Counselling Code : VBIT

VIGNANA BHARATHI®

Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Cyber Crime Prediction by Machine Learning Model XGB Classifier using Python

PROJECT GUIDE:

Mrs. A.Sandhya

Assistant Professor,

CSE

BATCH NUMBER: 20

Year/sem/section: CSE/IV-I/A

TEAM MEMBERS:

B. Shekhar	21P61A0527
B. Madhuri	21P61A0528
B. Uday	21P61A0539

ABSTRACT

This presentation explores the significance of crime prediction through data analytics and machine learning in enhancing public safety and optimizing law enforcement operations. It discusses the challenges associated with predictive policing, such as data bias, privacy concerns, and the complexity of criminal behavior. Various research methodologies, including machine learning, geospatial analysis, and social network analysis, are examined to provide insights into predicting future crime. The presentation concludes with future research directions, emphasizing ethical considerations and equitable practices in the deployment of predictive analytics for crime prevention.

Keywords: Crime prediction, Data analytics, Public safety, Xgbooster classifier, predictive policing.

INTRODUCTION

- **Importance of Predicting Future Crime:** Enhances proactive law enforcement, optimizes resource allocation, and ensures fair policing practices through data-driven insights.
- **Challenges in Predicting Future Crime:** Issues include data bias and quality, privacy concerns, and the complexity of criminal behavior.
- **Research Approaches and Methodologies:** Utilizes machine learning algorithms, geospatial analysis, social network analysis, and predictive analytics platforms to address prediction challenges.
- **Applications and Implications:** Improves community policing, enables targeted interventions, and necessitates ethical considerations to ensure fairness and accountability in predictive policing.

LITERATURE SURVEY

	LITERATURE SURVEY	PROS	CONS
[1]	XGBoost for Cyber Crime Classification	• High accuracy • Fast execution • Handles imbalanced data effectively • Regularization reduces overfitting	• Complex to tune • Requires large datasets • Overfitting risk if hyperparameters are not carefully controlled
[2]	Random Forest for Cyber Crime Prediction	• Robust to overfitting • Easy to interpret • Effective with large datasets	• Computationally expensive • May struggle with extreme data imbalance
[3]	SVM (Support Vector Machine) for Cyber Crime Data	• Good for high dimensional data • Works well with smaller datasets	• Not suitable for large datasets • Slower in training compared to tree-based models
[4]	Deep Learning for Complex Cyber Crime Patterns	• Excellent for unstructured data (e.g., images, texts) • Learns complex patterns and interactions	• Requires extensive computational resources • Prone to overfitting without proper regularization

RESEARCH GAP / CHALLENGES

- **Data Quality and Bias:** Address issues with incomplete, biased, or underreported crime data that impact model accuracy and fairness.
- **Privacy and Ethical Concerns:** Balance the use of personal data for predictions with the need to protect privacy and prevent algorithmic bias.
- **Complexity of Crime Patterns:** Model the multifaceted and evolving nature of criminal behavior effectively.
- **Model Interpretability and Real-Time Application:** Enhance the interpretability of models and ensure they can process and analyze data in real-time for actionable insights.

PROBLEM STATEMENT

“Predicting Future Crime is new requirement for current era. Many of scholars has work on this domain and proposed various models but need to improve the accuracy of the work. Hence accuracy of prediction is the basic issue in this type of work. Further many of scholar has not improve the input dataset like pre-processing steps, as this increase the learning of the model. To resolve all above issue this model introduces the new learning method with improved results.”

OBJECTIVES

- To perform a thorough analysis of the research area.
- To study the problems in the System through fact-finding techniques.
- To develop conceptual, logical, and physical models for the system.
- To enhance the learning model detection accuracy.
- To propose a Predicting Future Crime model with the trained dataset.
- To document our efforts and analysis in a proper comprehensible manner.

EXISTING SYSTEM

Existing crime prediction systems primarily use traditional statistical methods and basic machine learning models. They often rely on historical data to identify patterns but lack advanced analytics and real-time adaptability. These systems face issues with data quality, bias, and scalability, which limit accuracy and effectiveness in resource allocation, and struggle with integrating diverse data sources for timely insights.

PROPOSED SYSTEM

The proposed system will address the limitations of existing crime prediction models by using the XGBoost Classifier for improved accuracy. It will integrate diverse data sources and apply advanced algorithms to handle large, complex datasets. The system will feature robust hardware and software, ensuring scalability, reliability, and real-time processing. Its design aims to provide actionable insights, enhance resource allocation, and support proactive crime prevention.

PROPOSED METHODOLOGY

- Requirement Analysis
- Data Collection and Preparation
- Model Development
- System Integration
- Testing and Validation
- Deployment and Maintenance
- Documentation and Reporting

Modules

1.**Data Collection and Preprocessing Module**

- Collect crime-related data from multiple sources (databases, APIs).
- Handle missing or incomplete data using techniques like imputation.
- Normalize and clean the dataset for consistent input into the model.
- Tools: Pandas, NumPy.

2. **Feature Engineering Module**

- Identify and create features relevant to crime prediction.
- Encode categorical data and scale numerical features.
- Perform feature selection to reduce dimensionality and improve model performance.
- Tools: Scikit-learn, Featuretools.

3. **Model Development Module**

- Implement the XGBoost classifier for predicting crime likelihood.
- Optimize hyperparameters using grid search or randomized search.
- Train, validate, and test the model.
- Tools: XGBoost, Scikit-learn.

4.Geospatial Analysis Module

- Incorporate location-based data (latitude, longitude) for crime pattern analysis.
- Generate heatmaps for high-crime areas.
- Tools: Geopandas, Folium, Matplotlib.

5.Social Network Analysis Module (Optional)

- Analyze networks of criminal activities or social connections.
- Identify influential nodes or patterns within the network.
- Tools: NetworkX.

6.Prediction and Visualization Module

- Provide real-time or batch predictions for new data inputs.
- Visualize predictions using charts, graphs, or dashboards.
- Tools: Matplotlib, Seaborn, Plotly, Dash.

Implementation

# How to Run the Intrusion Detection System App

## Overview

```
python app.py --port 5001
```

## Prerequisites

- Python 3.10
- Git
- Virtual Environment (Optional)
- Conda (Optional)

## Setup Instructions

### 1. Create a Virtual Environment (Optional)

#### Using Virtual Environment

```
\```bash
python -m venv intrusion_detection_env
```

# Activate the virtual environment

# On Windows

```
.\intrusion_detection_env\Scripts\activate
```

# On macOS and Linux

source intrusion_detection_env/bin/activate

\```

#### Using Conda

\```bash

conda create --name intrusion_detection_env python=3.10

conda activate intrusion_detection_env

\```

### 3. Install Dependencies

\```bash

pip install -r requirements.txt

## Running the App

### Start the Application

\```bash

python app.py

\```

## Troubleshooting

- Ensure that all dependencies are correctly installed.
- If encountering any issues, refer to the console or
- terminal output for error messages.

## Conclusion

Follow these instructions to set up and run the Intrusion Detection System application. For any further assistance or queries, please reach out for support.

```
python app.py --port 5001
```

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- Intel Processor 2.0 GHz or above.
- 2 GB RAM or more.
- 160 GB or more Hard Disk Drive or above

SOFTWARE REQUIREMENTS:

- Microsoft Windows 7/8.
- HTML/Python.
- Ms-Office package.

REFERNECS

- **FOR MY SQL**

- http://www.mysql.com/

- **FOR CSS**

- http://cssed.sourceforge.net/

- **FOR OTHER USEFUL REFERENCES**

- http://www.w3schools.com/default.asp
- http://en.wikipedia.org/

- **cask anaconda ->** //www.anaconda.com/

- git.->https://git-scm.com/