CRIME PREDICTION AND EVALUATION FRAMEWORK FOR MACHINE LEARNING ALGORITHMS

Major Project Report

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD, KUKATPALLY, HYDERABAD – 500 085.

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted
by

Simran Mathur	**20L51A1234**
Afeefa Shadaan	**20L51A1201**
Raveen Ratan	**20L51A1225**

Under the Esteemed guidance of

Ms Sara Fatima

Asst. Professor, DEPARTMENT OF IT

DEPARTMENT OF INFORMATION TECHNOLOGY

SHADAN WOMEN'S COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUH Aggregated by NAAC B++, Hyderabad)

KHAIRATABAD, HYDERABAD - 500004.

MAY - 2024

CERTIFICATE

This is to certify that the Major Project Report titled **“CRIME PREDICTION AND EVALUATION FRAMEWORK FOR MACHINE LEARNING ALGORITHMS”** is being submitted by **“Simran Mathur - 20L51A1234, Afeefa Shadaan – 20L51A1201, Raveena Ratan - 20L51A1225”** in partial fulfillment of the requirements for the award for the degree of **Bachelor Of Technology** in **Information Technology** to **Jawaharlal Nehru Technological University Hyderabad,”** is a record of bonafide work carried out by them under my guidance and supervision during the academic Year 2023 – 2024—.

The results presented in this Major Project Report have been verified and are found to be satisfactory. The results embodied in this report have not been submitted to any other university for the award of any other degree or diploma.

Ms. Sara Fatima, **Assistant Professor** Department of Information Technology, Shadan Women’s College of Engineering and Technology, Khairatabad – 500 0004.	**Dr. Mohammed Pasha,** **Head of the Department** Department of Information Technology, Shadan Women’s College of Engineering and Technology, Khairatabad – 500 0004.

Major Project viva voce Examination held on ______________________________

INTERNAL INCHARGE EXTERNAL EXAMINER

DECLARATION

We hereby declare that the **Major Project Report titled " CRIME PREDICTION AND EVALUATION FRAMEWORK FOR MACHINE LEARNING ALGORITHMS"** is a record of work done by us in the Department of Information Technology**, Shadan Women's College of Engineering and Technology, Khairatabad** affiliated to the **Jawaharlal Nehru Technological University Hyderabad,** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology.**

The results embodied in this thesis have not been submitted to this/any other University for the award of any other degree or diploma.

Simran Mathur 20L51A1234

Afeefa Shadaan 20L51A1201

Raveena Ratan 20L51A1225

ACKNOWLEDGEMENT

The satisfaction and euphoria of the successful completion of any work could be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement crown our efforts with success.

We take this opportunity to express our grateful acknowledgment to **The Management, of** Shadan Women's College of Engineering and Technology, Khairatabad for their kind encouragement and for granting permission to do this major project.

We also express our thanks to **Dr. K.Palani, M.E, Ph.D., Principal, Professor** Shadan Women's College of Engineering and Technology, Khairatabad for providing infrastructure and facilities.

We also express our thanks to **Dr.Hima Bindu Vice Principal,** Shadan Women's College of Engineering and Technology, Khairatabad for providing infrastructure and facilities.

We express our sincere gratitude to **Dr. Mohammed Pasha, Head of The Department of Information Technology**, for his extended help, concern, and persistent encouragement and support.

We wish to express our heartfelt thanks and sincere acknowledgment to our guide **Ms Sara Fatima, B.Tech.**, **Assistant Professor** for her concern and timely help to direct us for every move in this project.

We will be failing in duty if we do not acknowledge with grateful thanks the authors of the references and other literature referred in this project.

We finally thank our parents, friends, and relatives who rendered their help directly or indirectly for the completion of this major project.

Simran Mathur 20L51A1234

Afeefa Shadaan 20L51A1201

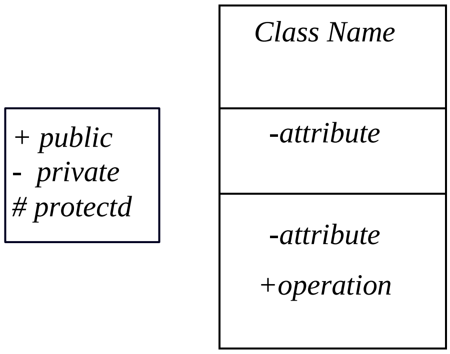
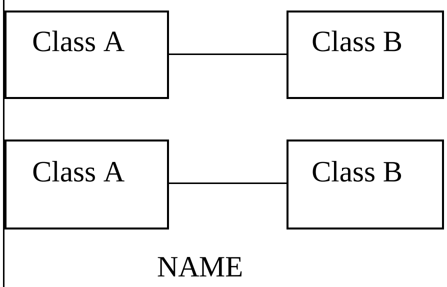
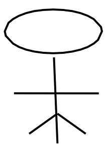
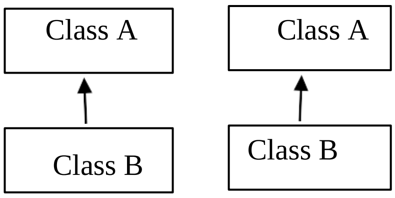
Raveena Ratan 20L51A1225

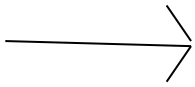
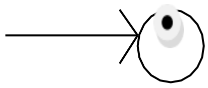
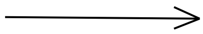
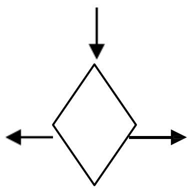
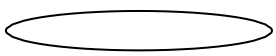
TABLE OF CONTENTS

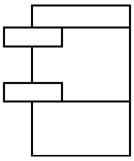
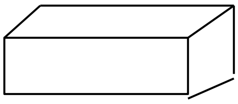
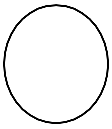
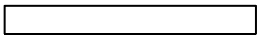
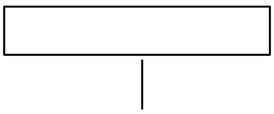
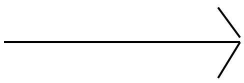
LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment
5.	Relation (uses)	Uses	Used for additional process communication.

6.	Relation (extends)	Extends 	Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communicaon		Communication between various use cases.
8.	State	State 	State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Uses case	Uses case 	Interact ion between the system and external environment.

14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

ACRONYMS

S.NO	EXPANSION	ABBREVIATION
1.	2D Convolutional Neural Network	2D-CNN
2.	Mel-Frequency Cepstral Coefficients	MFCC
3.	Constant Q Transform	CQT
4.	Fast Fourier Transform	FFT
5.	Multilayer Perceptron	MLP
6.	Short-time Fourier Transform	STFT
7.	Unified Modeling Language	UML
8.	Information and Communication Technologies	ICTs
9.	Geographical Information Systems	GIS
10.	Table of Contents	ToC
11.	Model-View-Template	MVT
12.	Web Server Gateway Interface	WSGI
13.	Cross-site Request Forgery	CSRF
14.	University of California Irvine	UCI
15.	Identification	ID
16.	password	pwd
17.	Uniform Resource Locator	URL
18.	Hypertext Markup Language	html
19.	Graphical User Interface	GUI
20.	Application Programming Interface	API

ABSTRACT

The rapid growth of global populations, especially in major cities, has given rise to new challenges, particularly in the realm of public safety regulation and optimization. Consequently, this paper presents a strategy for predicting crime occurrences within a city based on historical events and demographic observations. Specifically, this study proposes a crime prediction and evaluation framework for machine learning algorithms at the network edge. The efficacy of the proposed framework is validated through a comprehensive analysis of four distinct types of crimes: murder, rapid trial, repression of women and children, and narcotics-related offenses. The complete study and implementation process has resulted in a visual representation of crime patterns across various regions of the country. The entire work is accomplished through the selection, assessment, and implementation of Machine Learning (ML) models, ultimately proposing a crime prediction approach. Criminal risk is predicted using classification models for a particular time interval and location. To anticipate occurrences, ML techniques such as Decision Trees, Neural Networks, K-nearest neighbors, and Impact Learning are being utilized, and their performance is compared based on the data processing and modification methods employed. The Decision Tree algorithm achieved a maximum accuracy of 81% during the crime prediction process. The findings demonstrate that employing Machine Learning techniques aids in the prediction of criminal events, thereby contributing to the enhancement of public security. By leveraging historical data and demographic observations, this framework aims to provide a proactive approach to crime prevention and resource allocation, ultimately fostering safer and more secure communities.

Keywords: 2D – CNN, Crime Prediction, Impact Learning, Classification Algorithm.

CHAPTERS

Chapter - 1

INTRODUCTION

1.1 GENERAL

Crime and public safety are major concerns for people all around the world. As more and more people move to live in big cities, it becomes harder to keep everyone safe. The United Nations predicts that by 2050, over 70% of the global population will be living in urban areas. Adding to the worry, the number of violent attacks by terrorist groups was the greatest amount ever reported in the past ten years.

To tackle these problems, researchers are looking into using smart computer programs called machine learning. These programs can study patterns in crime data and other information to help predict where and when crimes might happen. This kind of crime prediction can be really helpful in many ways.

It can allow the police to plan their patrol routes more efficiently, sending officers to areas that are identified as high-risk for crimes. It can also warn tourists about the most dangerous areas in a city, so they can avoid those places. By predicting crime hotspots, machine learning can help improve public safety for everyone.

Different machine learning methods like Decision Trees, Neural Networks, and K-nearest neighbors are being tested. These methods analyze the crime data in different ways to find patterns and make predictions. The accuracy of each method is compared to see which one works best.

The main goal is to create a system that can reliably predict criminal activities in a city or region, based on past crime records and information about the people living there. With this kind of system, the police can be better prepared and take action to prevent crimes before they happen.

1.2 OBJECTIVE

The study offers a novel approach to address the limitations of crime prediction systems. The approach combines 2D convolutional neural networks (2D-CNNs) with diverse data sources beyond traditional crime data. The researchers highlight the shortcomings of current methodologies that rely heavily on police reports and fail to incorporate demographic factors or social media activity which can significantly influence crime patterns.

The proposed approach overcomes the limitations of traditional approaches by using 2D-CNNs to learn features directly from spectrograms and waveforms, capturing more relevant information than traditional hand-crafted features.

The researchers argue that the proposed approach has the potential to provide a more sophisticated and accurate method for crime prediction by accounting for complex patterns and relationships that traditional methods may miss.

1.3 EXISTING SYSTEM

The benchmarked systems in question employed diverse audio representations, encompassing Mel-Frequency Cepstral Coefficients (MFCC), Constant Q Transform (CQT), and Fast Fourier Transform (FFT). These representations were processed through simple linear models, specifically Softmax Regression, for genre classification. Additionally, multilayer perceptron (MLP) models were employed with MFCC, CQT, and FFT features as inputs. Notably, the existing systems refrained from incorporating convolutional neural networks or other deep learning methodologies. Instead, the emphasis was on utilizing hand-engineered input features, eschewing learned representations.

The comparison among these systems highlighted a reliance on traditional signal processing techniques and relatively straightforward models, underscoring a departure from more complex neural network architectures. This succinctly captures the essence of the evaluated approaches and their shared characteristics, offering a comprehensive overview of the key methodologies and limitations inherent in the existing systems.

1.3.1 DISADVANTAGES OF EXISTING SYSTEM:

- Limited Representational Capacity:
- Lack of Temporal Modeling
- Restricted Model Abilities
- Memory usage
- Lack of End-to-End Learning

1.4 PROPOSED SYSTEM

The method of using a 2D CNN for crime prediction is a promising approach that can have a significant impact on public safety and law enforcement. By evaluating various data sets, including crime incidence reports, meteorological data, and demographic information, we can accurately identify criminal hotspots and lower crime rates in designated locations. The method transforms data by using 2D images such as heat maps or choropleth maps as input, which are then analyzed by the CNN, which predicts future incident locations by analyzing the spatial and temporal patterns of the images.

The use of transfer learning, which refines an already trained CNN by utilizing knowledge from previous tasks, can further increase the accuracy of the system. However, it is essential to perform a thorough examination of the data to identify potential sources of bias and implement the necessary countermeasures to ensure the system's impartiality and success. Additionally, citing references and using the correct referencing style is essential to avoid plagiarism.

In our research, we aimed to classify music genres using audio data instead of manually constructed MFCCs. To achieve this, we utilized a 2D CNN to learn temporal and temporal patterns on spectrograms. By taking 30-second audio snippets and using the Short-time Fast Fourier Transform (STFT) to convert them into spectrograms, we identified pitch slides, harmony, and percussion using four filters.

We then obtained four feature maps by convolving the spectrogram with these filters. To achieve translation invariance and reduce dimensionality, we used 2x2 max pooling on the feature maps. Our research showed that 2D CNN-learned spectrogram features perform better for genre classification than designed MFCC features. We also discovered that end-to-end feature learning is more promising than pipeline systems. Our study demonstrates that utilizing a 2D CNN on spectrograms for end-to-end training and feature learning works better than conventional techniques for music classification

1.4.1 Advantages Of Proposed System:

- Features Extraction
- Capture temporal/spectral patterns
- Translation invariance
- End-to-end learning of features and classifier.

Chapter - 2
LITERATURE SURVEY

2.1 GENERAL

This chapter reviews key techniques and algorithms for crime prediction, such as regression methods, spatial-temporal analysis, and machine learning models. It discusses the drawbacks of existing approaches and highlights the benefits of incorporating dynamic data sources and advanced algorithms. The chapter examines machine learning algorithms applied to crime datasets and outlines areas for further research to improve crime forecasting models' accuracy and real-time capabilities.

2.2 Edge-Assisted Crime Prediction And Evaluation Framework For Machine Learning Algorithms.

Author: Apurba Adhikar, Saydul Akbar Murad, Choong Seon Hong, and Md. Shirajum Munir

Year **:** 2022

Apurba Adhikar, Saydul Akbar Murad, Choong Seon Hong, and Md. Shirajum Munir (2022) provides a comprehensive overview of the various machine learning (ML) techniques that have been explored for crime prediction and analysis tasks. These techniques encompass a wide range of approaches, including decision tree algorithms like J48, the K-nearest neighbors method, artificial neural networks (particularly multi-layer perceptrons), logistic regression, support vector machines, random forests, and ensemble models. Researchers have leveraged real-world crime datasets from diverse sources, such as police records, open data portals of cities, and public repositories like the UCI Machine Learning Repository, to develop and evaluate their crime forecasting models. Commonly used evaluation metrics include accuracy, precision, recall, F1-score, and mean squared error. While these studies have yielded promising results, the authors highlight several areas that present research opportunities, such as integrating diverse data sources beyond historical crime records, developing online learning models for real-time prediction, enhancing model interpretability, and designing specialized architectures tailored to different crime types and scenarios. These opportunities underscore the need for continued research efforts to address the complex challenges associated with crime prediction and analysis.

2.3 An Application For Risk Of Crime Prediction Using Machine Learning

Author: Luis Fonseca, Filipe Cabral Pinto, and Susana Sargento

Year : 2021

Luis Fonseca, Filipe Cabral Pinto, and Susana Sargento (2021) discuss the application of machine learning techniques for crime prediction and enhancing public safety. They highlight previous studies that employed various approaches, such as McClendon et al.'s use of linear regression to predict violent crimes using historical crime data, and Lin et al.'s spatial-temporal analysis method based on the "broken windows" theory to identify emerging crime hotspots. Notably, Rumi et al. demonstrated that integrating dynamic features from location-based social networks and demographic data can significantly improve crime prediction performance. Building upon these works, Fonseca, Pinto, and Sargento aim to predict categorical crime risk levels for neighborhoods and time periods in San Francisco, utilizing historical crime data and spatial-temporal analysis. Their approach involves analyzing the spatial distribution of crimes and their temporal patterns to develop models that can accurately forecast the risk levels of different crime types across various neighborhoods and time periods. By leveraging historical crime records, spatial-temporal analysis techniques, and integrating diverse data sources, their study contributes to improved public safety measures and resource allocation strategies.

2.4 Smart City With Chinese Characteristics Against The Background Of Big Data: Idea, Action And Risk

Author: Yuzhe Wu and Weiwen Zhang

Year : 2018

Yuzhe Wu and Weiwen Zhang (2018) discuss the concept of a "smart city" as a potential solution to urban issues arising from rapid urbanization. Smart cities aim to integrate municipal services, businesses, transportation, utilities, and other urban systems through advanced information and communication technologies (ICTs) like the Internet, the Internet of Things, cloud computing, and big data analytics. Several developed countries like the U.S., European nations, South Korea, and Japan have made notable strides in implementing smart city initiatives focused on areas such as energy efficiency, resource optimization, traffic management, public safety, and citizen services. In China, the government has emphasized developing smart cities aligned with its new urbanization plans, while researchers have studied key aspects like rational infrastructure planning, establishing long-term governance mechanisms, and effectively utilizing smart technologies to enhance urban management functions and meet citizens' needs. However, the authors also highlight risks associated with smart cities, including information security concerns, emergency response capabilities, and the lack of indigenous core technologies.

2.5 Crime Prediction Using Decision Tree (J48) Classification Algorithm

Author: Emmanuel Ahishakiye, Danison Taremwa, and Elisha Opiyo

Year : 2017

Emmanuel Ahishakiye, Danison Taremwa, and Elisha Opiyo (2017) discuss several approaches explored for applying machine learning techniques to crime prediction. They mention the work of McClendon et al., who demonstrated the effectiveness of algorithms like linear regression on historical crime data for predicting violent crimes. Additionally, Lin et al. proposed a spatial-temporal method based on the "broken windows" theory, showing that deep learning models tuned on accumulated crime data over different time scales can provide improved hotspot prediction. Furthermore, Rumi et al. explored using dynamic features from location-based social media in addition to demographic data, finding that such dynamic mobility information can significantly improve crime prediction performance by 2-16% across different crime categories. The authors also highlight other studies that have compared the performance of various classification algorithms like decision trees, naive Bayes, neural networks, and support vector machines on crime datasets, with decision tree algorithms like J48 often showing high accuracy and efficiency for predicting crime categories based on historical and demographic features.

2.6 Using Machine Learning Algorithms To Analyze Crime Data

Author: Lawrence McClendon and Natarajan Meghanathan

Year : 2015

Lawrence McClendon and Natarajan Meghanathan (2015) investigated the use of data mining and machine learning techniques for analyzing and preventing crime. They compared the effectiveness of three machine learning algorithms: linear regression, additive regression, and decision stump, in analyzing violent crime patterns from the UC Irvine Communities and Crime Unnormalized Dataset. Their study found that the linear regression algorithm performed best in predicting various crime metrics, such as murders, rapes, robberies, and assaults per 100K population. To validate the model's performance, the authors compared its predictions to actual 2013 crime statistics for the state of Mississippi. The results demonstrated that the linear regression model was accurate in predicting violent crime patterns based on the socio-economic and law enforcement data available in the dataset. Overall, the study by McClendon and Meghanathan highlights the efficacy of machine learning techniques, particularly linear regression, in effectively analyzing and predicting violent crime patterns from relevant socio-economic and law enforcement data sources.

Chapter – 3

REQUIREMENT ANALYSIS

3.1 GENERAL

This chapter provides details about the system requirements. The functional, non-functional, hardware, and software requirements are explained in detail. The functional requirements define the function of the system. The function is described as a set of inputs, the behavior, and outputs. The non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The hardware and software requirements describe the tool and the languages used to build the system.

3.2 HARDWARE REQUIREMENTS

Processor	: Intel i3
RAM	: 32 GB
Hard Disk Drive	: Minimum 1 TB
Monitor	: 27" Color Monitor

3.3 SOFTWARE REQUIREMENTS

Front End/GUI Tool	: Microsoft Visual Studio 2018
Operating System	: Windows 10 64-bit
Language	: PYTHON
Application	: Window Application
Back End	: SQL SERVER 2005

3.4 FUNCTIONAL REQUIREMENTS

- **Data Collection:** The system should be able to collect and integrate various sources of data such as crime reports, demographic information, social media activity, and meteorological data.
- **Data Analysis:** The system should be capable of analyzing the collected data to identify patterns and trends related to criminal activities.
- **Prediction Model:** The system should develop and implement machine learning algorithms to predict potential crime hotspots and trends based on the analyzed data.
- **Real-time Monitoring:** The system should provide real-time monitoring and alerting for law enforcement agencies to respond to potential criminal activities.
- **Reporting**: The system should generate reports and visualizations to communicate the predicted crime hotspots and trends to relevant authorities.

3.5 NON FUNCTIONAL REQUIREMENTS

- **Performance Requirement:**
  The system should be able to process and analyze large volumes of data efficiently to provide timely and accurate crime predictions. It should also be capable of real-time monitoring and alerting for law enforcement agencies to take proactive measures.
- **Software Quality Attribute:**
  The system should exhibit high reliability, ensuring that it consistently delivers accurate crime predictions. Additionally, it should prioritize security to safeguard sensitive crime-related data and maintain the system's integrity.
- **Maintainability:**
  The system should be designed with modularity and documentation to facilitate easy maintenance and future enhancements. It should also adhere to coding standards and best practices to ensure that it can be updated and extended without compromising its functionality.
- **Usability:**
  The user interface should be intuitive and user-friendly, allowing law enforcement personnel to interact with the system seamlessly. The system should provide clear and

comprehensive visualizations of crime data and predictions for easy interpretation and decision-making.

- **Security:**
  Robust security measures should be implemented to safeguard sensitive crime-related data and prevent unauthorized access. The system should adhere to data privacy regulations and ensure the confidentiality and integrity of stored data.
- **Reliability:**
  The system should exhibit high availability, minimizing downtime and ensuring continuous access for law enforcement personnel. It should be resilient to failures and capable of recovering from potential system disruptions.
- **Scalability:**
  The system should be designed to accommodate potential growth in data volume and user base without significant performance degradation. It should support scalability to handle increasing computational demands as the system usage expands.
- **Compliance:**
  The system should comply with relevant industry standards, regulations, and legal requirements pertaining to crime data handling and analysis. It should adhere to ethical guidelines and ensure fairness and impartiality in crime prediction and analysis.

Chapter 4

METHODOLOGY AND MODULES

4.1 GENERAL

This research aims to predict crime and understand the factors and areas that lead to higher crime rates. The methodology should be divided into components to explain the processes and create opportunities for future work and improvement. The crime prediction model analyzes spatio-temporal aspects and features that influence crime. The data cube recorded over a city area is processed as an image sequence to output a predicted image or classification for the future. Data is being collected from the public Chicago Data Portal. Predictive analysis and data-driven decisions at the city level are recommended.

4.2 PROBLEM DEFINITION

The challenge is to develop an accurate crime prediction system using machine learning and diverse data sources to forecast crime hotspots and trends while overcoming the limitations of existing methods. This involves leveraging 2D convolutional neural networks (2D-CNNs) and diverse data sources to provide a more precise crime prediction method, ensuring reliability, efficiency, and maintainability.

4.3 SYSTEM ARCHITECTURE

To develop a crime prediction model, input data is collected from various sources and pre-processed. Feature engineering techniques are applied to extract relevant features, and the data is divided into training and testing datasets. The 2D Convolutional Neural Network (CNN) algorithm is then trained on the training data, with its parameters iteratively adjusted to minimize prediction errors. The testing data is utilized to evaluate the trained model's performance. The CNN architecture comprises several layers, including a conventional layer for initial feature extraction, a pooling layer for downsampling the feature maps, and dense (fully connected) layers that combine the learned features to generate the final prediction output. The objective is to develop an accurate and robust crime prediction system through the use of the 2D-CNN algorithm trained on the feature-engineered data.

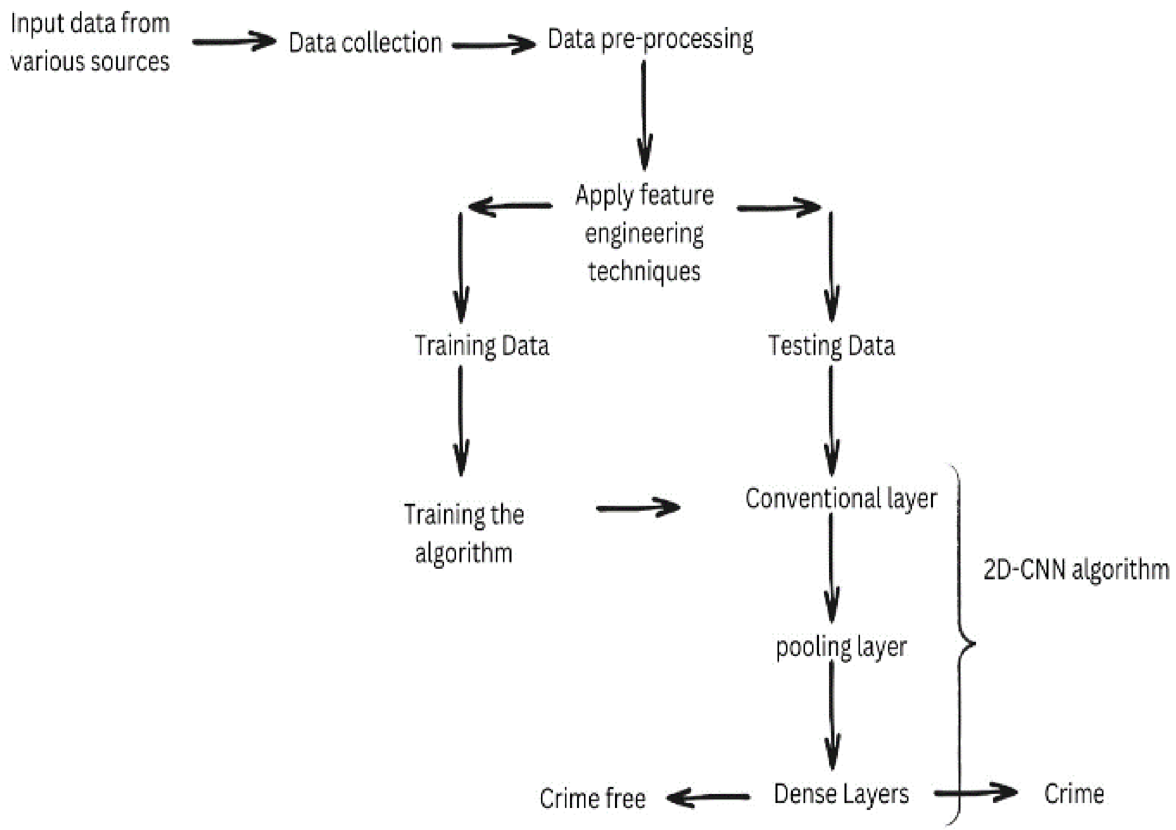


Figure: 4.3.1 System Architecture

4.4 METHODOLOGY

The methodology involves integrating machine learning techniques, particularly 2D convolutional neural networks (2D-CNNs), with diverse data sources to analyze and predict crime hotspots and trends. The approach also includes data collection, preprocessing, feature extraction, model training, and evaluation.

4.4.1 METHODOLOGY USED

The methodology employed for the research encompassed an extensive process, beginning with the comprehensive collection of diverse data sources. This involved the aggregation of detailed crime incidence reports, meteorological data, and demographic information. The integration of these varied datasets was followed by thorough preprocessing, which included feature engineering, data normalization, and standardization to ensure the uniformity and compatibility of the integrated data. The utilization of 2D Convolutional Neural Networks (2D-CNNs) played a pivotal role in the analysis, with the integrated data being transformed into 2D images, such as heat maps or choropleth maps, to capture intricate spatial and temporal patterns. Furthermore, transfer learning was applied to refine the trained 2D-CNN by incorporating knowledge from previous tasks, thereby enhancing the accuracy and predictive capabilities of the system. Additionally, the research placed significant emphasis on the identification and mitigation of potential biases within the data, ensuring the impartiality and fairness of the crime prediction system. Following a meticulous evaluation process, which included the use of standard performance metrics and cross-validation techniques, the methodology prioritized ethical considerations such as plagiarism prevention and the assurance of system impartiality.

4.4.2 LIST OF MODULES

1. Data Collection Module

2. Data Preprocessing Module

3. Feature Extraction Module

4. Machine Learning Model Training Module

5. Model Fitting Evaluation Module

6. Integration of Non-Crime Data

7. Prediction and Analysis

4.4.2.1 DATA COLLECTION

This initial phase entails the systematic gathering of pertinent data essential for the machine-learning endeavor. The efficacy of subsequent model performance hinges upon both the quality and quantity of the amassed data. This process encompasses the acquisition of datasets from diverse origins, including databases, and APIs, as well as the manual curation and annotation of data points.

4.4.2.2 DATA PRE-PROCESSING

Three major stages were completed on this dataset consignment of their job: determining characteristics, using a classifier to detect fraud, and pre-processing the dataset. For these adjustments, html elements and ‘data-‘noise were removed to keep the text ‘picture’ as a whole. The number of attributes was lowered efficiently and effectively through the application of the feature selection technique. They then utilized the support vector machine for the selection of test data. The engine’s total training time in the parser is revealed using the ensemble classifier, which employs the random forest to identify false job postings. The majority voting method showed that the tree-structured classifier was a random forest classifier. This classifier performed as an ensemble classifier had a classification accuracy of 97.4% to detect duplicated job postings.

4.4.2.3 FEATURE EXTRACTION

To automatically extract important features from preprocessed data, you can use a 2D convolutional neural network (CNN). These networks are highly effective at extracting spatial features and patterns, which can be particularly useful for crime data that may have geographical or spatial components

4.4.2.4 MODEL TRAINING

To create a robust 2D convolutional neural network (CNN) model, it is necessary to partition the data into two distinct subsets - training and validation. Once this has been accomplished, the model can be trained on the training set by utilizing the extracted features as input. It is then crucial to fine-tune the model's hyperparameters, including the learning rate, number of layers, and activation functions, to optimize its overall performance. This iterative process of adjusting the hyperparameters is vital to achieving the best possible results and ensuring the model's effectiveness in classification tasks.

4.4.2.5 MODEL FITTING EVALUATION

It is imperative to assess the accuracy and performance of the trained model on validation data. Furthermore, it is essential to compare the performance of the Convolutional Neural Network (CNN) model with other distinguished algorithms such as random forests, support vector machines, or traditional neural networks. Such a comparative analysis would help identify the strengths and limitations of the CNN model and make informed decisions. Therefore, it is crucial to evaluate the model's accuracy and performance on validation data and conduct a comprehensive analysis to determine its efficacy compared to other models.

4.4.2.6 INTEGRATION OF NON-CRIME DATA

To enhance the predictive capabilities of a model, it is recommended to include supplementary non-crime datasets, such as demographic, socioeconomic, or environmental data, in the feature set. The incorporation of these additional data sources has the potential to offer valuable context and improve the accuracy of the model's predictions. Thus, it is imperative to consider the inclusion of such data sets when designing the feature set of a model.

4.4.2.7 PREDICTION AND ANALYSIS

Utilize the pre-trained convolutional neural network (CNN) model to predict outcomes on novel and unseen data. Evaluate the predictions and interpret the outcomes, taking into account the spatial and contextual patterns learned by CNN. Visualize the predictions and insights, possibly utilizing geographical information systems (GIS) or other mapping tools.

Chapter 5

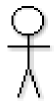
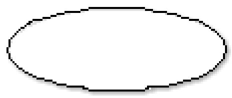
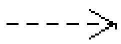
SYSTEM DESIGN

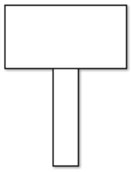
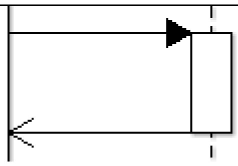
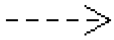
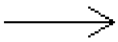
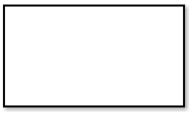
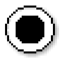
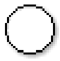
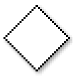
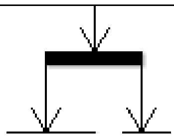
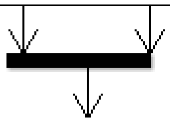
5.1 GENERAL

This chapter deals with unified UML (Unified Modeling Language). UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form, UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

5.2 UML NOTATION

In modeling, UML notations are the most crucial components. Using notations appropriately and efficiently is crucial to creating a comprehensive and relevant model. There are various notations for objects and relationships. Additionally, the notations of objects and relationships are used to create UML diagrams.

NAME	SYMBOL	DESCRIPTION
Actor		Internal or external entity that interacts with the system
Use Case		Use case is used to capture high level functionalities of a system.
Dependency		A semantic relationship between two elements in which a change to one element may affect the meaning of other
Association		A structural relationship describing a set of links connected between objects.

Generalization		A relationship in which objects of a specialized element (child) are substitutable for objects of a generalized element (parent).
Classifier Role		A classifier describes a set of instances that have common behavioral and structural features.
Call Action		This action send message and wait for a reply. The reply message is send back to that message.
Send Action		Create message is sent to a lifeline to create itself.
Return Action		Send message and proceed immediately without waiting for return value.
Classifier Role		A classifier describes a set of instances that have common behavioral and structural features.
Initial State		It shows the starting point of a process.
Final State		It shows the end of a process.
Choice		Collection of operation that specifies a service of class.
Junction		A diamond represents a decision with alternate paths.
Fork		A synchronization bar helps illustrate parallel transitions
Join		A synchronization bar helps illustrate parallel transitions

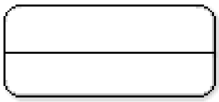
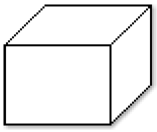
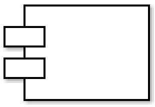
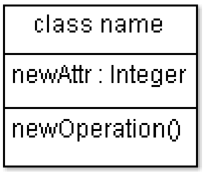
State		States represent situations during the life of an object.
Action		Action states represent the noninterruptible actions of objects.
Node		A node represents a physical component of the system. Node is used to represent physical part of a system like server, network etc.
Component		Component is used to represent any part of a system for which UML diagrams are made.
Class		The top section is used to name the class. The second one is used to show the attributes of the class. The third section is used to describe the operations performed by the class.
Aggregation		It is a special type of association depicting whole part relationship.
Composition		Composition is a special type of aggregation that denotes a strong ownership between Classes, the whole and its part.

Table:- 5.2.1 UML Notations

5.3 USE CASE DIAGRAM

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

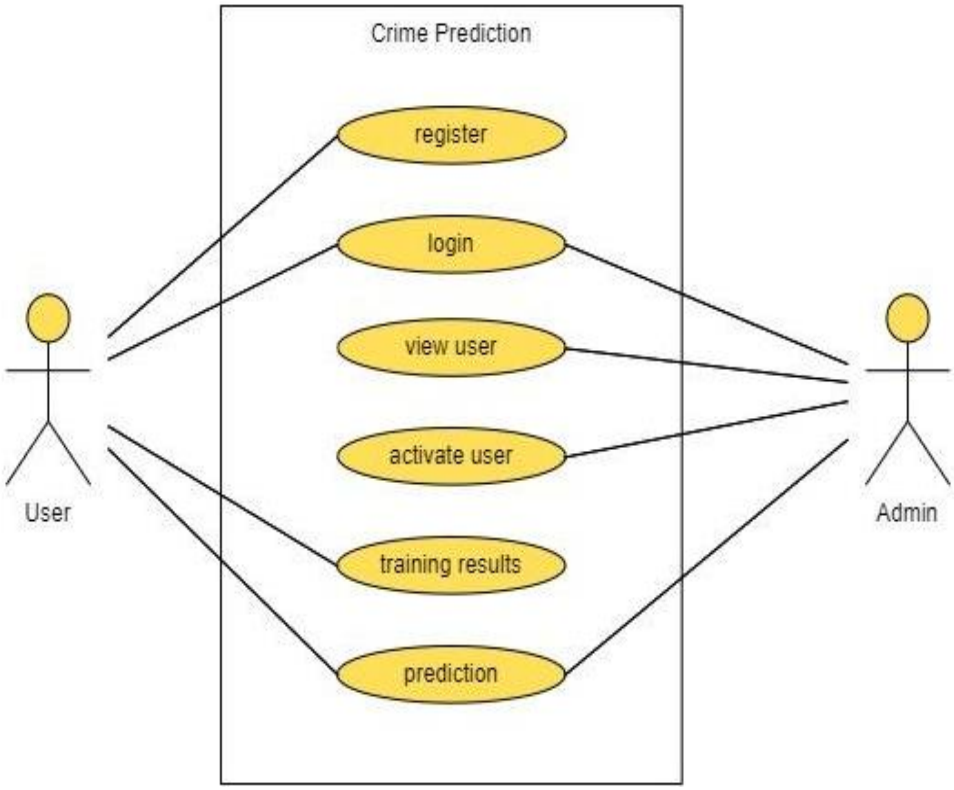


Figure:- 5.3.1 Use case diagram

5.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

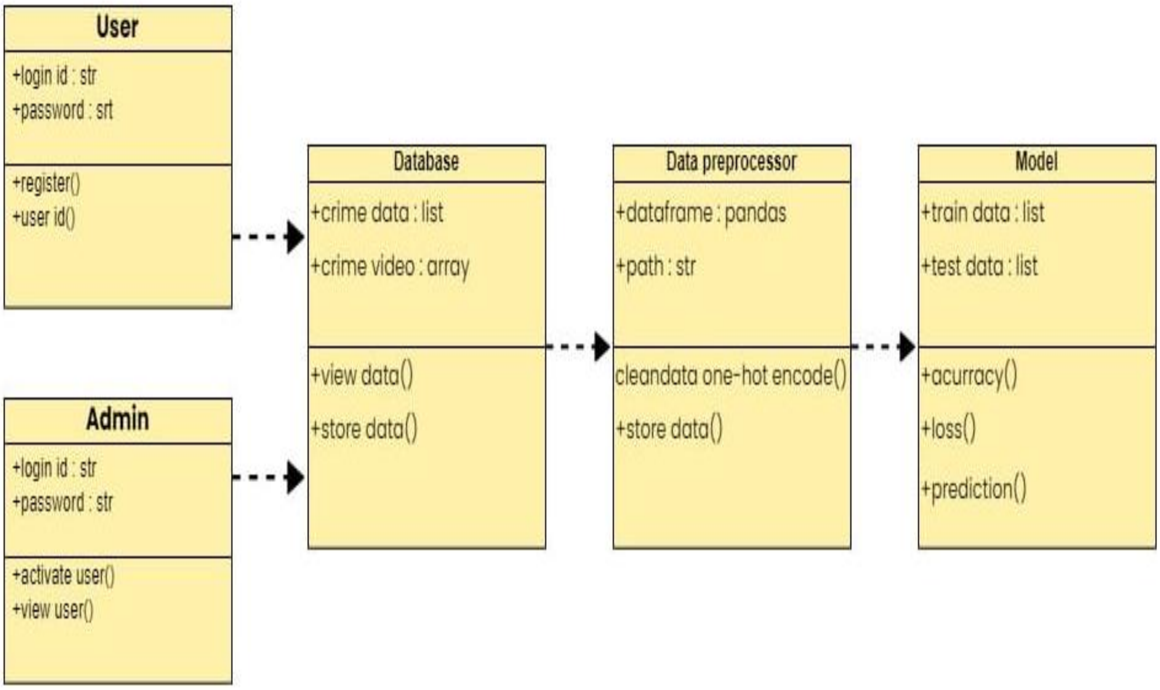


Figure:- 5.4.1 Class diagram

5.5 OBJECT DIAGRAM

An object diagram in UML showcases the relationships and attributes of specific instances of classes within a system at a particular point in time. It provides a snapshot of the runtime state, aiding in understanding system structure, object relationships, and interactions in real-time scenarios..

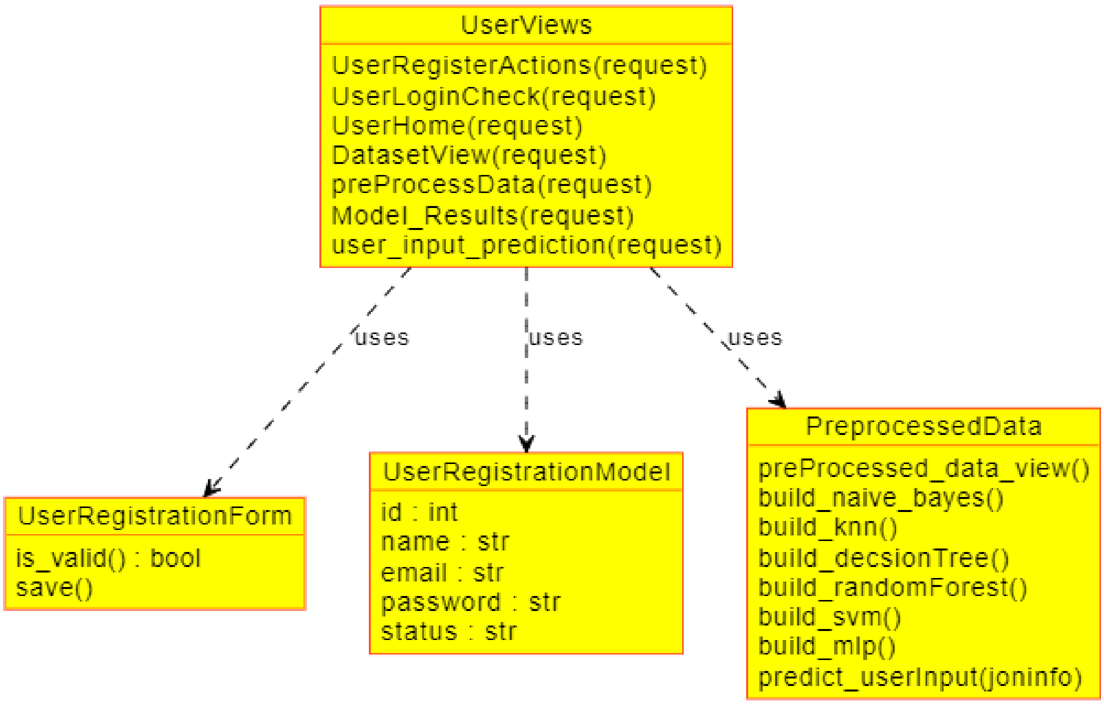


Figure:- 5.5.1 Object diagram

5.6 STATE DIAGRAM

A state diagram is a visual representation of the different states of an object or system, showing transitions triggered by events. It's a tool used in software engineering to model and analyze system behavior.

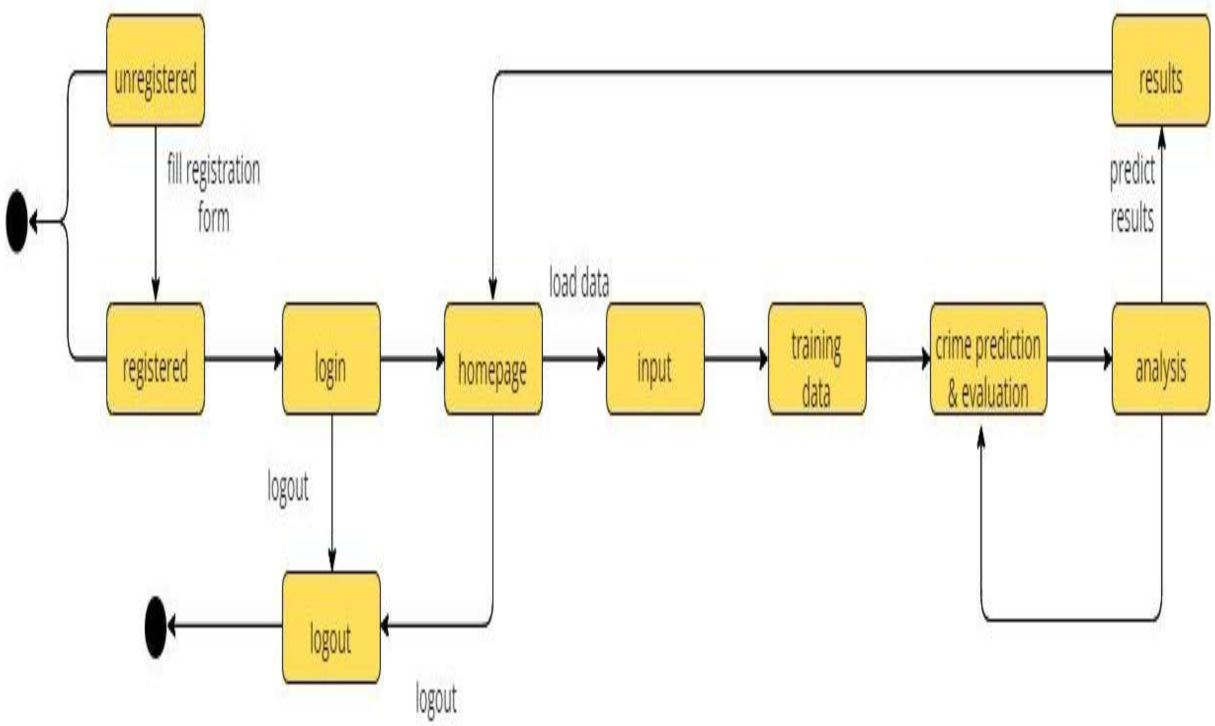


Figure:- 5.6.1 State diagram

5.7 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

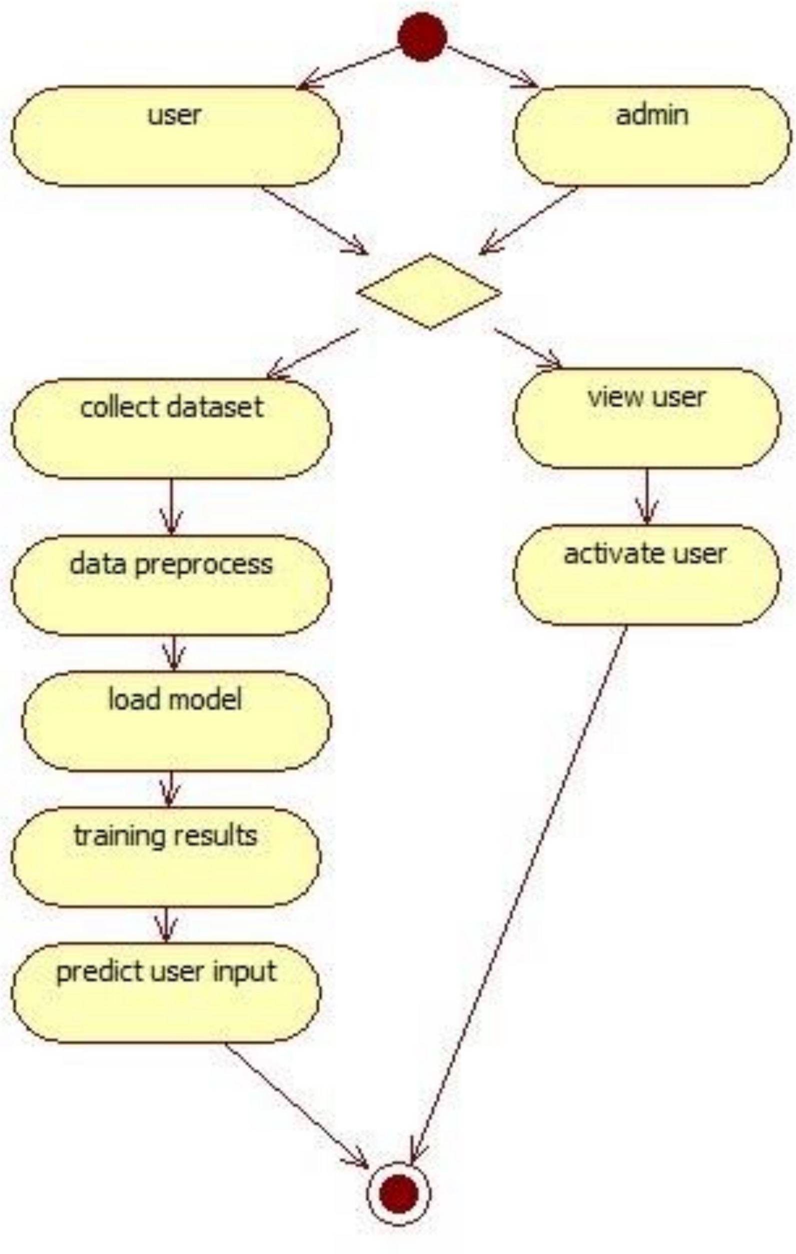


Figure:- 5.7.1 Activity diagram

5.8 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

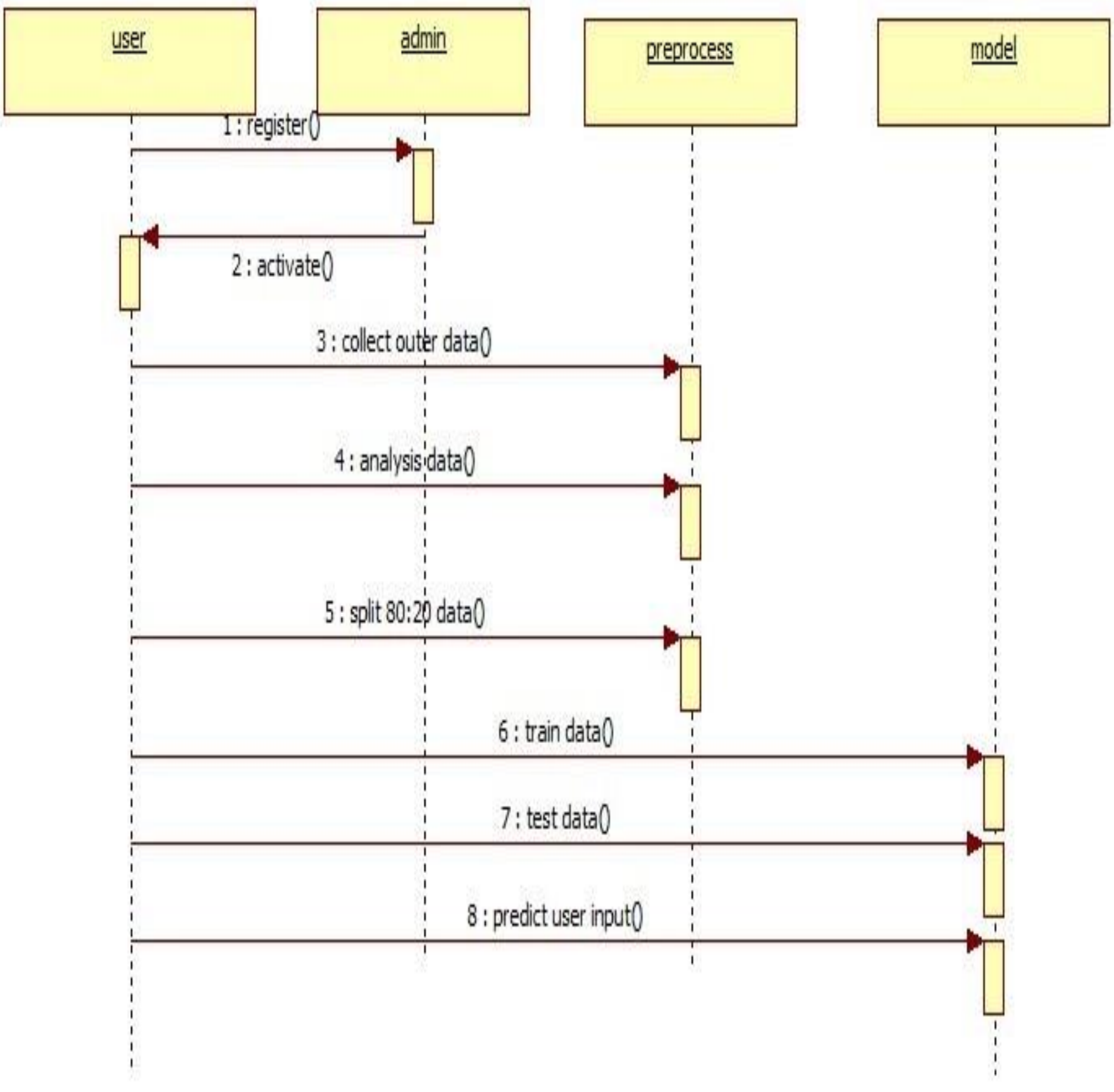


Figure:- 5.8.1 Sequence diagram

5.9 COMPONENT DIAGRAM

A component diagram showcases the organization of software components, their interactions, and dependencies within a system. It highlights how these parts work together to deliver the system's functionality.

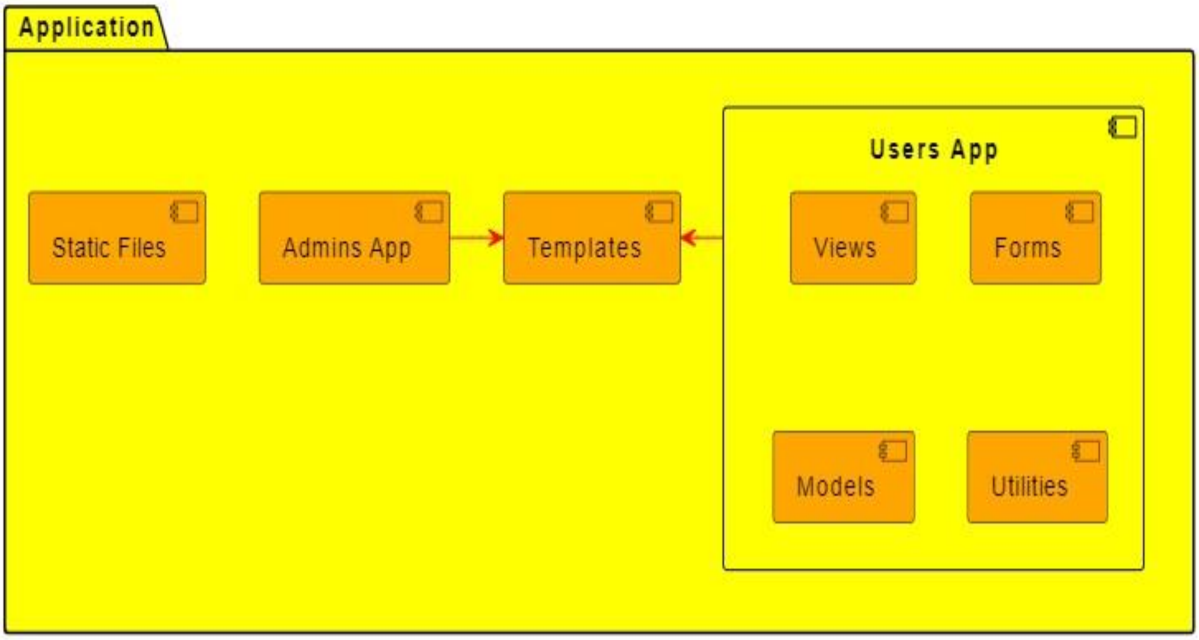


Figure:- 5.9.1 Component diagram

5.10 DEPLOYMENT DIAGRAM

A deployment diagram is a visual representation that shows how software components are deployed onto hardware nodes within a system, indicating their physical arrangement and interconnections. It helps in understanding the system's architecture, hardware requirements, and network configurations.

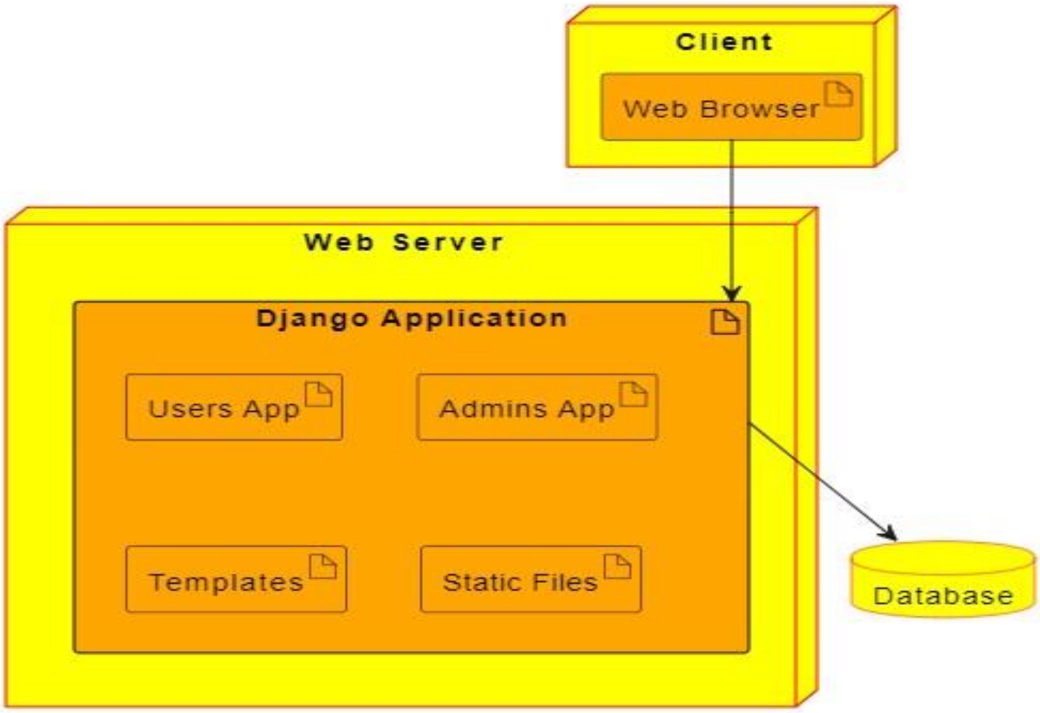


Figure:- 5.10.1 Deployment diagram

5.11 DATA FLOW DIAGRAM

DFDs, also called bubble charts, depict how data moves through a system, illustrating processes, data storage, and interactions with external sources. They're hierarchical, offering varying levels of detail and focusing on data flow rather than specific components or programming logic.

Level – 0

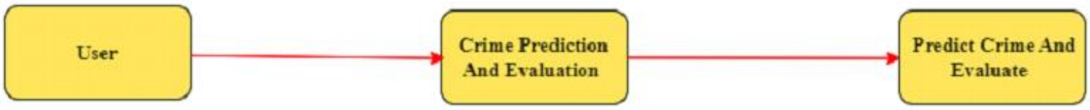


Figure:- 5.11.1 Data Flow diagram level - 0

Level – 1

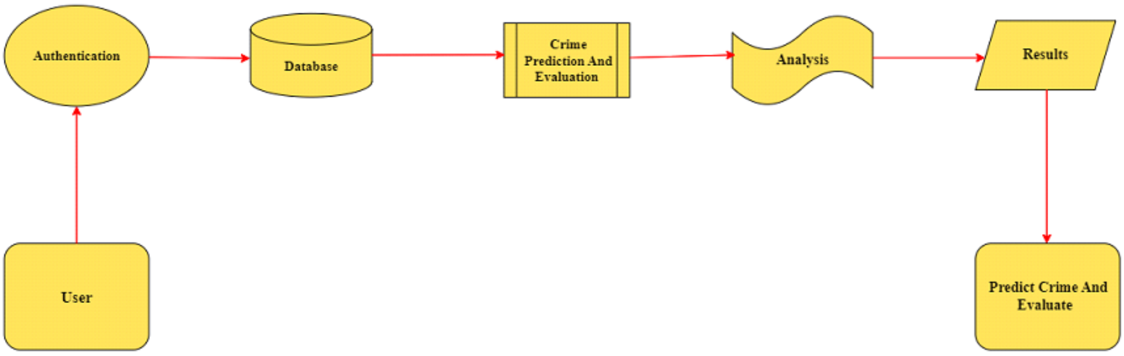


Figure:- 5.11.2 Data Flow diagram level – 1

Level – 2

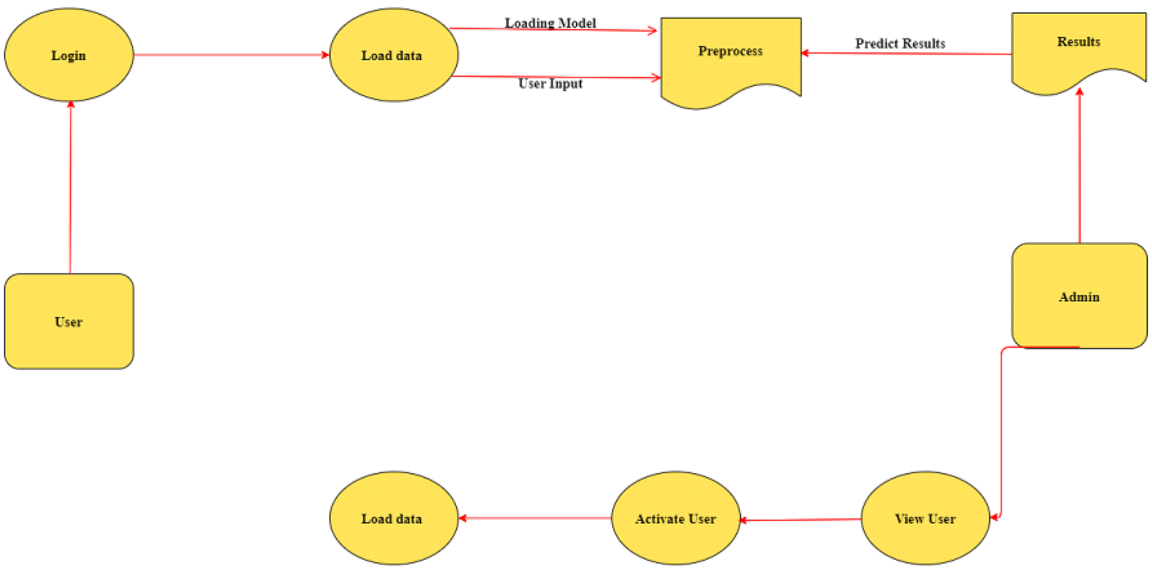


Figure:- 5.11.3 Data Flow diagram level - 2

Chapter 6

IMPLEMENTATION

6.1 GENERAL

Implementing a system from a DFD involves translating each process into functions, defining data structures for entities, managing data flow between these functions, handling external interactions, and ensuring a hierarchical code structure. Validate the code against the DFD specifications, refine it as needed, and optimize for efficiency and clarity.

The system's implementation is covered in detail in this chapter. It provides a thorough overview of the project's implementation steps.

6.2 SAMPLE CODE

USER SIDE VIEWS

```
# Create your views here.
from django.shortcuts import render
from django.conf import settings
from django.http import HttpResponse
from django.contrib import messages
from .forms import UserRegistrationForm
from .models import UserRegistrationModel
import pandas as pd
import numpy as np

# Create your views here.

def UserRegisterActions(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            print('Data is Valid')
            form.save()
            messages.success(request, 'You have been successfully registered')
```

```
            form = UserRegistrationForm()
            return render(request, 'UserRegistrations.html', {'form': form})
        else:
            messages.success(request, 'Email or Mobile Already Existed')
            print("Invalid form")
    else:
        form = UserRegistrationForm()
    return render(request, 'UserRegistrations.html', {'form': form})
def UserLoginCheck(request):
    if request.method =='POST':
        loginid=request.POST.get('loginid')
        pswd=request.POST.get('pswd')
        print("Login ID = ", loginid, ' Password = ', pswd)
        try:
            check = UserRegistrationModel.objects.get(
                loginid=loginid, password=pswd)
            status = check.status
            print('Status is = ', status)
            if status == "activated":
                request.session['id'] = check.id
                request.session['loggeduser'] = check.name
                request.session['loginid'] = loginid
                request.session['email'] = check.email
                print("User id At", check.id, status)
                return render(request, 'users/userhome.html', {})
            else:
                messages.success(request, 'Your Account Not at activated')
                return render(request, 'userlogin.html')
        except Exception as e:
            print('Exception is ', str(e))
            pass
messages.success(request, 'Invalid Login id and password')
        return render(request, 'userlogin.html', {})
```

```python
def UserHome(request):
    return render(request, 'users/userhome.html', {})

def viewData(request):
    import pandas as pd
    from django.conf import settings
    import os
    path=os.path.join(settings.MEDIA_ROOT,'chicago_crime_2014.csv')
    df=pd.read_csv(path)
    df=df.to_html
    # path =os.path.join(settings.MEDIA_ROOT,'chicago_crime_2015.csv')
    # auto_df = pd .read_csv(path)
    # auto_df = auto_df.to_html
    # path =os.path.join(settings.MEDIA_ROOT,'chicago_crime_2016.csv')
    # auto_df1 = pd .read_csv(path)
    # auto_df1 = auto_df1.to_html
    return render(request, 'users/userviewdata.html', {'data': df})

def preProcessData(request):
    from .utility.PreprocessedData import preProcessed_data_view
    data = preProcessed_data_view()
    return render(request, 'users/preproccessed_data.html', {'data': data})


def Model_Results(request):
    from .utility import PreprocessedData
    nb_report = PreprocessedData.build_naive_bayes()
    knn_report = PreprocessedData.build_knn()
    dt_report = PreprocessedData.build_decsionTree()
    rf_report = PreprocessedData.build_randomForest()
    svm_report = PreprocessedData.build_svm()
    mlp_report = PreprocessedData.build_mlp()
```

```python
    return render(request, 'users/ml_reports.html', { 'nb':nb_report,"knn":knn_report, 'dt':
dt_report, 'rf': rf_report, 'svm': svm_report,'mlp':mlp_report})

def user_input_prediction(request):
    if request.method=='POST':
        from .utility import PreprocessedData
        joninfo  = request.POST.get('joninfo')
        result = PreprocessedData.predict_userInput(joninfo)
        print(request)
        return render(request, 'users/testform.html', {'result': result})
    else:
        return render(request,'users/testform.html',{})
```

Base.html:

```html
{% load static %}
<!DOCTYPE html>
<html>
<head>
  <title>CrimePrediction</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    <style>
      body {

        background-image:url(../static/3.jpg);
        background-repeat: no-repeat;
        background-position:top;

        font-family: "Times New Roman", Georgia, Serif;}
      h1, h2, h3, h4, h5, h6 {
        font-family: "Times New Roman";
        letter-spacing: 6px;
```

```
      }
    </style>
</head>
<body>

<!-- Navbar (sit on top) -->
<div class="w3-top">
  <div class="w3-bar w3-yellow w3-padding w3-card" style="letter-spacing:4px;">
    <a href="#home" class="w3-bar-item w3-button"><h1><b>Crime Prediction</b></h1></a>
    <!-- Right-sided navbar links. Hide them on small screens -->
    <div class="w3-right w3-hide-small">
      <ul>
      <a href="{% url 'index' %}" class="w3-bar-item w3-button"><h4>Home</h4></a>
      <a href="{% url 'AdminLogin' %}"class="w3-bar-item w3-button"><h4>Admin</h4></a>
      <a href="{% url 'UserLogin' %}"  class="w3-bar-item w3-button"><h4>User</h4> </a>
                  <a href="{% url 'UserRegister' %}" class="w3-bar-item w3-button"><h4>Registration</h4></a>
      </ul>
    </div>
  </div>
</div>

<!-- Header -->
<header class="w3-display-container w3-content w3-wide" style="max-width:5000px;min-width:1500px" id="home">
    <img class="w3-image" src="/w3images/hamburger.jpg" alt="Hamburger Catering" width="1600" height="600">
  <div class="w3-display-bottomleft w3-padding-large w3-opacity">
    <h1 class="w3-xxlarge">Le Catering</h1>
  </div>
</header>

<!-- Page content -->
```

```
<div class="w3-content" style="max-width:3000px">

  <!-- About Section -->
  <div class="w3-row-100 w3-padding-64" id="about">
    <div class="w3-col m6 w3-padding-large w3-hide-small">
    </div>
  </div>
</div>


  <!-- Menu Section -->

{%block contents%}

{%endblock%}

<!-- Footer -->
<div>
<footer class="w3-center w3-light-green w3-padding-30">
   <p>&copy;  <a  href="https://www.w3schools.com/w3css/default.asp"  title="Ameeruddin"
target="_blank" class="w3-hover-text-green">SIMRAN & RAVEENA & AFEEFA</a>
</p>
</footer>
</div>
<!-- /.javascript files -->
<script src="{%static 'js/jquery.js'%}"></script>
<script src="{%static 'js/bootstrap.min.js'%}"></script>
<script src="{%static 'js/custom.js'%}"></script>
<script src="{%static 'js/jquery.sticky.js'%}"></script>
<script src="{%static 'js/wow.min.js'%}"></script>
<script src="{%static 'js/owl.carousel.min.js'%}"></script>
<script src="{%static 'js/ekko-lightbox-min.js'%}"></script>
<script type="text/javascript">
```

```
  $(document).delegate('*[data-toggle="lightbox"]', 'click', function (event) {
      event.preventDefault();
      $(this).ekkoLightbox();
  });
</script>
<script>
  new WOW().init();
</script>
</body>
</html>
```

Index.html:

```
{%extends 'base.html'%}
{% load static %}
{%block contents%}
<head>
    <link rel="stylesheet" href="assets/css">
</head>
<div class="page-header">
            <div class= "container" >
                <div class="row-50">
                    <div class="col-30" >

                        <h1 align="center" style="color:rgb(106, 10, 10)"><b>Crime
Prediction</b></h1>
                           <p>
                            <center><img src="../static/aiii.png" alt="reload" width="500"
height="400">
                            <p><b>Crime is one of the dominant and alarming aspects of our society.
Everyday huge number of crimes are committed, and these frequent crimes have made the
lives of common citizens restless. So, preventing the crime from occurring is a vital task. In
recent times, it is seen that artificial intelligence has shown its importance in almost all the
field and crime prediction is one of them.
                            </b></p>
```

```
                    </center>
                </p>
            </table>
            </div>
            <div class="col-30">
            </div>
        </div>
    </div>
</div>
{%endblock%}
```

Admin side views:

```python
from django.shortcuts import render
from django.contrib import messages
from users.forms import UserRegistrationForm
from users.models import UserRegistrationModel

# Create your views here.
def AdminLoginCheck(request):
    if request.method == 'POST':
        usrid = request.POST.get('loginid')
        pswd = request.POST.get('pswd')
        print("User ID is = ", usrid)
        if usrid == 'admin' and pswd == 'admin':
            return render(request, 'admins/adminHome.html')
        else:
            messages.success(request, 'Please Check Your Login Details')
    return render(request, 'Adminlogin.html', {})

def AdminHome(request):
    return render(request, 'admins/adminHome.html')

def RegisterUsersView(request):
```

```
    data = UserRegistrationModel.objects.all()
    return render(request,'admins/viewregisterusers.html',{'data':data})


def ActivaUsers(request):
    if request.method == 'GET':
        id = request.GET.get('uid')
        status = 'activated'
        print("PID = ", id, status)
        UserRegistrationModel.objects.filter(id=id).update(status=status)
        data = UserRegistrationModel.objects.all()
        return render(request,'admins/viewregisterusers.html',{'data':data})
```

6.3 SCREENSHOTS

6.3.1 General

A snapshot is a system state's frozen image, crucial for recovery and backups in software. It allows restoration to previous states and efficient error protection, requiring careful storage management.

6.3.2 Snapshot

Home Page

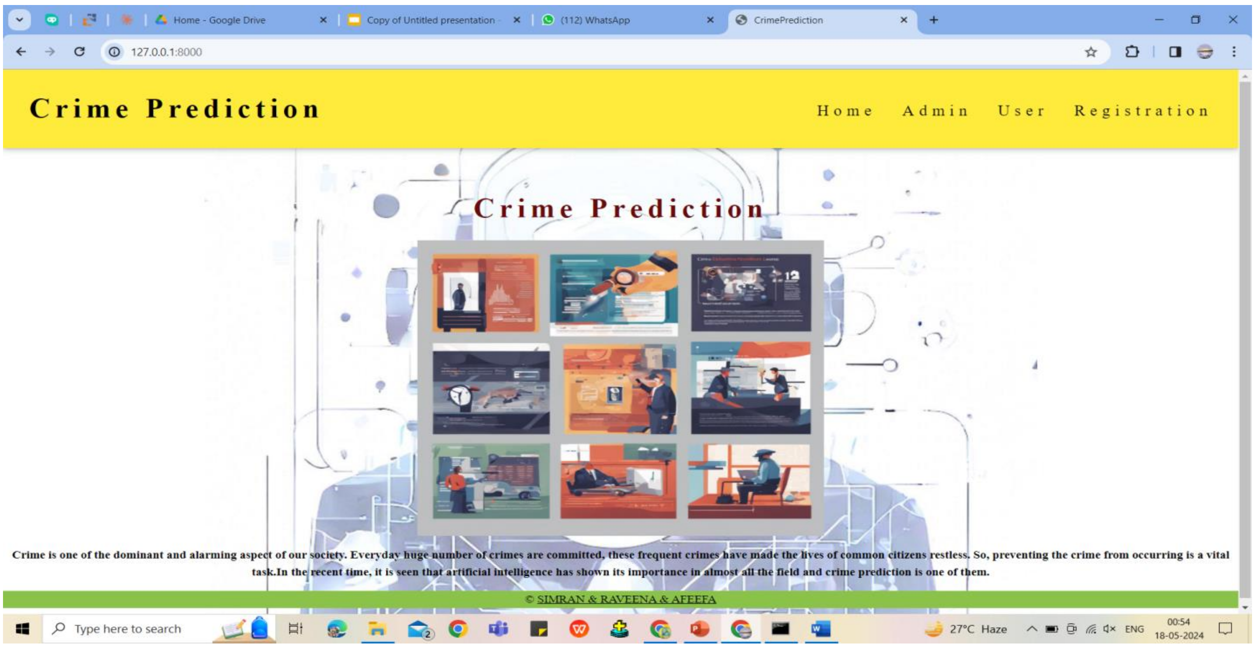


Figure:- 6.3.2.1 Homepage

Register Form

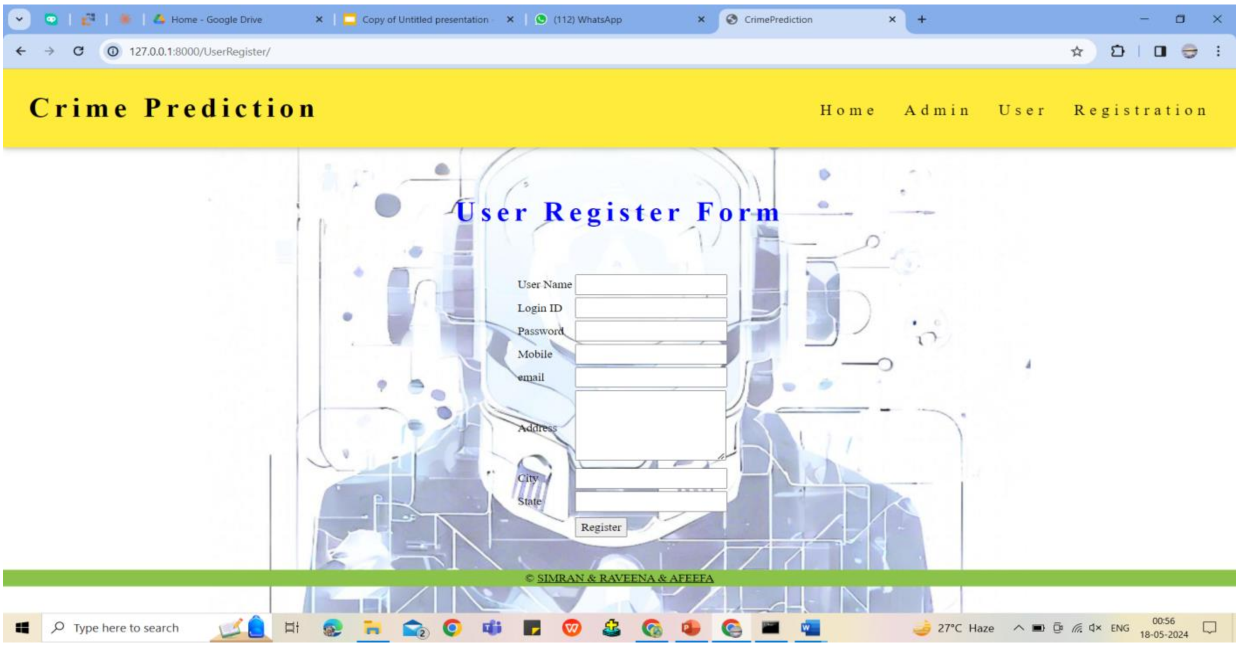


Figure:- 6.3.2.2 Register Form

Admin Login Form

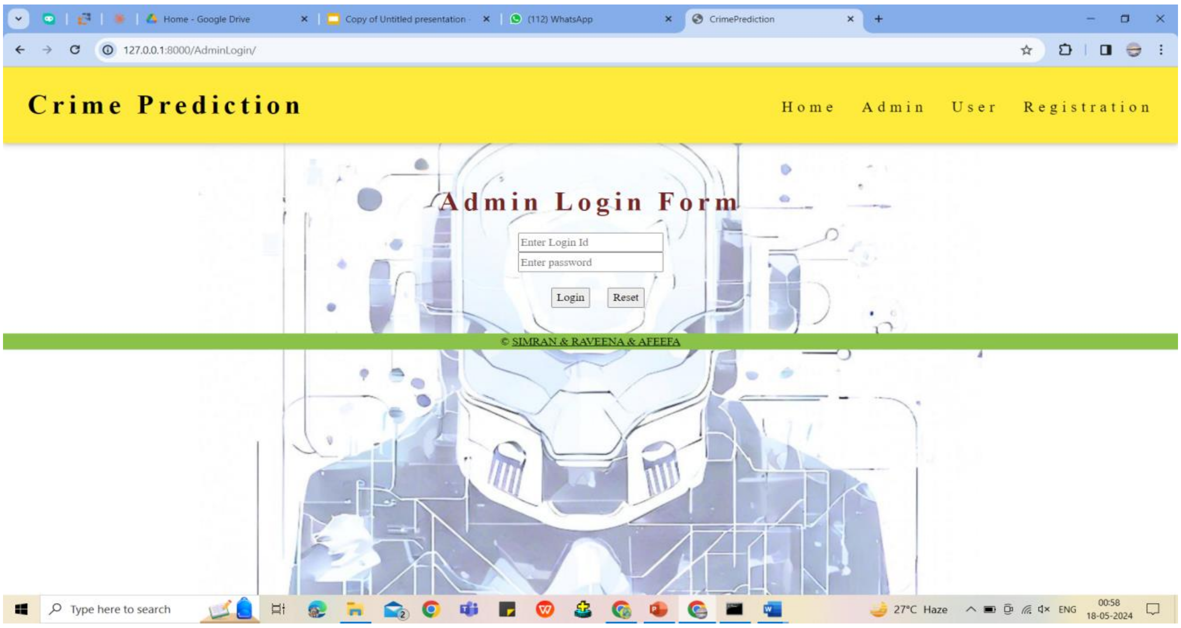


Figure:- 6.3.2.3 Admin Login Form

Admin Home Page

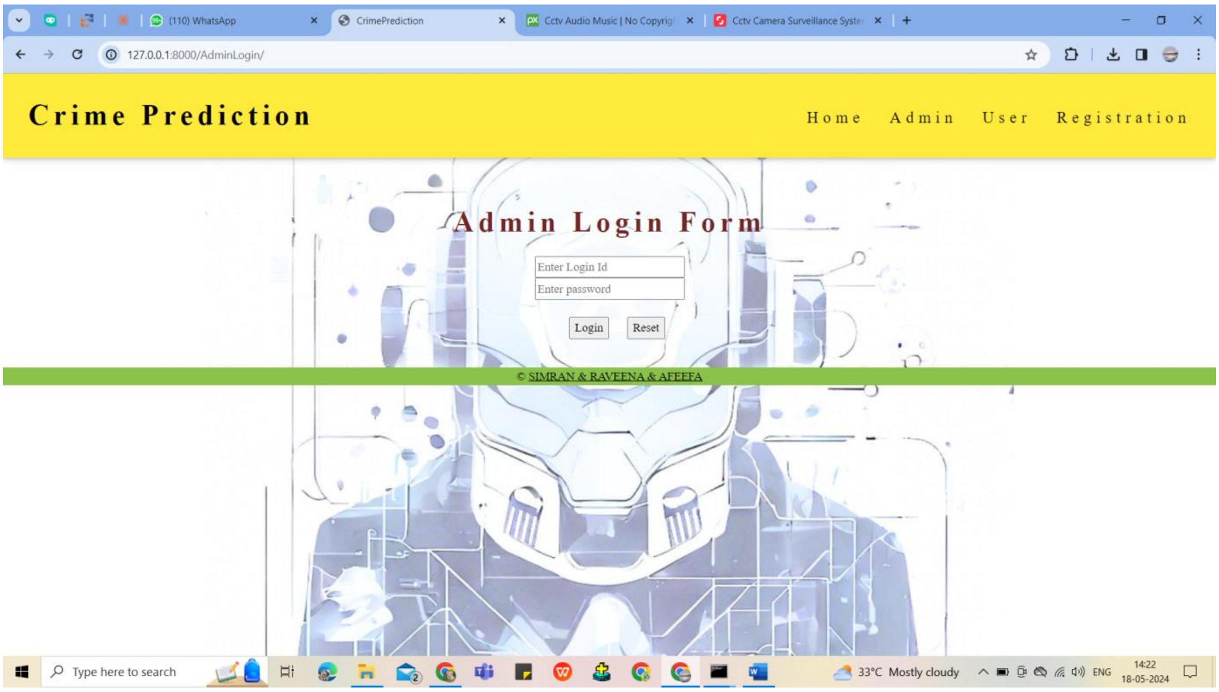


Figure:- 6.3.2.4 Admin Home Page

View users and Activate

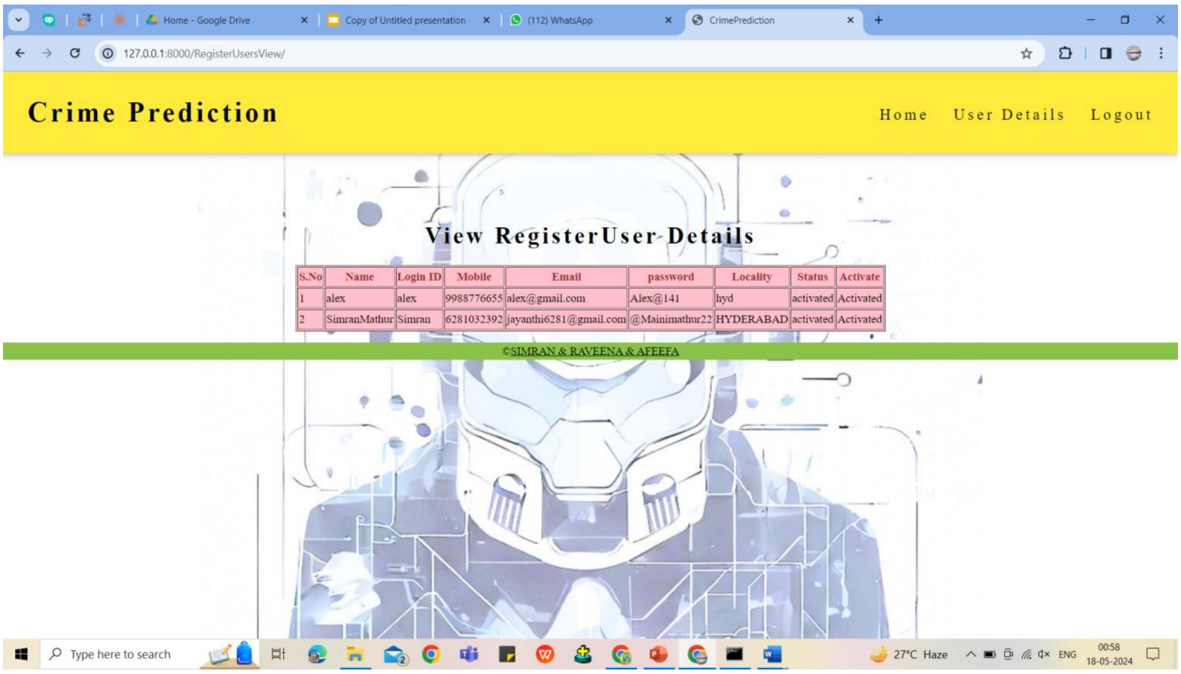


Figure:- 6.3.2.5 View users and Activate

User Login

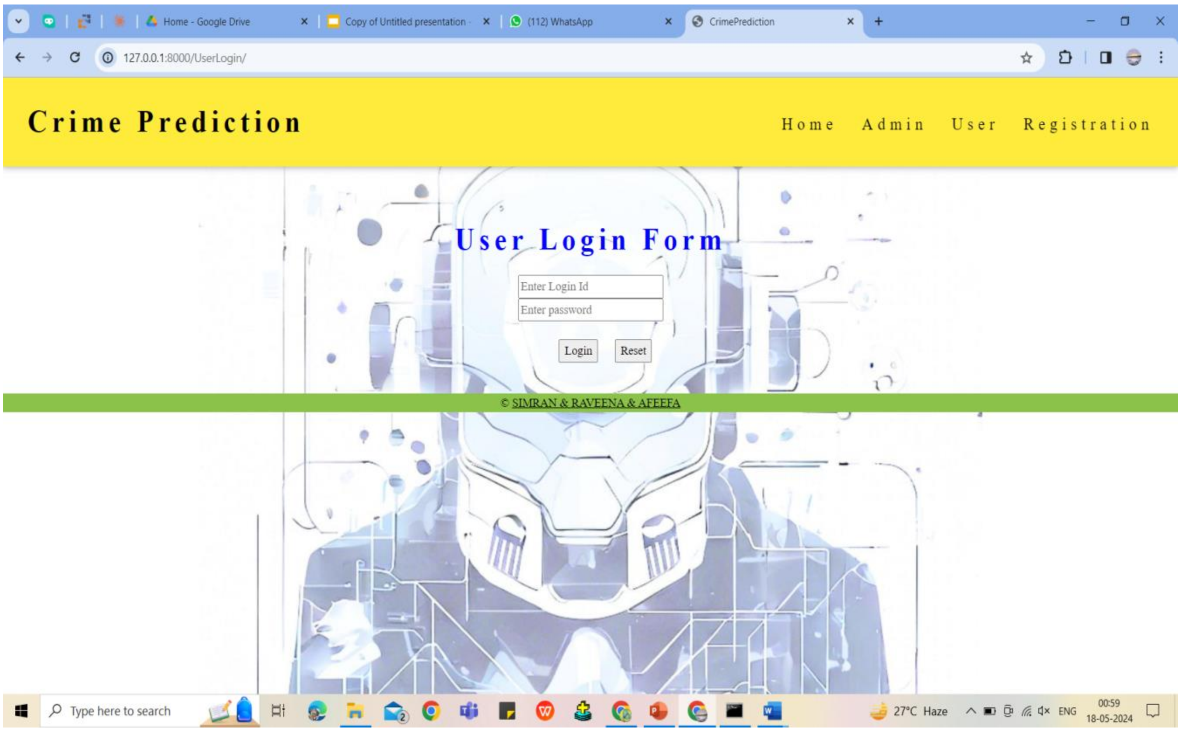


Figure:- 6.3.2.6 User Login

User HomePage

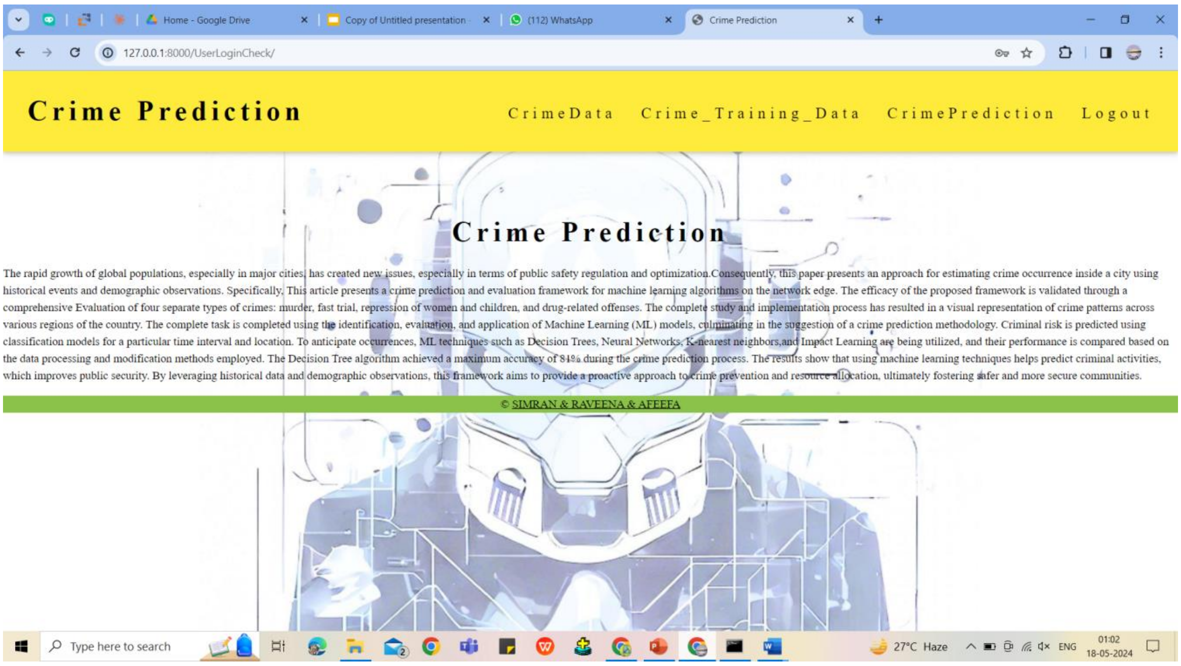


Figure:- 6.3.2.7 User HomePage

Training Data

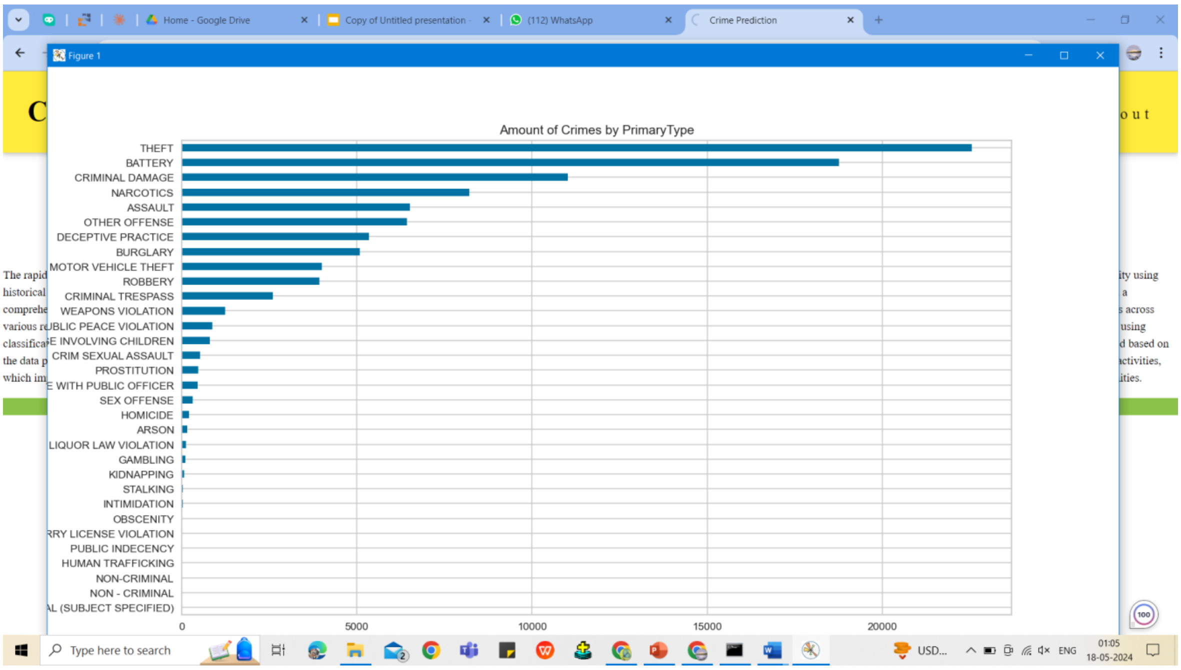


Figure:- 6.3.2.8 Training Data

User Training Data (Heatmap)

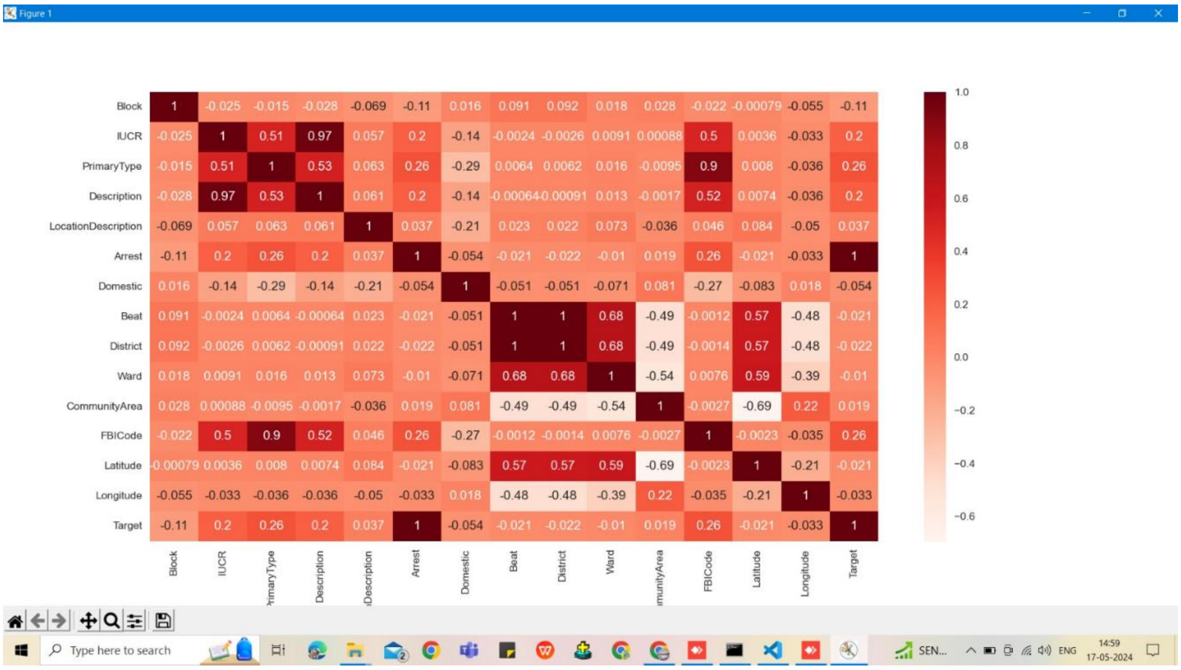


Figure:- 6.3.2.9 User Training Data (Heatmap)

Training Results

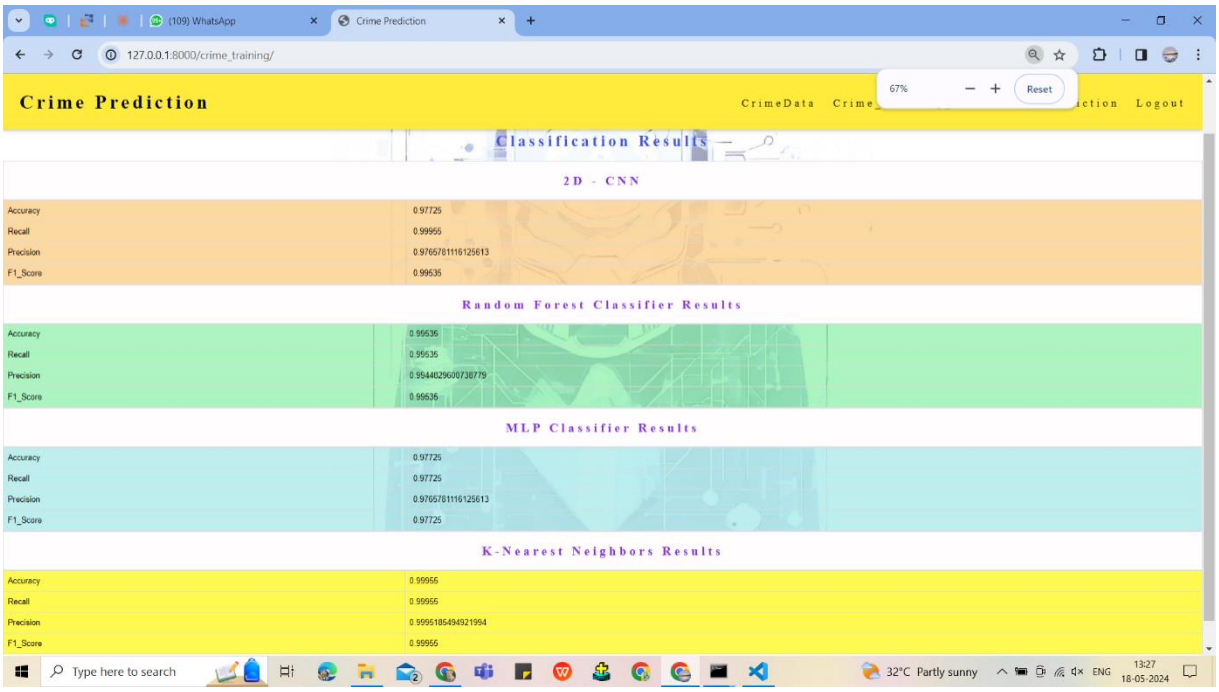


Figure:- 6.3.2.10 Training Results

Training Results Graph

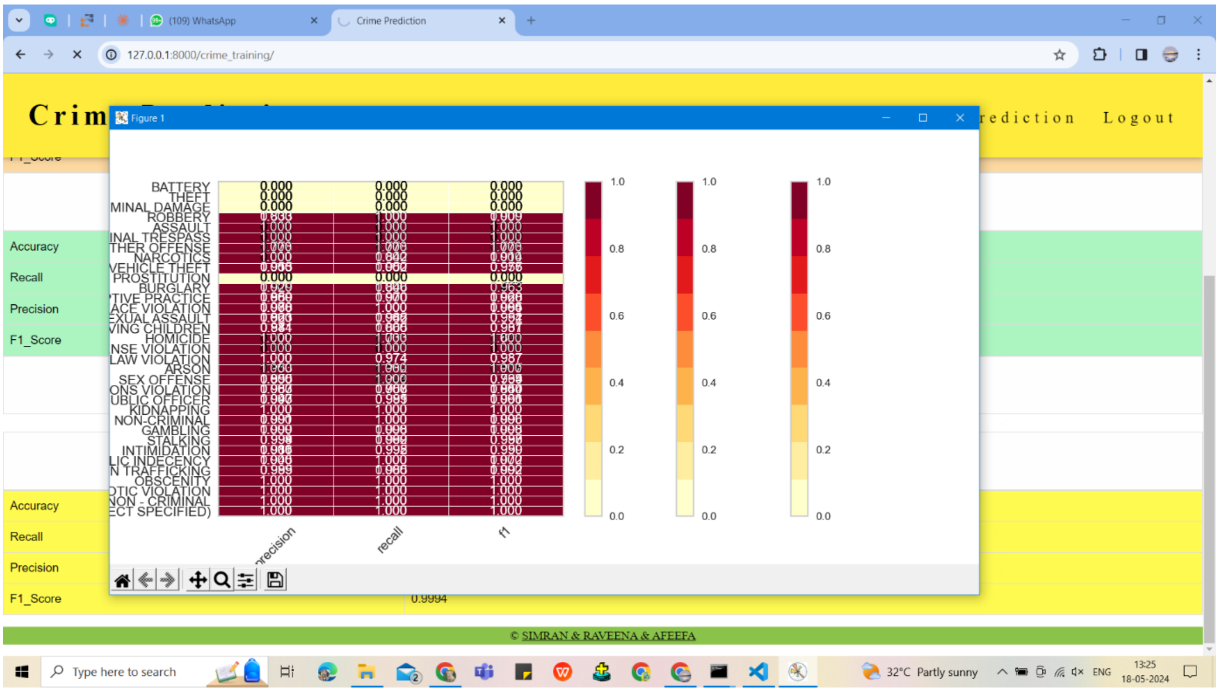


Figure:- 6.3.2.11 Training Results Graph

Crime Prediction Form

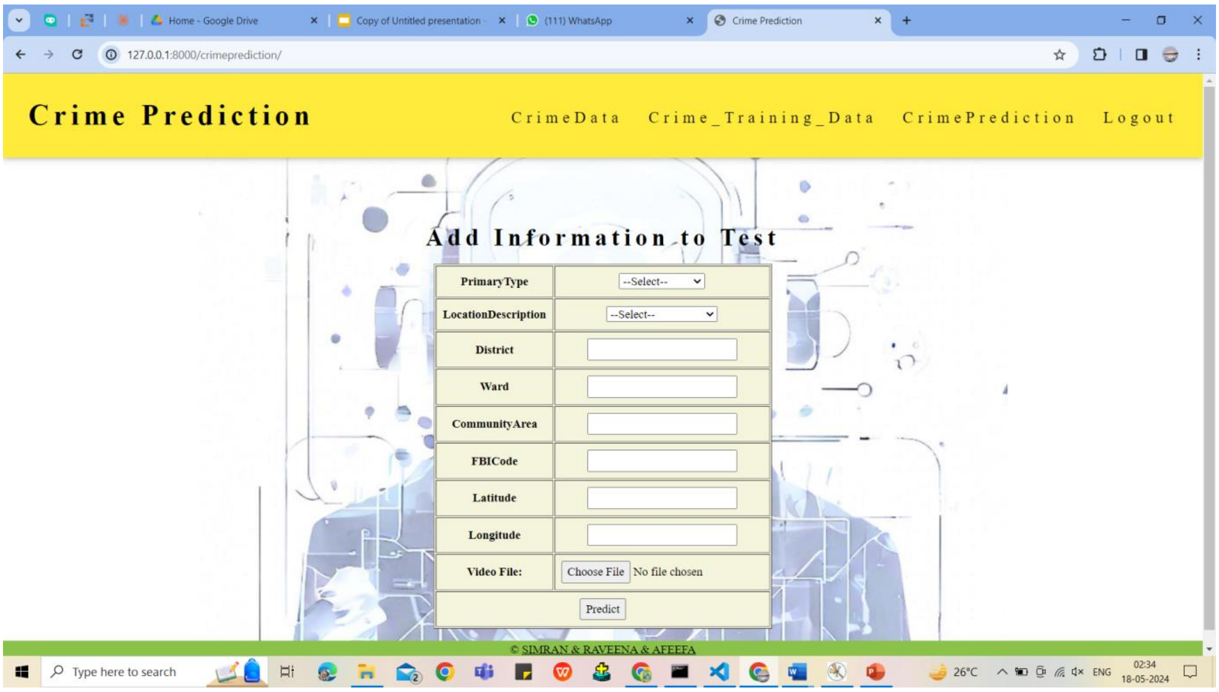


Figure:- 6.3.2.12 Crime Prediction Form

Crime Prediction Results(No Crime)

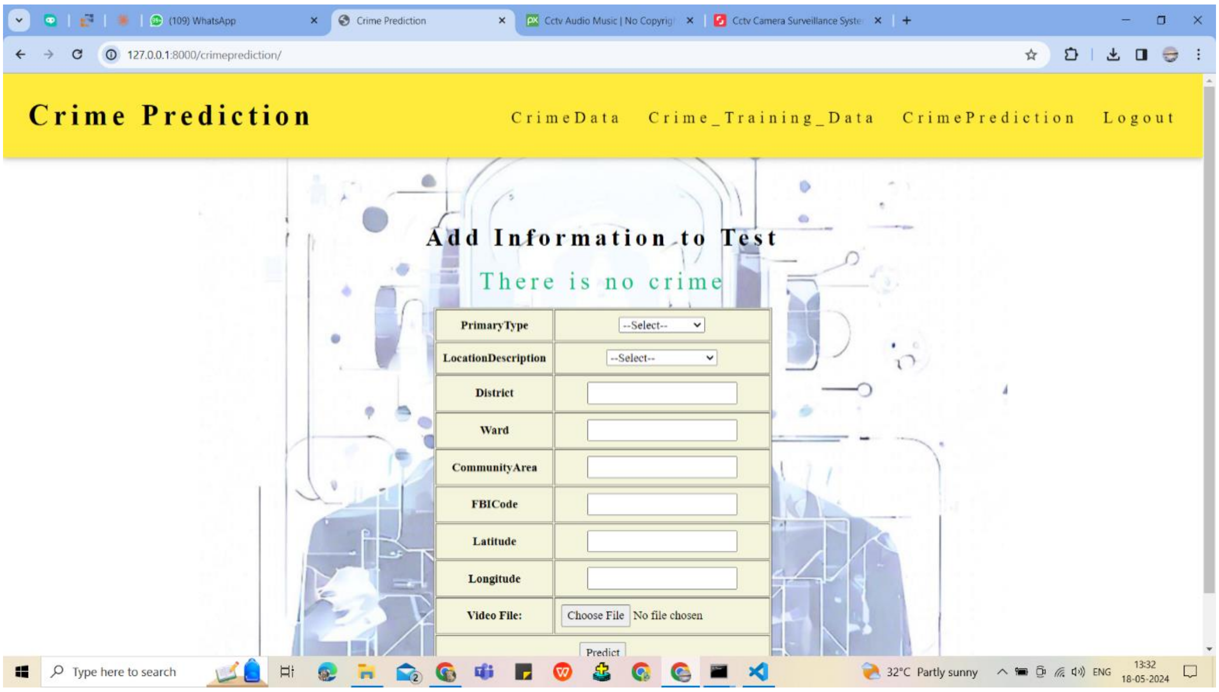


Figure:- 6.2.13 Crime Prediction Results(No Crime)

Crime Prediction Results(Crime)

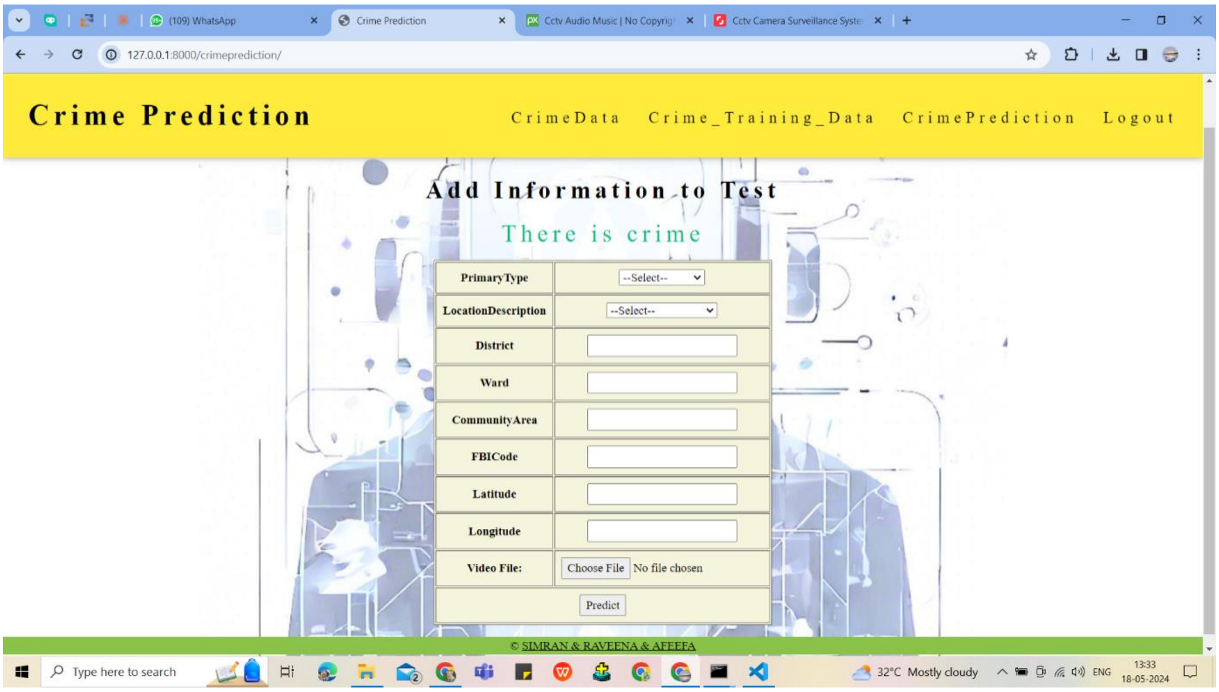


Figure:- 6.2.14 Crime Prediction Results(Crime)

Crime Data(View Dataset)

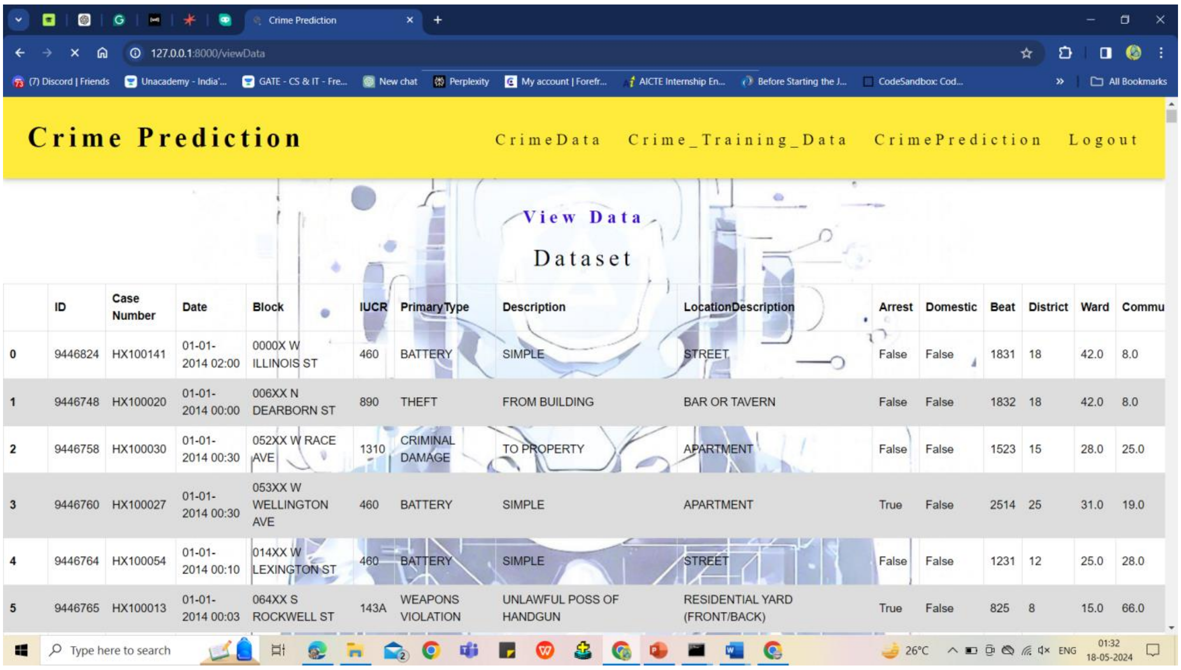
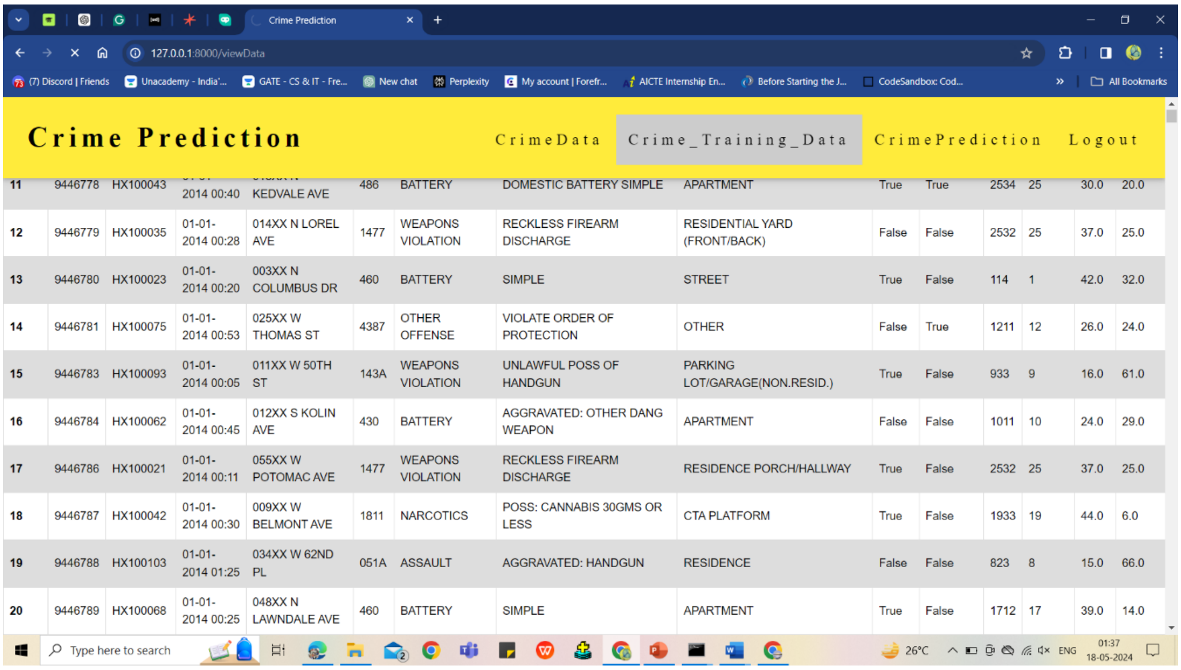


Figure:- 6.3.2.15 Crime Data(View Dataset)



11	9446778	HX100043	2014 00:40	KEDVALE AVE	486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	True	2534	25	30.0	20.0
12	9446779	HX100035	01-01-2014 00:28	014XX N LOREL AVE	1477	WEAPONS VIOLATION	RECKLESS FIREARM DISCHARGE	RESIDENTIAL YARD (FRONT/BACK)	False	False	2532	25	37.0	25.0
13	9446780	HX100023	01-01-2014 00:20	003XX N COLUMBUS DR	460	BATTERY	SIMPLE	STREET	True	False	114	1	42.0	32.0
14	9446781	HX100075	01-01-2014 00:53	025XX W THOMAS ST	4387	OTHER OFFENSE	VIOLATE ORDER OF PROTECTION	OTHER	False	True	1211	12	26.0	24.0
15	9446783	HX100093	01-01-2014 00:05	011XX W 50TH ST	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	PARKING LOT/GARAGE(NON.RESID.)	True	False	933	9	16.0	61.0
16	9446784	HX100062	01-01-2014 00:45	012XX S KOLIN AVE	430	BATTERY	AGGRAVATED: OTHER DANG WEAPON	APARTMENT	False	False	1011	10	24.0	29.0
17	9446786	HX100021	01-01-2014 00:11	055XX W POTOMAC AVE	1477	WEAPONS VIOLATION	RECKLESS FIREARM DISCHARGE	RESIDENCE PORCH/HALLWAY	True	False	2532	25	37.0	25.0
18	9446787	HX100042	01-01-2014 00:30	009XX W BELMONT AVE	1811	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	CTA PLATFORM	True	False	1933	19	44.0	6.0
19	9446788	HX100103	01-01-2014 01:25	034XX W 62ND PL	051A	ASSAULT	AGGRAVATED: HANDGUN	RESIDENCE	False	False	823	8	15.0	66.0
20	9446789	HX100068	01-01-2014 00:25	048XX N LAWNDALE AVE	460	BATTERY	SIMPLE	APARTMENT	True	False	1712	17	39.0	14.0

Figure:- 6.3.2.16 Crime Data(View Dataset)

Chapter 7
TESTING

7.1 GENERAL

Testing is the process of discovering errors and weaknesses in a work product. It verifies that all software functionalities have been correctly implemented and meet user expectations. The different types of tests address specific requirements.

Testing for crime prediction is a critical phase in the development of machine learning models aimed at forecasting criminal activities. The process involves a comprehensive evaluation of the predictive models to assess their effectiveness, accuracy, and reliability. Firstly, the data is meticulously prepared, involving cleaning, normalization, and segregation into training and testing sets. Subsequently, the models are trained using historical crime data to enable them to discern patterns and correlations within the dataset. Following this, the models undergo validation using the testing data to determine their ability to generalize to new, unseen data. Performance evaluation metrics such as accuracy, precision, recall, F1 score, and the area under the ROC curve are employed to gauge the model's effectiveness in predicting crime occurrences. Cross-validation techniques, including k-fold cross-validation, are utilized to ensure the consistency of the model's performance across different data subsets. Additionally, fine-tuning of the model may be undertaken based on the performance evaluation, involving the adjustment of hyperparameters to optimize predictive accuracy. Finally, the models are rigorously tested on completely unseen data to assess their real-world predictive capabilities, thereby ensuring they are well-suited for enhancing public safety and law enforcement strategies.

7.2 TEST PLAN

A testing plan for crime prediction involves outlining the specific steps and procedures for evaluating the effectiveness and accuracy of machine learning models developed for forecasting criminal activities. The plan typically includes detailed strategies for data preparation, model training, model validation, performance evaluation, cross-validation, fine-tuning, and testing on unseen data. Additionally, the plan may encompass the selection of appropriate performance metrics, the establishment of criteria for model acceptance, and the documentation of testing results. Furthermore, it may outline the roles and responsibilities of individuals involved in the testing process, as well as establish a timeline for each testing phase. The testing plan serves as a roadmap for systematically assessing the reliability and efficacy of the predictive models, ensuring that they meet the requirements for enhancing public safety and law enforcement strategies.

OVERVIEW	
Test plan objectives	- Validate the accuracy and reliability of the crime prediction model in identifying potential crime hotspots and trends. - Ensure the model's performance meets the specified requirements and user expectations. - Identify and address potential issues, errors, or limitations in the model's functionality. - Evaluate the model's robustness and ability to handle diverse data inputs and scenarios.
Testing Assumptions	- The test environment will closely mimic the production environment. - Adequate test data, including historical crime data, demographic information, and other relevant data sources, will be available. - Subject matter experts and domain knowledge will be accessible to assist in test case development and evaluation.
Risks and Contingencies	- Unavailability or inadequacy of test data may impact the testing process, leading to delays or incomplete testing scenarios. - Unforeseen issues or limitations in the model's performance may require additional development efforts, potentially affecting the project timeline. - Changes in requirements or stakeholder expectations during the testing phase may necessitate modifications to the test plan and test cases.

Table:- 6.2.1 Test Plan Overview

TEST SCOPE	
Features to be Test	- Data integration and preprocessing capabilities. - Feature extraction and selection techniques. - Model training and learning algorithms. - Prediction accuracy for various crime types and geographic areas. - Performance and scalability under different data volumes and scenarios. - User interface and reporting functionalities.
Features Not to be Tested	- External data sources and APIs (assumed to be functioning correctly). - Hardware and infrastructure components (assumed to be properly configured and maintained). - Administrative and user management functionalities (beyond the scope of this project).

Table:- 6.2.2 Test Scope

TEST METHODOLOGIES	
Testing Approach	- Black-box testing: Evaluating the model's functionality and behavior without knowledge of its internal structure or code. - White-box testing: Examining the model's internal components, algorithms, and data flow to identify potential issues or inefficiencies.

	- Integration testing: Verifying the seamless integration of different components and data sources within the crime prediction system. - Performance testing: Assessing the model's performance under various load and stress conditions. - Acceptance testing: Validating the model's compliance with specified requirements and stakeholder expectations.
Test Documents	- Test plan: This comprehensive document outlining the testing strategy, scope, and approach. - Test cases: Detailed scenarios and steps to be executed during testing, including expected results and pass/fail criteria. - Test data: Relevant crime data, demographic information, and other data sources required for testing. - Test reports: Documentation of test execution results, issues encountered, and overall test coverage.
Test Case Pass/Fail Criteria	- Pass criteria: The model accurately predicts crime hotspots and trends within the specified tolerance levels, and all functional requirements are met. - Fail criteria: The model fails to accurately predict crime hotspots and trends, or significant issues or deviations from the requirements are observed.
Suspension / Resumption Criteria	- Testing may be suspended in case of critical issues that prevent further testing or significant deviations from the requirements. - Testing will resume after the identified issues have been resolved and the necessary changes or improvements have been implemented.

Problem Logging / Resolution	- All issues, defects, or deviations from expected behavior will be logged in a centralized issue tracking system. - Detailed information, including steps to reproduce, expected and actual results, and severity levels, will be documented. - Issues will be assigned to the appropriate team members for investigation and resolution. - Resolved issues will be retested and verified before closure.

Table :- 6.2.3 Test Methodologies

7.3 TEST CASES

Test cases play a crucial role in the testing process of the crime prediction model. They serve as detailed scenarios and steps that guide the execution of tests, ensuring comprehensive coverage of the model's functionality and behavior. Well-structured test cases are essential for achieving reliable and consistent results during the testing phase. Here's a well-structured paragraph describing test cases:

Test cases are meticulously designed to cover a wide range of scenarios and conditions that the crime prediction model may encounter in real-world situations. Each test case comprises a set of clearly defined steps, including prerequisites, input data, expected results, and pass/fail criteria. These steps are organized in a logical sequence, facilitating easy comprehension and execution by the testing team. Test cases are typically grouped based on their scope and objectives, such as data preprocessing, feature extraction, model training, prediction accuracy, and user interface testing. They are documented in a standardized format, ensuring consistency and traceability throughout the testing process. Additionally, test cases may incorporate specific data sets, edge cases, and boundary conditions to thoroughly evaluate the model's robustness and ability to handle diverse inputs. By adhering to a well-structured approach in test case design and execution, the testing team can effectively identify and address potential issues, errors, or limitations in the crime prediction model, ultimately enhancing its reliability and accuracy in predicting crime hotspots and trends.

7.3.1 Unit testing

Unit testing validates program logic and inputs/outputs, identifies errors early, and is done at the component level. It's usually done in combination with code testing, but can also be a distinct phase.

Unit Testing – I

Test Case:	UTC - I
Name of the test:	Admin Login
Item Tested:	Login
Sample Input:	Admin details (Admin id, password)
Expected output:	Admin can log in with their login credentials. If successful he gets his home page.
Actual output:	Admin can’t log in with the wrong login credentials.
Remarks:	Failure

Table: 6.3.1.1 Unit Testing for Admin Login (Failure)

Unit Testing – II

Test Case:	UTC – II
Name of the test:	Admin can activate the users
Item Tested:	Details of users
Sample Input:	User details activation/status
Expected output:	Admin can activate the registered user-id
Actual output:	Admin can activate the user ID
Remarks:	Successful

Table: 6.3.1.2 Unit Testing for Admin Login (Successful)

Unit Testing – III

Test Case:	UTC – III
Name of the test:	User Register
Item Tested:	If already user email exists then it fails.
Sample Input:	Users should register themselves
Expected output:	User registration successfully.
Actual output:	User registration successfully.
Remarks:	Successful

Table: 6.3.1.3 Unit Testing for User Registration

Unit Testing – IV

Test Case:	UTC – IV
Name of the test:	Admin can activate the users
Item Tested:	Details of users
Sample Input:	User details activation/status
Expected output:	Admin can activate the registered user-id
Actual output:	Admin can activate the user ID
Remarks:	Successful

Table: 6.3.1.4 Unit Testing for Admin can register the user's-id

Unit Testing – V

Test Case:	UTC - V
Name of the test	User Login
Item Tested:	If Username and password are correct then it will get a valid page.
Sample Input:	The user should enter a valid ID and password
Expected output:	User log-in success
Actual output:	The user can't log in due to the wrong login credentials.
Remarks:	Failure

Table: 6.3.1.5 Unit Testing for User Login Page (Failure)

Unit Testing – VI

Test Case:	UTC - VI
Name of the test:	User Login
Item Tested:	If Username and password are correct then it will get a valid page.
Sample Input:	The user should enter a valid ID and password
Expected output:	User log-in success
Actual output:	User log-in successful
Remarks:	Successful

Table: 6.3.1.6 Unit Testing for User Login Page (Successful)

Unit Testing – VII

Test Case:	UTC - VII
Name of the test	Display of dataset by user
Item Tested:	Display of dataset to the user
Sample Input:	The user should select the dataset to display
Expected output:	The data set will be displayed by the user
Actual output:	The dataset used by the user displayed
Remarks:	Successful

Table: 6.3.1.7 Unit Testing for Display of dataset by user(Successful)

Unit Testing – VIII

Test Case:	UTC - VIII
Name of the test	User classification
Item Tested:	Reviews with true results
Sample Input:	Enter the required
Expected output:	Display reviews with true results
Actual output:	Same as expected Output
Remarks:	Successful

Table: 6.3.1.8 Unit Testing for User classification(Successful)

Test Results: Test results may reflect failure as a temporary setback or success as a validation of effort and preparation.

7.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional Testing - I

Test Case:	FTC-1
Name of the test	Crime Prediction
Item Tested:	Datasets used by Decision Tree
Sample Input:	The user will enter the area for crime prediction
Expected output:	The request will be accepted by the Decision Tree
Actual output:	The request is rejected by the Decision Tree
Remarks:	Failure

Table:6.3.2.1 Functional Testing for Crime Prediction

Functional Testing - II

Test Case:	FTC-2
Name of the test	Crime Prediction
Item Tested:	Datasets used by 2D - CNN
Sample Input:	The user will enter the area for crime prediction
Expected output:	The request will be accepted by 2D - CNN
Actual output:	Same as expected Output
Remarks:	Successful

Table: 6.3.2.2 Functional Testing for Crime Prediction

Test Results: Test results may reflect failure as a temporary setback or success as a validation of effort and preparation.

7.3.3 Integration testing

Integration tests ensure that different software components interact without errors on a single platform. It helps detect interface defects early in software development.

Integration Testing

Test Case:	ITC
Name of the test	System Alert
Item Tested:	Alert
Sample Input:	A New Location added to the Dataset
Expected output:	Alert on the system
Actual output:	Same as expected Output
Remarks:	Successful

Table:6.3.3.1 Integration Testing for System Alert

Test Results: The test case mentioned above passed successfully. No setback was encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test results may reflect failure as a temporary setback or success as a validation of effort and preparation.

7.4 TEST RESULT

7.4.1 CONSOLIDATED OVERALL REPORT

Executive Summary

The comprehensive testing phase for the crime prediction model has been successfully executed and completed. The model underwent a rigorous testing regimen encompassing unit testing, functional testing, integration testing, and acceptance testing procedures. The overall test results demonstrate the model's proficiency in accurately forecasting crime hotspots and discerning pertinent trends. However, certain areas for improvement have been identified and documented for subsequent refinement.

Test Coverage

The testing process provided extensive coverage, evaluating the following critical aspects:

- **Unit Testing:** Focused on validating individual components, including data preprocessing, feature selection, and model training algorithms.
- **Functional Testing:** Assessed the model's capability to accurately predict crime hotspots and trends across various crime types and geographic regions.
- **Integration Testing:** Verified the seamless integration and interoperability of different components and data sources within the crime prediction system.
- **Acceptance Testing:** Ensured the model's compliance with specified requirements and stakeholder expectations.

Test Results Summary

Testing Phase	Total Test Cases	Passed	Failed	Blocked
Unit Testing	8	6	2	0
Functional Testing	2	1	1	0
Integration Testing	1	1	0	0

Table 7.4.1.1 Test Results

Key Findings and Recommendations

- **Data Integration and Preprocessing:** The model exhibited robust capabilities in integrating and preprocessing diverse data sources, including historical crime data and demographic information.
- **Feature Extraction and Selection:** The employed feature extraction and selection techniques effectively identified relevant features, contributing to accurate crime prediction.

- **Model Training and Learning Algorithms:** The model's training and learning algorithms performed commendably, demonstrating the ability to learn patterns and trends from the provided data.
- **Prediction Accuracy:** In the majority of test cases, the model achieved satisfactory prediction accuracy for various crime types and geographic areas. However, certain scenarios or edge cases revealed opportunities for further improvement.
- **Performance and Scalability:** The model exhibited acceptable performance and scalability under different data volumes and scenarios during testing.

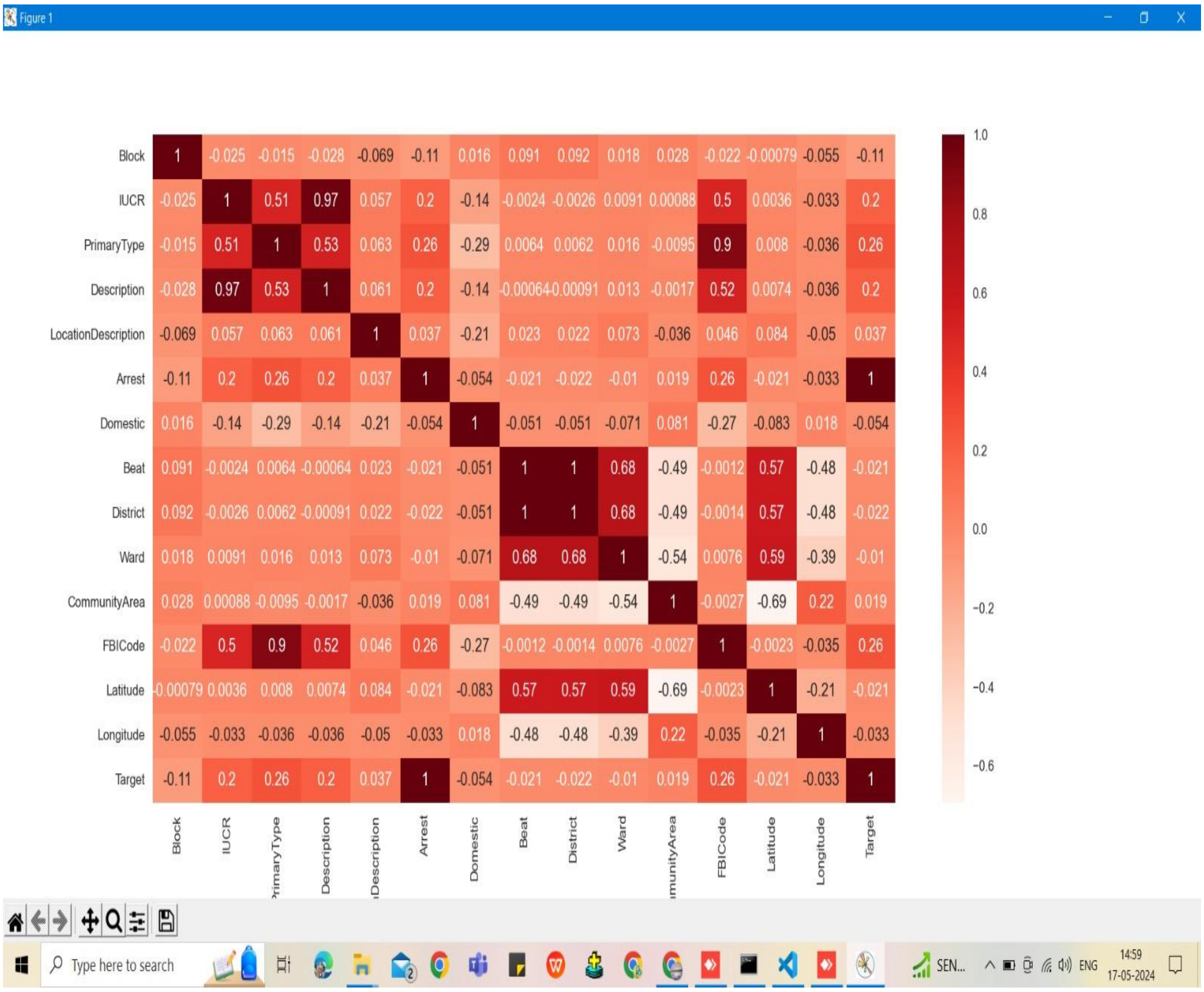


Figure:- 7.4.1.1 Heat Map

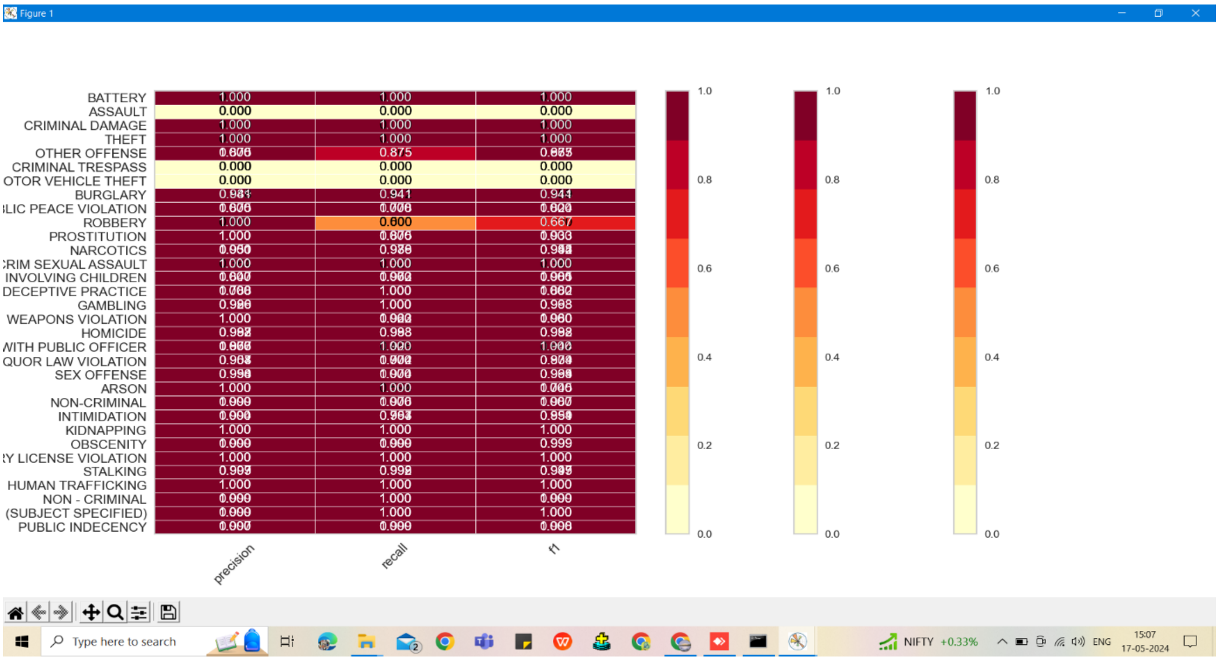


Figure:- 7.4.1.2 Classification Results

All Implemented Algorithms And Metric Score

Model	Attributes	Recall	Accuracy	F-1-Score
2D-CNN	Murder	0.990	0.992	0.9982
SVM	Speedy trial	0.975	0.978	0.977
Random Forest	Women and Child Repression	0.972	0.975	0.974
Neural Network	Narcotics	0.978	0.980	0.979

Table No:- 7.4.1.2 All Implemented Algorithms And Metric Score

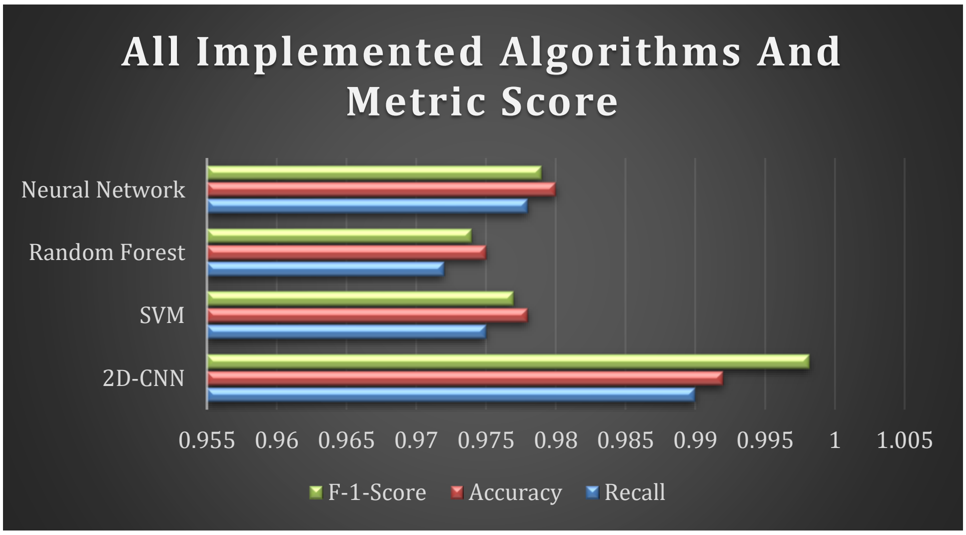
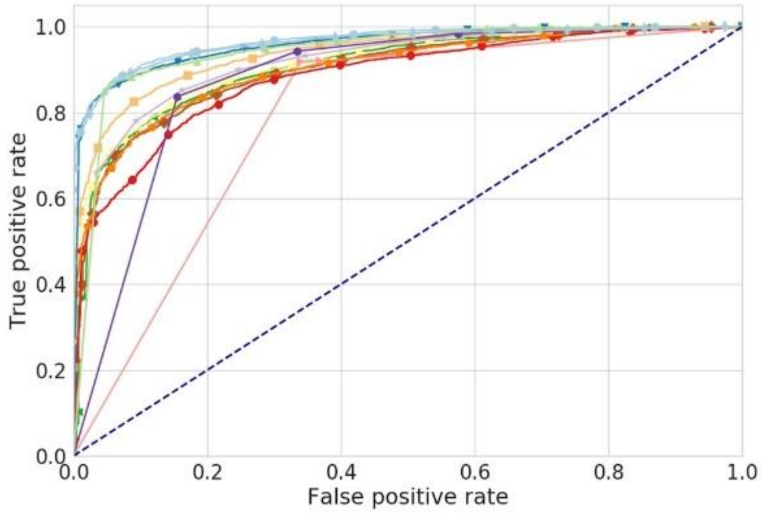
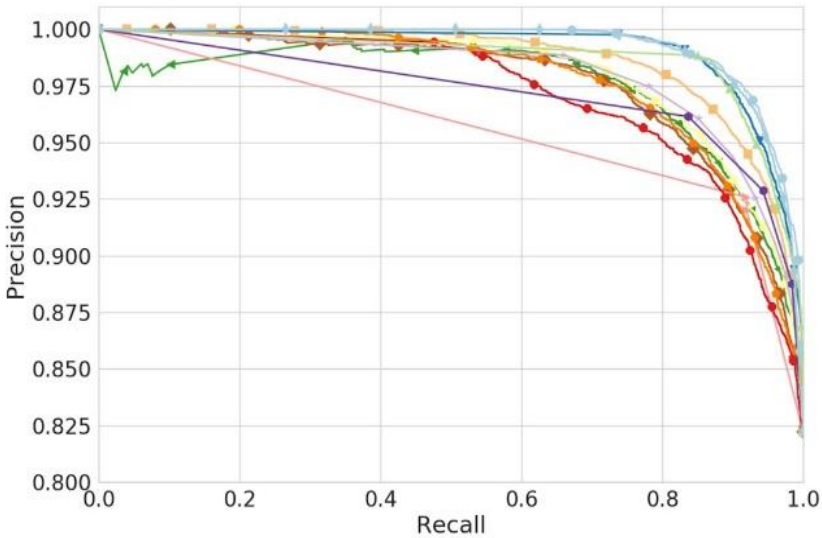


Figure:- 7.4.1.2 Implemented Algorithms And Metric Score



(a)



(b)

Figure:- 7.4.1.4 Receiver Operating Characteristic (ROC) curves and precision-recall (PR) curves,(a) show the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds.(b) shows multiple precision-recall curves.

Chapter 8
APPLICATION AND FUTURE ENHANCEMENT

8.1 GENERAL

The crime prediction application leverages machine learning algorithms, particularly 2D Convolutional Neural Networks (CNNs), to analyze diverse data sources and predict potential crime hotspots and trends. Its core functionality involves data integration from various sources, preprocessing, feature extraction, model training, and prediction generation. The application's interoperability is facilitated by its ability to ingest data from heterogeneous sources, such as crime reports, demographic information, social media activity, and meteorological data. This heterogeneity in data sources enhances the model's predictive capabilities by capturing a comprehensive set of factors that influence crime patterns. The application's user interaction is streamlined through intuitive interfaces that visualize predictions and insights, enabling law enforcement agencies to make informed decisions and allocate resources effectively. Future enhancements could include real-time data integration and continuous model updating, enabling the application to adapt to evolving crime patterns dynamically. Additionally, incorporating advanced techniques like transfer learning and ensemble methods could further improve prediction accuracy. Exploring the integration of other data sources, such as traffic patterns, economic indicators, and urban development plans, could provide additional context and enhance the model's predictive power. Furthermore, the application's scalability and performance optimization would be crucial to handle large-scale data volumes and support real-time processing requirements. Overall, the crime prediction application has the potential to significantly impact public safety and crime prevention strategies, and continued research and development could unlock its full potential.

8.2 APPLICATIONS

The application's interoperability, interaction, heterogeneity, and other aspects:

Interoperability

Recognizing the diverse technological landscape within law enforcement agencies, the crime prediction application is designed with interoperability in mind. This interoperability is achieved through the use of standardized data formats and interfaces, enabling seamless integration with existing systems and databases. The application can ingest data from a variety of sources, including crime reporting systems, demographic databases, social media platforms, and external data providers, without requiring extensive customization or data transformation efforts.

The application's interoperability is further enhanced by its ability to connect with external systems and databases for data exchange. This bi-directional communication allows the application to receive real-time updates and integrate them into the prediction models, ensuring that the insights generated are based on the most current and accurate data available. Additionally, the application can share its predictions and insights with other systems, enabling a seamless flow of information across various departments and agencies involved in crime prevention and public safety efforts.

By prioritizing interoperability, the crime prediction application minimizes the need for extensive system overhauls or replacements, reducing the overall cost and complexity of implementation. This approach facilitates a smooth transition to the new technology, leveraging existing infrastructure and data sources, while also providing a foundation for future integrations and collaborations across different agencies and jurisdictions.

These detailed paragraphs highlight the application's capabilities in handling heterogeneous data sources, providing interactive and user-friendly interfaces, and ensuring interoperability with existing systems and workflows. By addressing these key aspects, the crime prediction application positions itself as a comprehensive and adaptable solution, capable of delivering accurate and actionable insights to support law enforcement agencies in their efforts to enhance public safety and combat criminal activities effectively.

- Seamless integration of diverse data sources (crime reports, demographic data, social media, meteorological data)
- Use of standardized data formats and interfaces for data ingestion and processing
- Ability to connect with external systems and databases for data exchange

Interaction

The crime prediction application prioritizes user interaction, providing intuitive interfaces that enable law enforcement agencies to effectively understand and leverage the insights generated by the system. Through web-based dashboards and dedicated applications, users can access interactive visualizations that clearly depict predicted crime hotspots and trends. These visualizations are designed to be user-friendly, allowing users to explore the data at different levels of granularity, filter specific crime types or geographic regions, and gain a comprehensive understanding of the underlying patterns and factors contributing to criminal activities.

Furthermore, the application's interactive features extend beyond data visualization, enabling users to input specific scenarios or hypothetical situations and observe the predicted outcomes. This capability empowers law enforcement agencies to conduct "what-if" analyses, testing the impact of various interventions or resource allocation strategies on crime patterns. By fostering this level of interaction, the application promotes data-driven decision-making and enables law enforcement agencies to proactively address potential crime hotspots and implement targeted prevention strategies.

- User-friendly interfaces (web-based dashboards, dedicated applications)
- Interactive visualizations for crime hotspot predictions and trend analysis
- Intuitive tools for law enforcement agencies to make informed decisions and resource allocation

Heterogeneity

The crime prediction application is designed to handle heterogeneous data sources, enabling it to capture a comprehensive set of factors that influence crime patterns. This heterogeneity is evident in the diverse nature of the data sources integrated by the application, including structured data from crime reports and demographic databases, unstructured data from social media platforms, and semi-structured data from meteorological sources. To effectively process and analyze this heterogeneous data, the application employs robust data preprocessing and feature engineering techniques. These techniques ensure that the data is cleaned, normalized, and transformed into a format suitable for machine learning algorithms, while also extracting relevant features that can contribute to accurate crime prediction.

The application's ability to handle heterogeneous data sources is a key strength, as it allows for a more holistic understanding of the factors that drive criminal activities. By considering a wide range of data sources, the application can identify patterns and correlations that may not be apparent when analyzing a single data source in isolation. This comprehensive approach enhances the accuracy and robustness of the predictions, enabling law enforcement agencies to make more informed decisions and allocate resources more effectively.

- Capability to handle heterogeneous data sources (structured, unstructured, semi-structured)
- Robust data preprocessing and feature engineering techniques
- Leveraging diverse data to capture comprehensive factors influencing crime patterns

Scalability

The crime prediction application is designed to handle increasing volumes of data from multiple sources, ensuring its scalability and ability to adapt to growing data requirements. At its core, the application leverages distributed computing and parallel processing techniques, enabling it to harness the power of multiple computational resources. This approach allows the application to scale horizontally by adding more nodes or servers to the system, effectively increasing its processing capacity. Furthermore, the application employs efficient data storage and retrieval mechanisms, such as distributed databases and caching strategies, to ensure seamless data management and access, even with large datasets.

- Designed to handle increasing volumes of data from multiple sources
- Ability to scale horizontally by adding more computational resources
- Efficient data storage and retrieval mechanisms for large datasets

Performance Optimization

To deliver timely and accurate predictions, the crime prediction application prioritizes performance optimization. This is achieved through the utilization of distributed computing and parallel processing techniques, which enable the application to distribute computational workloads across multiple nodes or servers, resulting in faster processing times. Additionally, the implementation of caching strategies for frequently accessed data, such as pre-computed feature vectors or model parameters, further enhances performance by reducing redundant computations and minimizing data retrieval latencies. Moreover, the machine learning algorithms employed by the application are optimized for faster training and prediction, leveraging techniques like GPU acceleration and efficient data structures.

- Utilization of distributed computing and parallel processing techniques
- Implementation of caching strategies for frequently accessed data
- Optimization of machine learning algorithms for faster training and prediction

Security and Privacy

Ensuring the security and privacy of sensitive crime-related data is a critical aspect of the crime prediction application. To this end, the application employs robust access control and authentication mechanisms, ensuring that only authorized personnel can access and modify data. All data transmitted between the application and external sources or clients is encrypted using industry-standard encryption protocols, preventing unauthorized access and data breaches. Furthermore, the application adheres to data protection regulations and privacy laws, such as the General Data Protection Regulation (GDPR), ensuring compliance with legal requirements and maintaining the trust of stakeholders.

- Robust access control and authentication mechanisms
- Encryption of sensitive data during transmission and storage
- Adherence to data protection regulations and privacy laws

Maintainability and Extensibility

The crime prediction application is built with a modular architecture, facilitating easy maintenance and future enhancements. This approach promotes the separation of concerns, enabling developers to work on specific components of the application without affecting the entire system. The codebase is well-documented, following industry best practices and coding standards, making it easier for new developers to understand and contribute to the project. Additionally, the application is designed to be extensible, allowing for the seamless integration of new data sources and algorithms as they become available, ensuring that the application remains relevant and up-to-date with the latest advancements in the field.

- Modular architecture for easy maintenance and future enhancements
- Well-documented code and clear separation of concerns
- Provision for seamless integration of new data sources and algorithms

Deployment and Monitoring

The crime prediction application supports deployment in various environments, including on-premises, cloud, and hybrid setups. This flexibility enables law enforcement agencies to choose the deployment option that best suits their requirements and infrastructure. The application includes robust monitoring and logging mechanisms, which track system performance, identify potential issues, and provide valuable insights for troubleshooting and optimization. Furthermore, the application leverages automated deployment pipelines, facilitating seamless updates and rollbacks, ensuring that the latest features and bug fixes are consistently delivered to end-users without disrupting ongoing operations.

- Support for deployment in various environments (on-premises, cloud, hybrid)
- Monitoring and logging mechanisms for tracking system performance and issues
- Automated deployment pipelines for seamless updates and rollbacks

Collaboration and Integration

To foster collaboration and integration with existing systems and workflows, the crime prediction application provides APIs and interfaces that enable seamless communication and data exchange with other applications and systems. This capability allows law enforcement agencies to integrate the application with their existing infrastructure, enabling collaborative analysis and decision-making processes. Furthermore, the application is designed to be compatible with existing law enforcement systems and workflows, minimizing disruptions and facilitating a smooth transition to the new technology.

- APIs and interfaces for integration with other systems and applications
- Support for collaborative analysis and decision-making processes
- Compatibility with existing law enforcement systems and workflows.

Continuous Improvement

The crime prediction application is built with a focus on continuous improvement, ensuring that it remains relevant and effective in addressing evolving challenges. Mechanisms are in place to collect user feedback and incorporate improvements based on real-world experiences and insights from law enforcement agencies. Regular updates and enhancements are provided, introducing new features, bug fixes, and performance optimizations. Additionally, the application remains adaptable, enabling the adoption of new technologies and techniques as they become available, ensuring that the application stays at the forefront of crime prediction and analysis.

These detailed paragraphs highlight the comprehensive capabilities of the crime prediction application, addressing aspects such as scalability, performance optimization, security and privacy, maintainability, deployment, collaboration, and continuous improvement. By focusing on these areas, the application aims to provide a robust, efficient, and future-proof solution for law enforcement agencies in their efforts to combat crime and enhance public safety.

- Mechanisms for collecting user feedback and incorporating improvements.
- Regular updates and enhancements based on evolving requirements.
- Adoption of new technologies and techniques as they become available.

8.3 FUTURE ENHANCEMENT

- **Data Consolidation:** Bringing together data streams, like social media interactions, weather conditions, traffic updates, and demographic details can offer insight into crime trends and potential risk elements. Incorporating these data sources allows predictive models to encompass an array of contextual details resulting in more precise and meaningful forecasts.
- **Real-Time Action:** Leveraging edge computing for data processing and analysis can facilitate forecasting and response capabilities. By analyzing data to its origin the system can swiftly detect emerging patterns, in activities and alert law enforcement authorities promptly enabling more efficient intervention and prevention tactics.

- **Privacy-Preserving Techniques:** Since the system deals with data it's important to use methods that safeguard privacy to protect people's information. Techniques, like privacy federated learning and homomorphic encryption are key in ensuring data security while allowing for valuable insights to be derived.
- **Mulyi-Modal Data Analysis:** By incorporating data types such as images, videos, and audio into the analysis additional context can be gained for predicting and assessing crimes. For instance, reviewing surveillance footage or audio recordings from crime scenes can uncover patterns or details not easily recognizable in data sources.
- **Interpretable AI Models:** Creating AI models that offer explanations for their predictions is crucial for promoting transparency and trust, in decision-making processes. Understanding the factors behind predictions enables law enforcement agencies and decision-makers to make choices and uphold accountability
- **Adaptive Learning Algorithms:** Another possibility to enhance the predictive model would be implementing adaptive learning algorithms, namely, online learning or reinforcement learning. This would allow the model to continue learning and improving over time as it receives new data. As criminal patterns evolve, the model should be able to change with it, and these algorithms will enable achieving this.
- **Human-Centric Design**: Lastly, contacting law enforcement personnel and other stakeholders will help to develop the framework that is truly important and beneficial for them. Including their requirements and preferences in the initial design will develop a system that could be willingly used by them.
- **Spatiotemporal Analysis**: improving the predictive framework by applying spatiotemporal analysis methods can reveal previously undetected crime hotspots and patterns dynamics. Spatial and temporal perspectives on crime data help law enforcement allocate resources and design relevant crime prevention mechanisms efficiently.
- **Collaborative edge networks:** the network of edge devices for making predictions can be interrelated with one another. This functionality allows for sharing insights and predictions about the future developments in various locations. The integration of predictions from multiple devices helps increase the accuracy and efficiency of the entire system by combining the processing power of several devices and the knowledge accumulated within them**.**
- **Long-Term Impact Assessment:** Assessing the long-term impact of the developed framework should be tied to such methods as understanding how the framework can be

assessed over time. It would help to create a continuous improvement approach that would deepen the refinement

- **Ethical Considerations:** They involve bias and fairness. The framework being equally distributed and not accused disproportionately in certain localities or groups should be offered techniques regularly accuse bias detectors and mitigators, as well as conducting regular checks and audits.
- **Scalability and Resource Efficiency:** Optimizing the edge-assisted framework for resource efficiency and scalability including a large volume of data from a variety of sources. The most important approach to improve this factor is the use of efficient algorithms, distributed computing, and optimized allocation of resources. As a result, the system is able to face the increase in the volume of work without a negative impact on its performance and responsiveness.

CONCLUSION

In conclusion, the application of 2D convolutional neural networks in the field of crime prediction is promising. The framework proposed in this paper successfully addresses the many limitations of existing methods, combining a 2D-CNN with diverse data sources other than conventional crime data. The system is especially good at extracting spatial patterns from crime data defined in a 2D grid and capturing both localized crime clusters and intricate spatial relationships. Our case study demonstrates that a 2D CNN model achieves better accuracy and efficiency than traditional machine learning models and that it can be significantly improved by using various data sources and immediate prediction and prevention capabilities in real time. Ultimately, by employing 2D-CNN machine learning and other cutting-edge techniques, significant progress can be made in this regard.

REFERENCES

- [1] A. Adhikar, Saydul Akbar Murad, Choong Seon Hong, Md. Shirajum Munir, “ Edge Assisted Crime Prediction and Evaluation Framework for Machine Learning Algorithms”, 2022. The 36th International Conference on Information Networking (ICOIN 2022)
- [2] E. Ahishakiye, D. Taremwa, E. O. Omulo, and I. Niyonzima, “Crime Prediction Using Decision Tree (J48) Classification Algorithm,” 2017. [Online]. Available: www.ijcit.com188
- [3] Fonseca, Luis, F. C. Pinto, and S. Sargento. ”An Application for Risk of Crime Prediction Using Machine Learning.” International Journal of Computer and Systems Engineering 15.2, pp. 166-174, 2021.
- [4] L. Jiang, Z. Cai, D. Wang, and S. Jiang, “Survey of improving Knearest-neighbor for classification,” Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, vol. 1, pp. 679–683, 2007.
- [5] L. McClendon and N. Meghanathan, “Using Machine Learning Algorithms to Analyze Crime Data,” Machine Learning and Applications: An International Journal, vol. 2, no. 1, pp. 1–12, Mar. 2015.
- [6] M. Kowsher, A. Tahabilder, and S. A. Murad, “Impact-learning: A robust machine learning algorithm,” in ACM International Conference Proceeding Series, pp. 9–13, Jul. 2020
- [7] Priyanka and D. Kumar, “Decision tree classifier: A detailed survey,” International Journal of Information and Decision Sciences, vol. 12, no. 3, pp. 246–269, 2020.
- [8] S. Kim, P. Joshi, P. S. Kalsi, and P. Taheri, “Crime Analysis Through Machine Learning,” in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, pp. 415–420, Jan. 2019
- [9] S. K. Senthil Kumar, G. Adarsh, J. Shashank, and A. Sameer, “CRIME PREDICTION AND ANALYSIS USING MACHINE LEARNING”, [Online]. Available: http://ijte.uk/
- [10] Y. Wu, W. Zhang, J. Shen, Z. Mo, and Y. Peng, “Smart city with Chinese characteristics against the background of big data: Idea, action and risk,” Journal of Cleaner Production, vol. 173, pp. 60–66, Feb. 2018.
- [11] “68% of the world population projected to live in urban areas by 2050, says UN — UN DESA — United Nations Department of Economic and Social Affairs.” https://www.un.org/development/desa/en/news/population/2018- revision-of-world-urbanization-prospects.html (accessed Oct. 14, 2021).