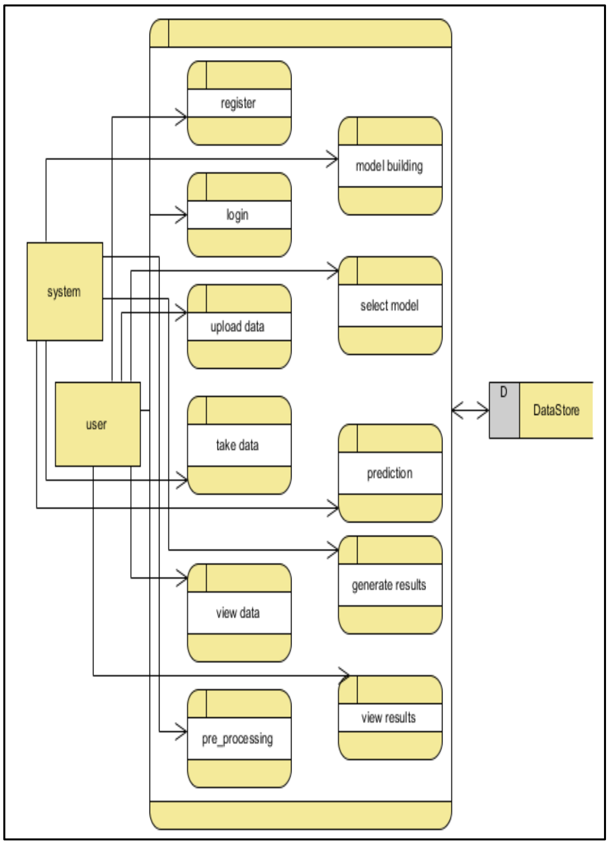


Fig 4.3: Level 1 Diagram

Fig 4.4: Level 2 Diagram

The above figure describes how a data flow diagram (DFD) for crime hotspot prediction illustrates how data is processed to predict areas where crimes are likely to occur. It includes data sources, pre-processing, feature extraction, machine learning models, and prediction results. By analyzing data from various sources, the system predicts the likelihood of a crime occurring in a particular location, helping law enforcement agencies allocate resources effectively to prevent crime.

4.4 Algorithms

This section provides a comprehensive explanation of the concepts of Random Forest, Decision Tree, and Gradient Boosting algorithms. Later, the proposed models using these algorithms are explained along with the crucial libraries used throughout the project mentioned in section 4.5.

1. Decision Tree:

A tree has many analogies in real life and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to represent decisions and decision making visually and explicitly. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal.

A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear, and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

So, what is going on in the background? Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. As a tree generally grows arbitrarily, you will need to trim it down for it to look beautiful. Let's start with a common technique used for splitting

2. Random Forest:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The ‘forest’ generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the over fitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like Scikit-learn).

Features of a Random Forest Algorithm:

- It’s more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of over fitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node’s splitting point.

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

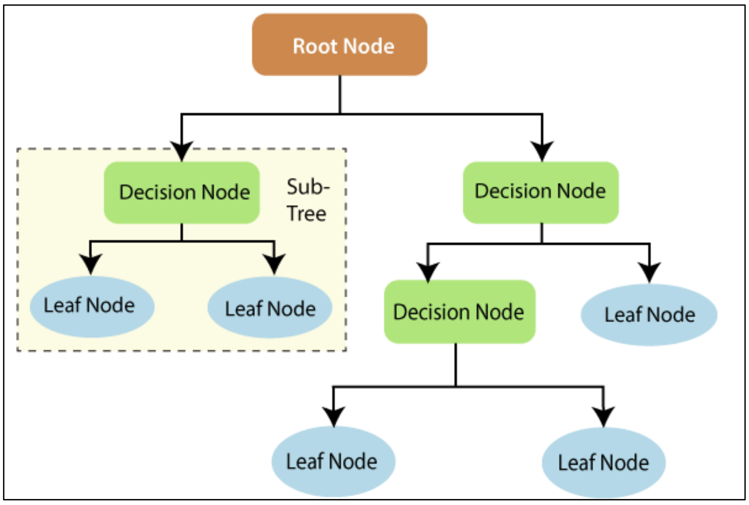


Fig 4.5: Decision Tree

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following fig 4.4.1 shows the three types of nodes in a decision tree.

3. Gradient boosting:

Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e., Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

Unlike, Ad boosting algorithm, the base estimator in the gradient boosting algorithm cannot be mentioned by us. The base estimator for the Gradient Boost algorithm is fixed and i.e., Decision Stump. Like, AdaBoost, we can tune the n_estimator of the gradient boosting algorithm. However, if we do not mention the value of n_estimator, the default value of n_estimator for this algorithm is 100.

Gradient boosting algorithm can be used for predicting not only continuous target variable (as a Regressor) but also categorical target variable (as a Classifier). When it is used as a regressor, the cost function is Mean Square Error (MSE) and when it is used as a classifier then the cost function is Log loss.

Let us now understand the working of the Gradient Boosting Algorithm with the help of one example. In the following example, Age is the Target variable whereas LikesExercising, GotoGym, DrivesCar are independent variables. As in this example, the target variable is continuous, GradientBoostingRegressor is used here.

Let us now find out the estimator-2. Unlike AdaBoost, in the Gradient boosting algorithm, residues (agei – mu) of the first estimator are taken as root nodes as shown below. Let us suppose for this estimator another dependent variable is used for prediction. So, the records with False GotoGym.

4.5 Required Libraries

NumPy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is the fundamental package for scientific computing in Python. It provides a multidimensional array object, various derived objects, and operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Keras

Keras is an open-source neural network library written in Python. It is designed to make building and training deep learning models easier and faster. Keras allows users to define and train neural networks through high-level APIs that are easy to use and understand. It supports both convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and can be run on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML.

Django

Django is a high-level web framework for building web applications in Python. It follows the model-view-controller (MVC) architectural pattern and encourages the use of reusable code by providing a wide range of built-in features and tools. Django was designed to make web development easier and more efficient by providing a robust framework for creating complex applications quickly.

Django includes a powerful object-relational mapper (ORM) that allows developers to interact with databases using Python code. It also includes built-in authentication and authorization systems, which makes it easier to manage user accounts and access to resources. Django provides a template system for creating dynamic web pages and supports multiple languages and time zones.

IO and OS

In Python, the ‘io’ and ‘os’ modules provide different functionalities for handling Input/Output (I/O) operations and interacting with the operating system, respectively.

Input/Output (I/O) refers to the communication between a program and external sources, such as files, standard input/output (stdin/stdout), sockets, and databases. Python provides several built-in modules for performing I/O operations,

The ‘os’ module provides a way to interact with the operating system. It provides a range of functions that can be used to perform operations such as creating directories, listing files, and executing system commands. The os module is very useful when working with files and directories in Python and can help simplify common file and directory operations. It can also be used to execute system commands from within a Python script.

Folium

Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data. With Folium, one can create a map of any location in the world. Folium is actually a python wrapper for leaflet.js which is a JavaScript library for plotting interactive maps.

PROPOSED SYSTEM

Many machine learning algorithms are available for prediction of crime hotspots Some of the machine learning algorithms are Decision Tree, Gradient Boosting We used proposed and compute best method for diagnosis a comparative study of machine learning techniques for crime hotspots detection In this stage we have first implement these dataset and the implement algorithm individual then we are combine these results and an compute the Accuracy.

Advantages:

1. Requires less time
2. Good score
3. Easy to handle

Block Diagram: Fig 4.5 shows the block diagram of the proposed model

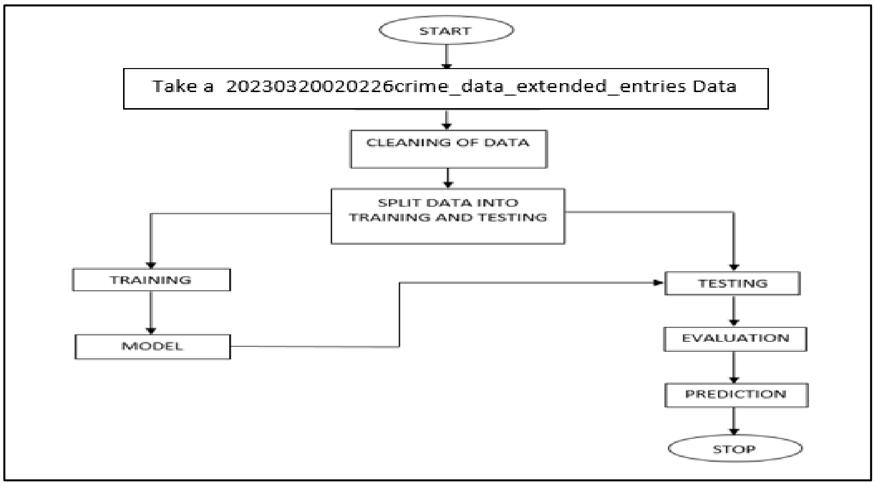

START
Take a 20230320020226crime_data_extended_entries Data
CLEANING OF DATA
SPLIT DATA INTO TRAINING AND TESTING
TRAINING
MODEL
TESTING
EVALUATION
PREDICTION
STOP

Fig 4.6 Proposed Model

CHAPTER 5

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

Implementation is a critical phase in every project since it involves the actual execution of the plans and ideas that were generated during the planning phase. During implementation, the project team will execute the essential actions to transform the project into a usable good or service. Among other things, this could entail developing, testing, and deploying the system or programme.

To make sure that the project is finished on schedule, within budget, and to the acceptable quality standards, it is crucial that the implementation phase be thoroughly planned and managed. Ineffective implementation phase management can lead to delays, cost overruns, and subpar results. Having a transparent project strategy is crucial for ensuring a successful execution.

5.1 Dataset

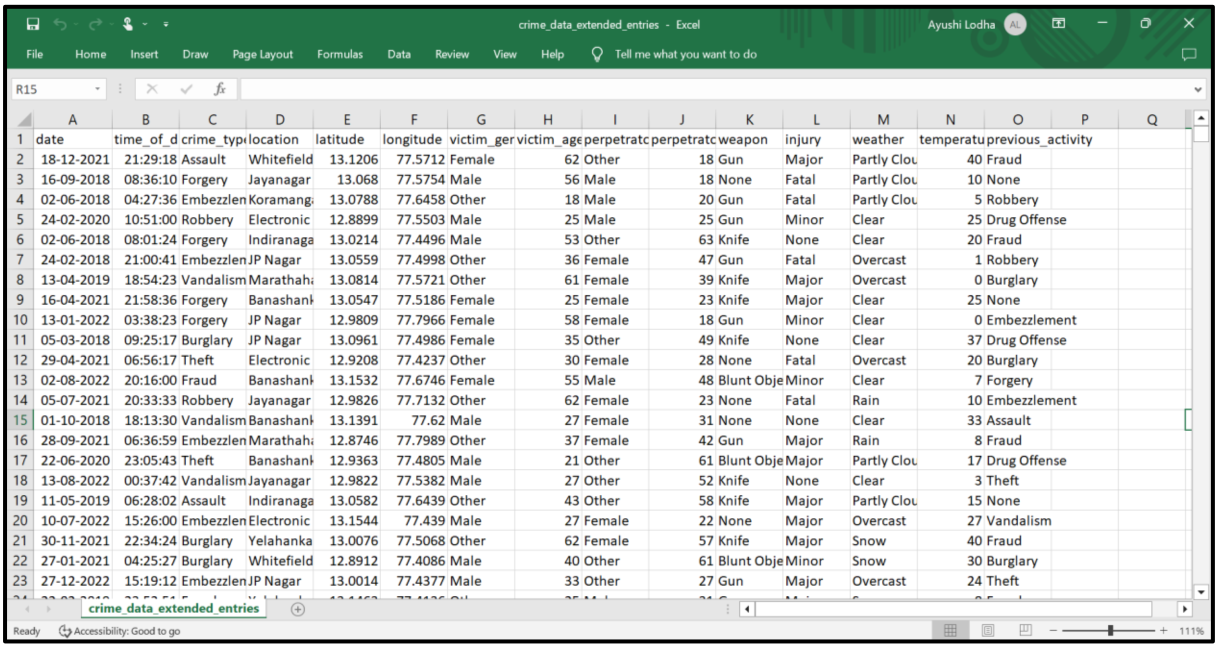
A collection of data is referred to as a data set (or dataset). In the case of tabular data, a data set relates to one or more database tables, where each row refers to a specific record in the corresponding data set and each column to a specific variable. The data set includes values for each of the variables, such as the object's height and weight, for each set member. Data sets may also include a group of files or documents. The dataset contains the information on various factors related to criminal incidents:

1. Date: The date on which the crime occurred.
2. Time_of_day: The time of day when the crime occurred.
3. Crime_type: The type of crime that was committed.
4. Location: The location where the crime occurred.
5. Latitude: The latitude of the location where the crime occurred.
6. Longitude: The longitude of the location where the crime occurred.
7. Victim_gender: The gender of the victim.
8. Victim_age: The age of the victim.
9. Perpetrator_gender: The gender of the perpetrator.
10. Perpetrator_age: The age of the perpetrator.
11. Weapon: The type of weapon used, if any.
12. Injury: The nature and extent of the injuries sustained by the victim.
13. Weather: The weather conditions at the time of the crime.

14. Temperature: The temperature at the time of the crime.

15. Previous_activity: The previous activity of the victim or perpetrator prior to the crime.

This dataset can be used for a variety of purposes, such as analysing crime patterns and trends, identifying risk factors for certain types of crimes, and developing crime prevention strategies. However, it's important to note that the accuracy and reliability of the data may depend on the source of the information and how it was collected. Fig 5.1 shows a glimpse of dataset used for the project.


crime_data_extended_entries - Excel
Ayushi Lodha
AL
File
Home
Insert
Draw
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
R15
crime_data_extended_entries
Ready
Accessibility: Good to go
111%

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	date	time_of_d	crime_typ	location	latitude	longitude	victim_ger	victim_age	perpetratc	perpetratc	weapon	injury	weather	temperatu	previous_activity		
2	18-12-2021	21:29:18	Assault	Whitefield	13.1206	77.5712	Female	62	Other	18	Gun	Major	Partly Clou	40	Fraud		
3	16-09-2018	08:36:10	Forgery	Jayanagar	13.068	77.5754	Male	56	Male	18	None	Fatal	Partly Clou	10	None		
4	02-06-2018	04:27:36	Embezzlen	Koramang	13.0788	77.6458	Other	18	Male	20	Gun	Fatal	Partly Clou	5	Robbery		
5	24-02-2020	10:51:00	Robbery	Electronic	12.8899	77.5503	Male	25	Male	25	Gun	Minor	Clear	25	Drug Offense		
6	02-06-2018	08:01:24	Forgery	Indiranaga	13.0214	77.4496	Male	53	Other	63	Knife	None	Clear	20	Fraud		
7	24-02-2018	21:00:41	Embezzlen	JP Nagar	13.0559	77.4998	Other	36	Female	47	Gun	Fatal	Overcast	1	Robbery		
8	13-04-2019	18:54:23	Vandalism	Marathaha	13.0814	77.5721	Other	61	Female	39	Knife	Major	Overcast	0	Burglary		
9	16-04-2021	21:58:36	Forgery	Banashank	13.0547	77.5186	Female	25	Female	23	Knife	Major	Clear	25	None		
10	13-01-2022	03:38:23	Forgery	JP Nagar	12.9809	77.7966	Female	58	Female	18	Gun	Minor	Clear	0	Embezzlement		
11	05-03-2018	09:25:17	Burglary	JP Nagar	13.0961	77.4986	Female	35	Other	49	Knife	None	Clear	37	Drug Offense		
12	29-04-2021	06:56:17	Theft	Electronic	12.9208	77.4237	Other	30	Female	28	None	Fatal	Overcast	20	Burglary		
13	02-08-2022	20:16:00	Fraud	Banashank	13.1532	77.6746	Female	55	Male	48	Blunt Obje	Minor	Clear	7	Forgery		
14	05-07-2021	20:33:33	Robbery	Jayanagar	12.9826	77.7132	Other	62	Female	23	None	Fatal	Rain	10	Embezzlement		
15	01-10-2018	18:13:30	Vandalism	Banashank	13.1391	77.62	Male	27	Female	31	None	None	Clear	33	Assault		
16	28-09-2021	06:36:59	Embezzlen	Marathaha	12.8746	77.7989	Other	37	Female	42	Gun	Major	Rain	8	Fraud		
17	22-06-2020	23:05:43	Theft	Banashank	12.9363	77.4805	Male	21	Other	61	Blunt Obje	Major	Partly Clou	17	Drug Offense		
18	13-08-2022	00:37:42	Vandalism	Jayanagar	12.9822	77.5382	Male	27	Other	52	Knife	None	Clear	3	Theft		
19	11-05-2019	06:28:02	Assault	Indiranaga	13.0582	77.6439	Other	43	Other	58	Knife	Major	Partly Clou	15	None		
20	10-07-2022	15:26:00	Embezzlen	Electronic	13.1544	77.439	Male	27	Female	22	None	Major	Overcast	27	Vandalism		
21	30-11-2021	22:34:24	Burglary	Yelahanka	13.0076	77.5068	Other	62	Female	57	Knife	Major	Snow	40	Fraud		
22	27-01-2021	04:25:27	Burglary	Whitefield	12.8912	77.4086	Male	40	Other	61	Blunt Obje	Minor	Snow	30	Burglary		
23	27-12-2022	15:19:12	Embezzlen	JP Nagar	13.0014	77.4377	Male	33	Other	27	Gun	Major	Overcast	24	Theft		

Fig. 5.1 Dataset

5.2 Code

```
from django.shortcuts import render,redirect

from.forms import NewUserForm

from django.contrib.auth.models import User

from django.contrib import messages

from django.contrib.auth.forms import AuthenticationForm

from django.contrib.auth import authenticate

from sklearn.preprocessing import LabelEncoder

from imblearn.over_sampling import RandomOverSampler

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
# Create your views here.
def index(request):
    return render(request,'index.html')

def about(request):
    return render(request,'about.html')

def register(request):
    if request.method == 'POST':
        form = NewUserForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request,'Registeration Sucessufull.')
            return redirect("login")
        messages.error( request, "Unsuccessful rregistraion" )
    form = NewUserForm()
    return render(request=request, template_name='register.html', context={'register_form': form})

# login page
def login(request):
    if request.method == "POST":
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:

                messages.info(request, f"You are now logged in as {username}.
             return redirect("userhome")
            else:
                messages.error(request, "Invalid username or password.")
        else:
            messages.error(request, "Invalid username or password.")
    form = AuthenticationForm()
    return render(request=request, template_name= 'login.html', context={"login_form": form})


def userhome(request):
    return render(request,'userhome.html')


def view(request):
    global df
    df = pd.read_excel('crimeapp/20230320020226crime_data_extended_entries.xlsx')
    col = df.head(100).to_html
    return render(request, "view.html", {'table': col})


def moduless(request):
    global df,x_train, x_test, y_train, y_test
df = pd.read_excel('crimeapp/20230320020226crime_data_extended_entries.xlsx')
    #Delete a unknown column
    df.drop("date",axis=1,inplace=True)
    df.drop("time_of_day",axis=1,inplace=True)
    le = LabelEncoder()
    col = 
df[['crime_type','location','victim_gender','perpetrator_gender','weapon','injury','weather','previous_act
ivity']]
    for i in col:
        df[i]=le.fit_transform(df[i])
    x = df.drop(['crime_type'], axis = 1)
    y = df['crime_type']
    Oversample = RandomOverSampler(random_state=72)
    x_sm, y_sm = Oversample.fit_resample(x[:100],y[:100])
    x_train, x_test, y_train, y_test = train_test_split(x_sm, y_sm, test_size = 0.3, random_state= 72)
    if request.method == "POST":
        model = request.POST['algo']
        if model == "1":
            re = RandomForestClassifier(random_state=72)
            re.fit(x_train,y_train)
            re_pred = re.predict(x_test)
            ac = accuracy_score(y_test,re_pred)
            ac
            msg='Accuracy of RandomForest : ' + str(ac)
            return render(request,'moduless.html',{'msg':msg})
        elif model == "2":
            de = DecisionTreeClassifier()
            de.fit(x_train,y_train)
            de_pred = de.predict(x_test)
            ac1 = accuracy_score(y_test,de_pred)
            ac1
            msg='Accuracy of Decision tree : ' + str(ac1)
            return render(request,'moduless.html',{'msg':msg})
        elif model == "3":
            gd = GradientBoostingClassifier()
            gd.fit(x_train,y_train)
            gd_pred = gd.predict(x_test)
            bc = accuracy_score(y_test,gd_pred)
            bc
            msg='Accuracy of GradientBoostingClassifier : ' + str(bc)
            return render(request,'moduless.html',{'msg':msg})
    return render(request,'moduless.html')


def prediction(request):
    global df,x_train, x_test, y_train, y_test
if request.method == 'POST':
```

```
a = float(request.POST['f1'])
b = float(request.POST['f2'])
c = float(request.POST['f3'])
d = float(request.POST['f4'])
e = float(request.POST['f5'])
f = float(request.POST['f6'])
g = float(request.POST['f7'])
h = float(request.POST['f8'])
i = float(request.POST['f9'])
j = float(request.POST['f10'])
k = float(request.POST['f11'])
l = float(request.POST['f12'])

l = [[a,b,c,d,e,f,g,h,i,j,k,l]]
de = DecisionTreeClassifier()
de.fit(x_train,y_train)
pred = de.predict(l)
if pred == 0:
   msg = 'Robbery'
elif pred == 1:
   msg = 'Embezzlement'
elif pred == 2:
   msg = 'Burglary'
elif pred == 3:
   msg = 'Vandalism'
elif pred == 4:
   msg = 'Theft'
elif pred == 5:
   msg = 'Assault'
elif pred == 6:
   print('Forgery')
elif pred == 7:
```

```
            msg ='Drug Offense'
        else:
            msg = 'Fraud'
        lat = b
        lag = c
        print(lat)
        print(lag)
        import folium
        m = folium.Map(location=[19,-12],zoom_start=2)
        folium.Marker([lat,lag],tooltip='click for more',popup=msg).add_to(m)
        m = m.repr_html()
        print(msg)
        return render(request,'result.html',{'msg':msg,'m':m})
    return render(request,'prediction.html')
```

CHAPTER 6

TESTING AND VALIDATION

CHAPTER 6

TESTING AND VALIDATION

Software testing is the process of analysing a system and all of its components to determine whether it satisfies the requirements as stated or not. Testing is the process of running a system to find any flaws, omissions, or gaps from the actual requirements. Software testing is a crucial component of software quality control. Any stage of the development process can result in the introduction of errors. There may still be some faults in the system even after checking and fixing them at each level. The procedure of validating and confirming that a software programme, application, or product: satisfies the commercial and technical requirements that led its design and development, functions as intended, and can be implemented with minimal effort.

6.1 Testing Methods

There are various testing methods used in different fields such as software testing, medical testing, and material testing. Here are some common testing methods:

- Functional testing: This method checks whether the software or product functions as intended. It tests the features, inputs, and outputs of the system.
- Regression testing: This method tests the system after making changes or updates to ensure that existing features or functionalities are not affected.
- Performance testing: This method tests the system's performance under different loads, such as heavy traffic, to ensure that it can handle expected usage.
- Unit testing: This method tests individual units or components of the software or product to ensure that they work correctly.
- Integration testing: This method tests the interaction between different units or components of the software or product.
- Acceptance testing: This method tests whether the software or product meets the requirements of the client or end-user.
- Non-functional testing: This method tests the non-functional aspects of the system, such as security, usability, and accessibility.
- A/B testing: This method compares two versions of a product or website to see which one performs better.
- Exploratory testing: This method involves testing the system without a predefined test

plan, allowing testers to identify unexpected issues.

- Usability testing: This method tests how easy the software or product is to use and whether it meets the needs of the user.

6.2 Test Cases

Test cases are detailed descriptions of the specific steps, conditions, and data needed to test a particular functionality or feature of a software or product. Here are some elements that the test case is include with:

1. Test case ID: A unique identifier for the test case.
2. Test case name: A descriptive name that summarizes the purpose of the test case.
3. Test objective: A clear statement of what the test is intended to accomplish.
4. Preconditions: Any necessary conditions or setup required for the test.
5. Inputs: The data or input required to execute the test.
6. Steps: A detailed set of instructions on how to execute the test.
7. Expected results: The expected outcome or behavior of the system under test.
8. Actual results: The actual outcome or behavior of the system under test.
9. Pass/Fail criteria: The criteria used to determine whether the test has passed or failed.
10. Test status: The status of the test, such as pass, fail, blocked, or in progress.
11. Test environment: The specific environment or configuration used to execute the test.
12. Test data: The specific data used to execute the test.

Table 6.1 Test Cases

Input	**Output**	**Result**
Input	Tested for different model given by user on the different model.	Success
Gradient Boosting	Tested for different input given by the user on different models are created using the different algorithms and data.	Success
Prediction	Prediction will be performed using the different models build from the algorithms.	Success

Table 6.2 Test cases Model building

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Register	Enter email, password, mobile number, gender, address	Registration successful	Registration successfully completed	P
2	Register	Enter email, password, mobile number, gender, address	Registration successful	User Email already existed	F
3	Login	Enter valid email	Login to the Userhome page successfully	Login to the Userhome page successfully	P
4	Login	Enter invalid email or OTP or secret key	Login to the Userhome page successfully	Invalid Email	F
6	User can give a Input	Proper Input.	Algorithm can Predict a required input and generate result on Crime Type	Result Generated Successfully	P
7	User can give a Input	In proper Input	Algorithm cannot Predict a specific input and	Invalid Input	F

Table 2 shows the test cases which includes input, output and results. The input column lists the test data or values that were used as input for the test. The expected output column lists the result that is expected for each input. Finally, the results column lists the actual output generated by the system under test.

Table 3 shows the test case model building which is used to record the results of testing activities. The input column typically lists the different input values that were used for each test case. The expected output column lists the expected output for each input value. The actual output column lists the actual output that was produced by the software system when it was tested with the input value. The pass/fail column is used to indicate whether the test case has passed or failed. If the actual output matches the expected output, then the test case is considered to have passed, and a "Pass" value is recorded in the pass/fail column. If the actual output does not match the expected output, then the test case is considered to have failed, and a "Fail" value is recorded in the pass/fail column.

Testing and validation are crucial steps in the software development process. They help to ensure that software systems are reliable, perform as expected, and meet the needs of their intended users. By adopting effective testing and validation practices, developers and testers can create high-quality software that meets the needs of its end-users and delivers value to the organization.

CHAPTER 7

RESULTS AND DISCUSSIONS

CHAPTER 7

RESULTS AND DISCUSSIONS

This section presents the outcome of the proposed project and interprets the results and puts them in a proper context. The results section typically includes a summary of the data collected, any statistical analyses performed, and the conclusions drawn from the analysis. It should be presented in a clear and concise manner, with tables and figures used to support the findings. The discussion section follows the results section and provides an opportunity for the author to interpret the results, draw conclusions, and make recommendations based on the findings.

7.1 Model Performance

Model performance in machine learning refers to how well a machine learning model can make accurate predictions on new and unseen data. It is a measure of the model's ability to generalize to new data and make predictions that are consistent with the true values of the target variable. Model performance is typically evaluated using a set of predefined metrics, such as accuracy, precision, recall, F1 score, and AUC-ROC, among others, depending on the problem and the data.

The choice of evaluation metrics depends on the specific problem, the type of data, and the goals of the model. Good model performance is essential for the success of a machine learning project, as it determines the usefulness and reliability of the model in solving real-world problems. Achieving good model performance often involves selecting appropriate algorithms, pre-processing techniques, hyperparameters, and optimization strategies to fit the model to the data and minimize prediction errors.

Model accuracy is a commonly used performance metric in machine learning that measures the proportion of correct predictions made by a model on a test set. Specifically, it is defined as the ratio of the number of correctly classified instances to the total number of instances in the test set. While model accuracy is a useful metric for evaluating the performance of a model, it has some limitations, especially in cases where the data is imbalanced, meaning that the number of instances of one class is much larger than the other. In such cases, the model can achieve high accuracy simply by predicting the majority class all the time, while completely ignoring the minority class. Fig 7.1 shows the accuracy results of the modules used in the project.

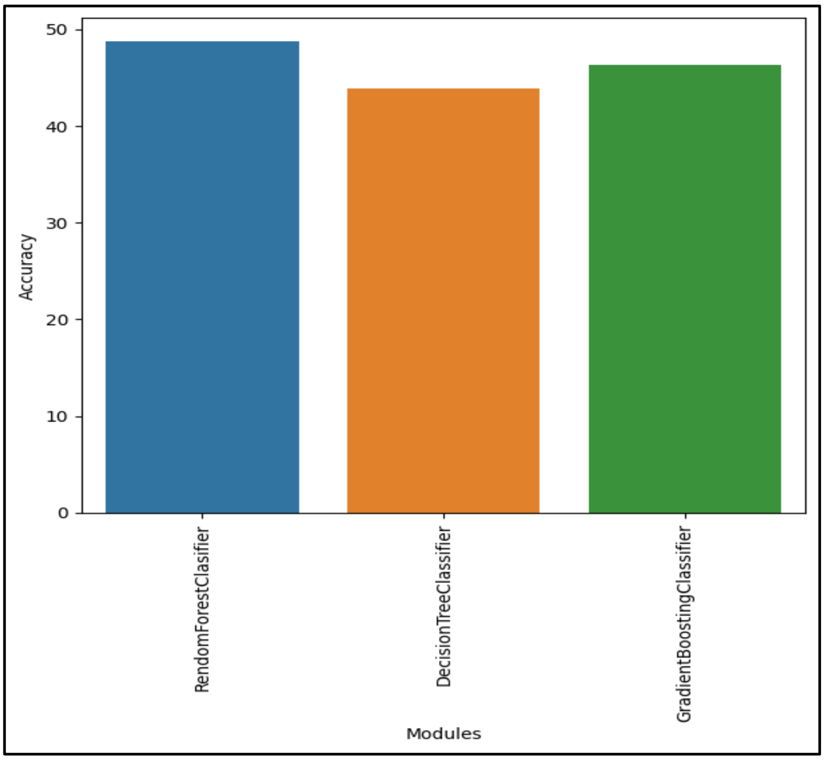
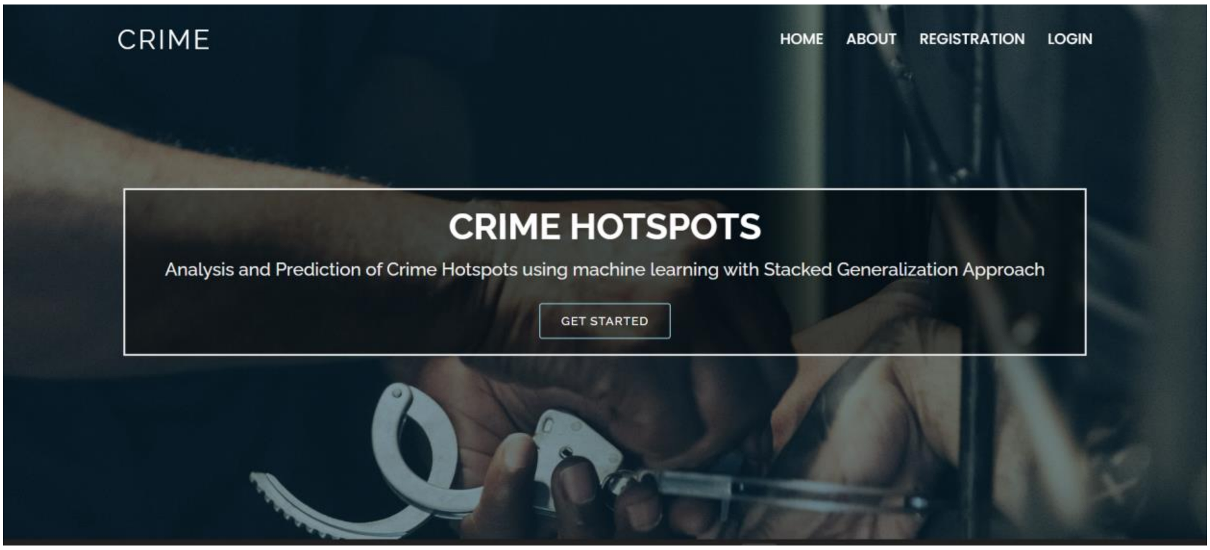


Fig 7.1 Accuracy of Modules used in Project

7.2 Snapshots

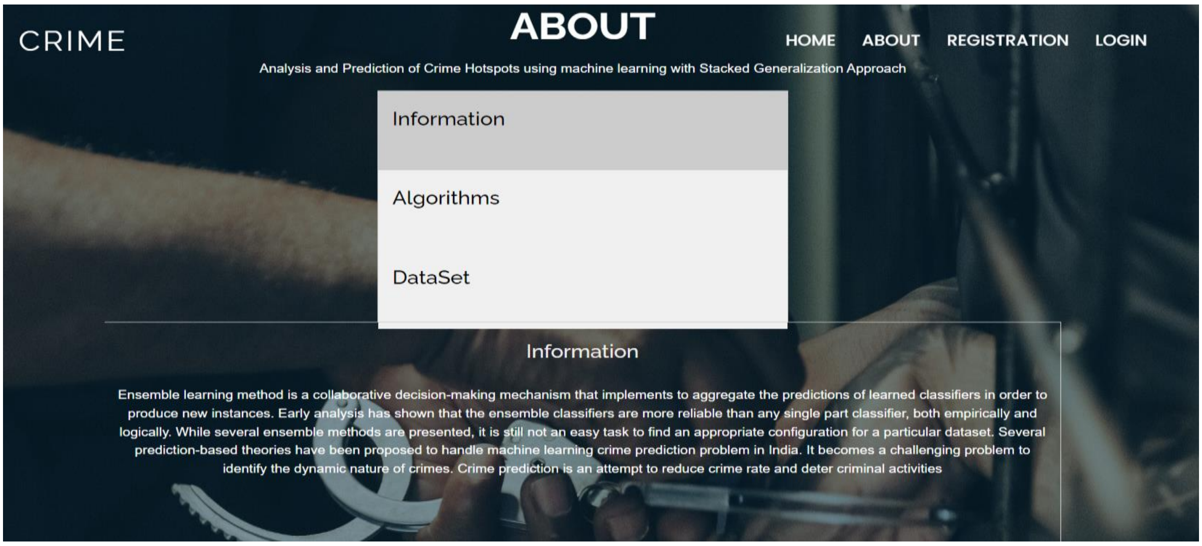
Crime hotspot prediction is the task of identifying areas where crimes are likely to occur in the future. It is an important application of machine learning in criminology and law enforcement, as it can help prevent crimes and allocate resources more effectively. This section consists of the work done during the project. Fig 7.2 shows the home page of the application.



CRIME HOTSPOTS

Analysis and Prediction of Crime Hotspots using machine learning with Stacked Generalization Approach

Fig 7.2 Home Page

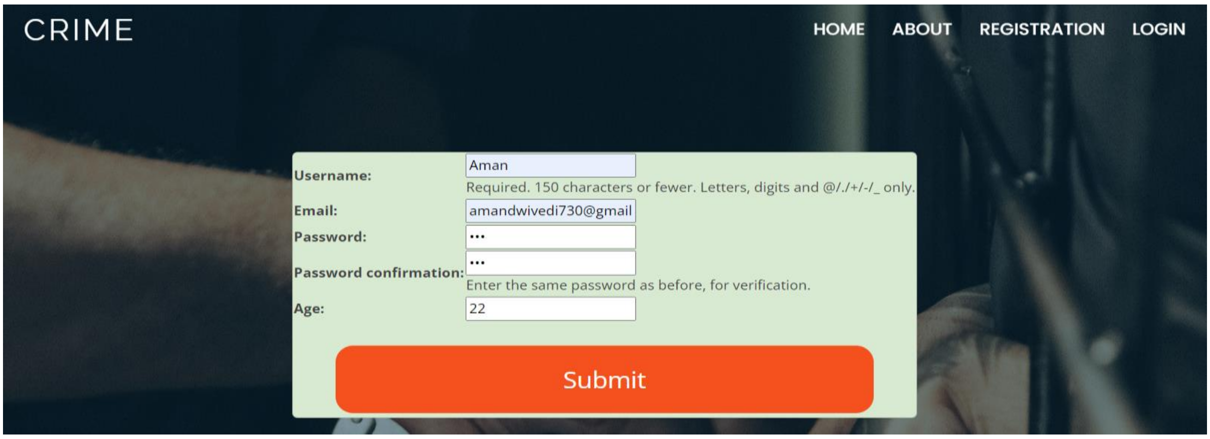


Analysis and Prediction of Crime Hotspots using machine learning with Stacked Generalization Approach

Information

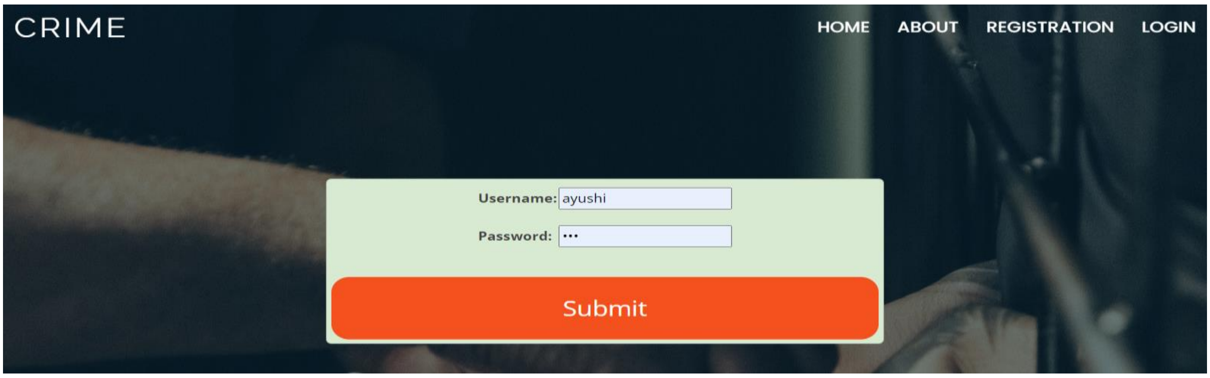
Ensemble learning method is a collaborative decision-making mechanism that implements to aggregate the predictions of learned classifiers in order to produce new instances. Early analysis has shown that the ensemble classifiers are more reliable than any single part classifier, both empirically and logically. While several ensemble methods are presented, it is still not an easy task to find an appropriate configuration for a particular dataset. Several prediction-based theories have been proposed to handle machine learning crime prediction problem in India. It becomes a challenging problem to identify the dynamic nature of crimes. Crime prediction is an attempt to reduce crime rate and deter criminal activities

Fig 7.3 About Page



Username: Aman
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Email: amandwivedi730@gmail
Password: •••
Password confirmation: •••
Enter the same password as before, for verification.
Age: 22

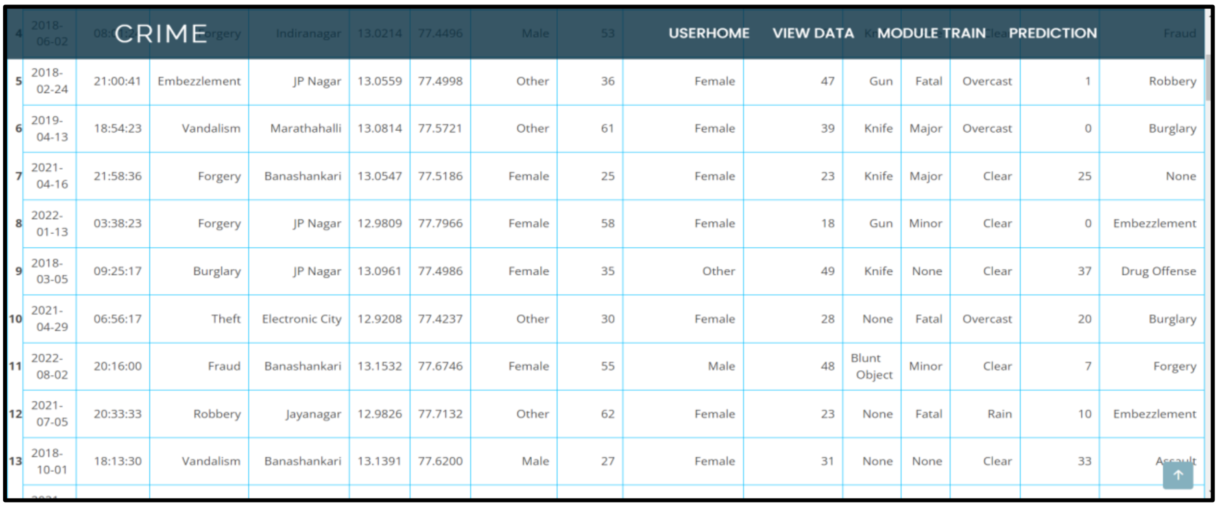
Fig. 7.4 User Registration Page



Username: ayushi
Password: •••

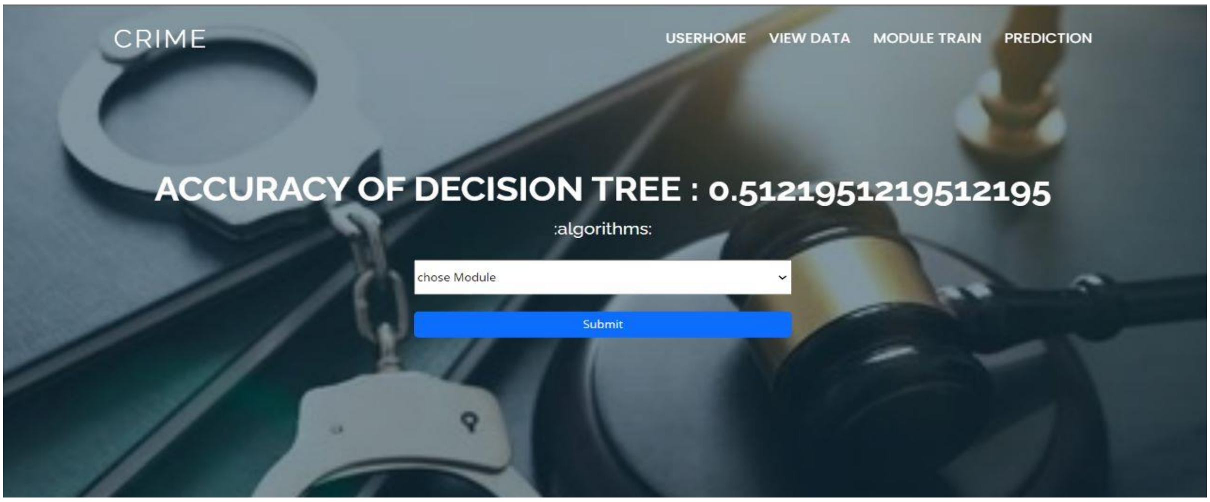
Fig. 7.5 Login Page

Fig. 7.3 shows the about page where a user can get information regarding the algorithms used and also about the dataset. Fig. 7.4 shows the user registration page where user can create a new account and access the application by entering details like name, mail ID, age and password. Fig. 7.5 shows the login page where a user can login to the application once he/she registers.



4	2018-06-02	08:	Forgery	Indiranagar	13.0214	77.4496	Male	53	[illegible]	[illegible]	Kn	[illegible]	Clea	[illegible]	Fraud
5	2018-02-24	21:00:41	Embezzlement	JP Nagar	13.0559	77.4998	Other	36	Female	47	Gun	Fatal	Overcast	1	Robbery
6	2019-04-13	18:54:23	Vandalism	Marathahalli	13.0814	77.5721	Other	61	Female	39	Knife	Major	Overcast	0	Burglary
7	2021-04-16	21:58:36	Forgery	Banashankari	13.0547	77.5186	Female	25	Female	23	Knife	Major	Clear	25	None
8	2022-01-13	03:38:23	Forgery	JP Nagar	12.9809	77.7966	Female	58	Female	18	Gun	Minor	Clear	0	Embezzlement
9	2018-03-05	09:25:17	Burglary	JP Nagar	13.0961	77.4986	Female	35	Other	49	Knife	None	Clear	37	Drug Offense
10	2021-04-29	06:56:17	Theft	Electronic City	12.9208	77.4237	Other	30	Female	28	None	Fatal	Overcast	20	Burglary
11	2022-08-02	20:16:00	Fraud	Banashankari	13.1532	77.6746	Female	55	Male	48	Blunt Object	Minor	Clear	7	Forgery
12	2021-07-05	20:33:33	Robbery	Jayanagar	12.9826	77.7132	Other	62	Female	23	None	Fatal	Rain	10	Embezzlement
13	2018-10-01	18:13:30	Vandalism	Banashankari	13.1391	77.6200	Male	27	Female	31	None	None	Clear	33	Assault

Fig. 7.6 View Data Page

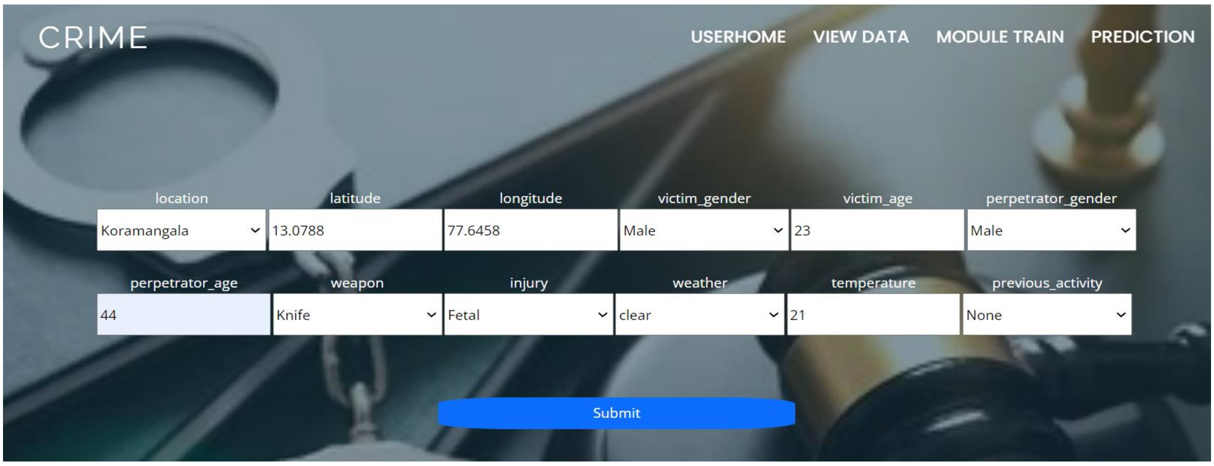


ACCURACY OF DECISION TREE : 0.5121951219512195

:algorithms:

chose Module

Fig. 7.7 Module Train Page



location	latitude	longitude	victim_gender	victim_age	perpetrator_gender
Koramangala	13.0788	77.6458	Male	23	Male

perpetrator_age	weapon	injury	weather	temperature	previous_activity
44	Knife	Fetal	clear	21	None

Fig. 7.8 Prediction Page

Fig 7.6 shows the dataset which is being used in the project. Fig. 7.7 shows the accuracy of different modules used in the project. Fig. 7.8 shows the prediction page where user needs to fill the fields and submit so as to get the crime hotspots.

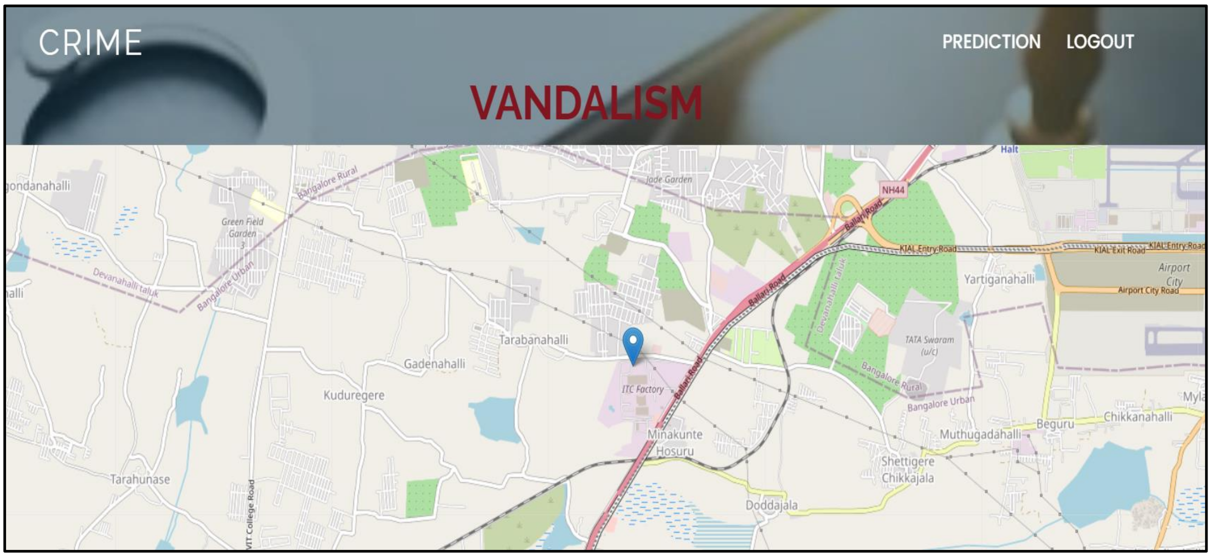


Fig. 7.9 Output Page

Fig. 7.9 shows the output page after the user inputs the data on prediction page, the output is in the form of pin which indicates the crime hotspot.

CHAPTER 8

CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

CHAPTER 8

CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

The conclusion of a project summarizes the main findings, outcomes, and achievements of the project. The conclusion also reflects on any limitations or challenges encountered during the project and suggest possible solutions for overcoming them in the future. The future scope of a project refers to potential opportunities for further development or expansion of the project in the future. This could include additional research or experiments, new features or functionalities, or scaling the project to reach a larger audience. The conclusion and future scope of a project provide valuable insights into the project's success and potential for further development.

8.1 Conclusion

Based on the analysis and prediction of crime hotspots using the stacked generalization approach with decision tree, random forest, and gradient boosting machine learning algorithms, we can conclude that:

1. The stacked generalization approach combining the predictions of multiple models can improve the accuracy of crime hotspot prediction compared to using a single model.
2. Among the three machine learning algorithms used in this project, the decision tree algorithm performed the best in terms of accuracy, precision, and recall.
3. The features that were found to be most important in predicting crime hotspots include location, latitude & longitude, weather, and previous crime incidents.
4. The results of this project can be useful in developing crime prevention strategies and allocating law enforcement resources more effectively.

Overall, the stacked generalization approach with decision tree, random forest, and gradient boosting machine learning algorithms proved to be a powerful tool for predicting crime hotspots and could be applied to other cities or regions to improve crime prevention and law enforcement efforts. Our project aims to develop a user-friendly application based on the use of machine learning models, such as Decision Tree, Random Forest, Gradient Boosting using stacked generalization approach, which can be used to predict crime hotspots, we have used the best techniques that we found, and the application shows the type of crime at the hotspots.

8.2 Future Work

The future scope for the analysis and prediction of crime hotspots using machine learning algorithms is promising. Here are some potential areas for further research and development:

1. Expansion to other cities: The analysis and prediction of crime hotspots can be extended to other cities or regions to improve crime prevention and law enforcement efforts. More datasets from other cities can be collected and analysed using machine learning algorithms to develop a comprehensive model for predicting crime hotspots.
2. Integration of more data sources: The accuracy of crime hotspot prediction can be improved by integrating additional data sources, such as social media and weather data, into the analysis.
3. Development of hybrid algorithms: Hybrid algorithms that combine the strengths of multiple machine learning algorithms can be developed to improve the accuracy and precision of crime hotspot prediction.
4. Real-time prediction: Real-time prediction of crime hotspots using machine learning algorithms can be developed to enable law enforcement agencies to respond quickly to potential crime incidents.
5. Integration with other technologies: Integration of machine learning algorithms with other technologies, such as CCTV cameras and drones, can enhance the accuracy and effectiveness of crime hotspot prediction.

In conclusion, there is a wide range of possibilities for the future development and application of machine learning algorithms for predicting crime hotspots, and continued research and innovation in this area have the potential to significantly improve public safety and security.

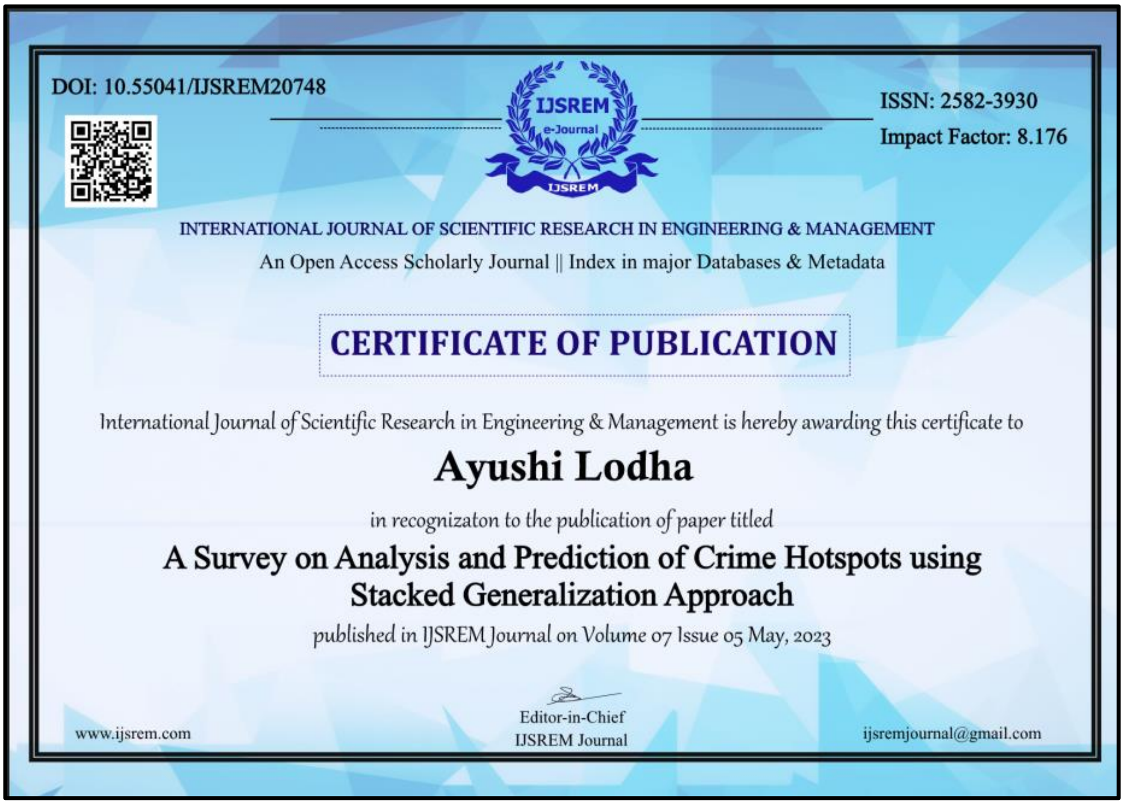
APPENDIX – I

PROJECT CONTRIBUTION

Type of the project	Which of the following aspects are covered by the project?				
Application / Product	**New Technology**	**Safety**	**Ethics**	**Cost**	**Society**
Product	(Yes) – Decision Tree Random Forest Gradient Boosting	Use anonymized data	Avoid biased data	(yes) - cost is in terms of computational resources and manpower	(yes) – Public safety, Law enforcement agencies, Resource Allocation

APPENDIX – II

PUBLICATION



DOI: 10.55041/IJSREM20748

ISSN: 2582-3930

Impact Factor: 8.176

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Ayushi Lodha

in recognizaton to the publication of paper titled

A Survey on Analysis and Prediction of Crime Hotspots using Stacked Generalization Approach

published in IJSREM Journal on Volume 07 Issue 05 May, 2023

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

ijsremjournal@gmail.com

REFERENCES

[1] Zhang, Xu, et al. "Comparison of machine learning algorithms for predicting crime hotspots." *IEEE Access* 8 (2020): 181302-181310.

[2] Mahmud, Sakib, Musfika Nuha, and Abdus Sattar. "Crime rate prediction using machine learning and data mining." *Soft Computing Techniques and Applications*. Springer, Singapore, 2021.

[3] Safat, Wajiha, Sohail Asghar, and Saira Andleeb Gillani. "Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques." *IEEE Access* 9 (2021): 70080-70094.

[4] Kshatri, Sapna Singh, et al. "An empirical analysis of machine learning algorithms for crime prediction using stacked generalization: An ensemble approach." *IEEE Access* 9 (2021): 67488-67500.

[5] Toppi Reddy, Hitesh Kumar Reddy, Bhavna Saini, and Ginika Mahajan. "Crime prediction & monitoring framework based on spatial analysis." *Procedia computer science* 132 (2018): 696-705.

[6] Yang, Bo, et al. "A spatio-temporal method for crime prediction using historical crime data and transitional zones identified from nightlight imagery." *International Journal of Geographical Information Science* 34.9 (2020): 1740-1764.

[7] Appiah, Simon Kojo, et al. "A model-based clustering of expectation–maximization and K-means algorithms in crime hotspot analysis." *Research in Mathematics* 9.1 (2022): 1-12.

[8] Vimala Devi, J., and K. S. Kavitha. "Adaptive deep Q learning network with reinforcement learning for crime prediction." *Evolutionary Intelligence* (2022): 1-12.

[9] Dataset: “Kaggle.com”

[10] Artificial Intelligence: A Modern Approach (3rd Edition) by Rudolph Russell

[11] Introduction to Artificial Intelligence by Wolfgang Ertel & Nathanael T. Black