



SAFE CITY VISION

Sanjana¹, Shreya², Khushi³, Ms.Aarti Chahal⁴

Department of CSE (AI &DS),
Panipat Institute of Engineering and Technology, Samalkha, Panipat, India^{1,2,3,4}
Mail ID: aarti.cse.ai-ds@piet.co.in

Abstract—This paper presents Safe City Vision, a machine-learning-based crime detection and analysis project intended to examine historical crime patterns, support prediction, and provide decision-oriented visual analytics. The project uses crime records containing geographic, temporal, and categorical information and applies preprocessing, feature extraction, classification, visualization, and interactive analysis. The source report describes K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, time-series forecasting, and real-time analysis as components or proposed extensions. The documented project evaluation reports an overall accuracy of 82%, while the conclusion reports 85%; these values are retained as reported results rather than reconciled by assumption. The study also considers privacy, secure storage, fairness, and the responsible use of predictive crime analytics.

Keywords: crime detection, crime analysis, machine learning, KNN, SVM, predictive policing, data visualization, safe city.

INTRODUCTION

The project titled "Safe City Vision" focuses on examining historical crime data to uncover patterns and trends across regions, with the goal of enhancing public safety and supporting law-enforcement agencies. The report describes the use of data analytics, visualization, and machine learning to understand the frequency and distribution of crimes and to identify high-risk areas and time periods.

Modern crime analysis has evolved from manual records and eyewitness accounts toward digital data, artificial intelligence, machine learning, and large-scale analytics. These technologies can process large datasets and reveal patterns that are difficult to identify manually. However, privacy, data quality, algorithmic bias, interoperability, and ethical deployment remain important challenges.

The project aims to provide a dynamic analytical framework that can support crime monitoring, classification, trend analysis, and resource planning. The report also discusses real-time surveillance, social-media monitoring, deep learning, and blockchain-based secure storage as complementary or future capabilities.

RELATED WORK

Crime forecasting has been investigated using statistical, spatial, temporal, and machine-learning approaches. Mohler et al. developed a self-exciting point-process model for crime forecasting [1]. Wang et al. studied spatial-temporal crime patterns and reported locality and periodicity in crime events [2]. Perry et al. examined predictive policing as a decision-support approach for crime forecasting and operational planning [3].

Machine-learning methods including KNN, SVM, decision trees, Random Forest, and clustering have been used for crime classification and prediction. Akshaya and Suruliandi investigated K-means clustering and SVM for crime prediction [4]. Hotspot mapping has also been studied as a means of



identifying spatial concentrations of crime [5]. Kaur and Sharma explored LSTM networks for crime-rate forecasting [6]. These studies motivate a combined analytical workflow rather than reliance on a single model.

PROBLEM STATEMENT AND OBJECTIVES

Conventional crime reporting can be delayed, fragmented, and dependent on subjective or incomplete information. Periodic analysis may fail to capture rapidly changing crime patterns, while heterogeneous datasets make direct comparison difficult. A dynamic analytical system is therefore required to clean and organize crime data, identify relevant features, visualize distributions, classify crime patterns, and provide predictive insights.

The project objectives stated in the source report include designing a crime detection and analysis system, applying Python-based machine-learning techniques, evaluating classification performance, and supporting real-time monitoring. The report identifies an intended target accuracy of 80% and emphasizes practical use in crime monitoring and prevention.

SYSTEM METHODOLOGY

The methodology begins with data collection followed by preprocessing, feature extraction, model development, evaluation, and presentation. The report describes publicly available crime records containing categorical and numerical variables such as crime type, location, time, and severity. Preprocessing includes data cleaning, scaling or normalization, categorical encoding, and, where required, augmentation or oversampling. Fig. 1 presents the entity relationship diagram summarizing the principal data entities used by the system.



Fig. 1. Entity relationship diagram.

As shown in Fig. 1, the entity relationship structure represents major crime-analysis entities such as state, district, year, crime type, and occurrence information. This provides a conceptual representation of the data relationships used by the analytical system.



Fig. 2. Data flow diagram.

Fig. 2 summarizes the movement from input crime data through preprocessing and analysis toward model output and decision support. The source report describes real-time processing as an intended extension in which incoming surveillance data are captured, preprocessed, classified, and presented as alerts.

A. Data Preprocessing

The preprocessing stage removes or handles missing values, standardizes numerical variables, encodes categorical fields, and attempts to reduce class imbalance. For KNN and SVM, scaling is particularly important because distance and feature magnitudes influence classification. The source report also describes data augmentation and oversampling for improving representation of underrepresented categories.

B. Dataset and Model Architecture

Fig. 3 shows an overview of the crime dataset used in the project. The project report describes a crime dataset containing labeled crime information with geographic, temporal, and categorical attributes. Different sections of the report state different dataset sizes; therefore, the exact count should be verified from the original dataset before publication. The paper retains the dataset characteristics that are consistently described: structured crime records, multiple crime categories, geographic information, and temporal information.

Fig. 3. Dataset overview.

TABLE I. SUMMARY OF SYSTEM COMPONENTS

Table I summarizes the primary components of the analytical pipeline, from input data through the software environment used for implementation.

Component	Configuration / Description
Input data	Crime type, location, time, severity and related attributes
Preprocessing	Cleaning, scaling, encoding, augmentation/oversampling
Primary	KNN and SVM



Component	Configuration / Description
models	
Additional analysis	Random Forest, regression, clustering
Evaluation	Accuracy; confusion matrix; precision/recall/F1 discussed
Environment	Python, scikit-learn, NumPy, Pandas, Matplotlib

C. Model Architecture

KNN classifies an observation according to the labels of nearby observations, while SVM learns a separating hyperplane between classes. The report states that important features include location patterns, time of occurrence, incident frequency, and crime-related characteristics. Distance-weighted KNN and SVM hyperparameter tuning are identified as possible improvements.

The report also describes a Random Forest classifier in its implementation appendix for tabular crime analysis, together with Random Forest regression for estimating total-crime values and K-means clustering for grouping similar crime patterns, as illustrated by the dataset excerpt in Fig. 4. These components provide complementary supervised and unsupervised views of the data.

#	Column	Non-Null Count	Dtype
0	STATE/UT	9842 non-null	object
1	DISTRICT	9842 non-null	object
2	YEAR	9842 non-null	int64
3	MURDER	9842 non-null	int64
4	ATTEMPT TO MURDER	9842 non-null	int64
5	CULPABLE HOMICIDE NOT AMOUNTING TO MURDER	9842 non-null	int64
6	RAPE	9842 non-null	int64
7	CUSTODIAL RAPE	9842 non-null	int64
8	OTHER RAPE	9842 non-null	int64
9	KIDNAPPING & ABDUCTION	9842 non-null	int64
10	KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS	9842 non-null	int64
11	KIDNAPPING AND ABDUCTION OF OTHERS	9842 non-null	int64
12	DACOITY	9842 non-null	int64
13	PREPARATION AND ASSEMBLY FOR DACOITY	9842 non-null	int64
14	ROBBERY	9842 non-null	int64
15	BURGLARY	9842 non-null	int64
16	THEFT	9842 non-null	int64
17	AUTO THEFT	9842 non-null	int64
18	OTHER THEFT	9842 non-null	int64
19	KIDTS	9842 non-null	int64
20	OTHER DR CRIMES	9842 non-null	int64
21	TOTAL DR CRIMES	9842 non-null	int64

Fig. 4. Crime dataset fields used in the project.

D. Training and Evaluation

The report specifies training configurations including a batch size, iterative training, early stopping, and accuracy-based evaluation. For the KNN/SVM workflow, the model is evaluated on unseen data and a confusion matrix is used to examine category-level errors. The project identifies misclassification between visually or statistically similar crime categories as an important limitation.

For the Random Forest implementation documented in the appendix, the data are divided into training and testing subsets and the classifier is fitted using scikit-learn. The implementation also calculates classification accuracy and a classification report. The regression module reports MAE and R^2 , while K-means uses an elbow curve to select a suitable number of clusters.



IMPLEMENTATION AND RESULTS

Fig. 5 presents a sample of the dataset records used during implementation. The software environment is based on Python and common data-science libraries. Pandas is used for data manipulation, NumPy for numerical processing, Matplotlib and Seaborn for visualization, and scikit-learn for machine-learning algorithms. Jupyter Notebook and Google Colab are described in the report as development environments.

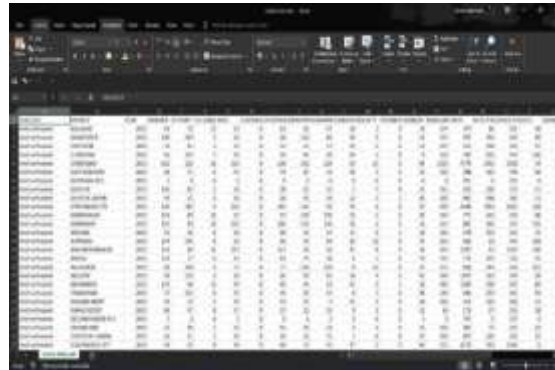


Fig. 5. Sample dataset used for crime analysis.

The project includes interactive analysis in which users can select a state, crime type, and year range. A linear-regression component can estimate future values for a selected crime series. A separate graph-based prototype represents city-to-city crime-risk scores as weighted edges and applies Dijkstra's algorithm to obtain a minimum accumulated-risk route.

TABLE II. REPORTED PERFORMANCE VALUES

Table II lists the accuracy figures reported across different sections of the source document; these values are not reconciled into a single number.

Metric / Finding	Reported Value in Source
Target accuracy	80%
Evaluation finding	82% overall accuracy
Conclusion finding	85% accuracy
Validation accuracy	75–80%
Test accuracy	75% (reported)

Because the source report contains multiple, internally inconsistent accuracy statements, this paper does not combine them into a single unsupported value, as summarized in Table II. The results should be understood as project-reported figures requiring verification against the final experimental logs or dataset.

A. Error Analysis

The report indicates stronger performance for categories with more distinctive patterns and lower performance for categories with overlapping characteristics. Assault and vandalism are described as more difficult categories, while robbery and fraud are reported to have stronger results. These differences suggest that feature quality, class balance, and category separability substantially influence performance.



B. Visual Analysis

Correlation analysis and heatmaps, such as the one shown in Fig. 6, are used to understand relationships among numerical variables. Visualization supports exploratory identification of highly related crime categories and can help analysts interpret model inputs before training or deployment.

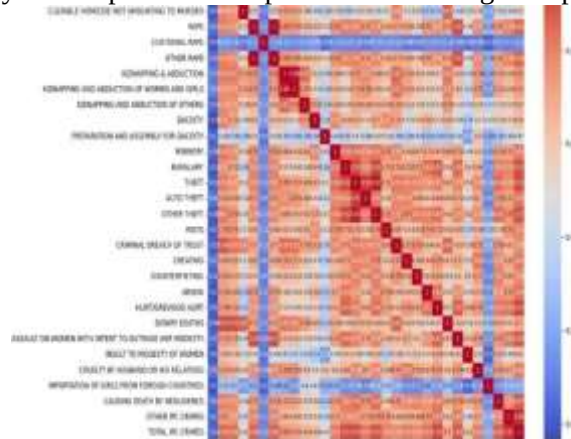


Fig. 6. Heat map of crime-related variables.

The heat map in Fig. 6 provides a compact representation of pairwise relationships among variables. Such visual analysis can guide feature selection and identify redundant or strongly correlated inputs before classification or regression.

C. Hardware and Software Requirements

The project report also specifies the hardware and software environment used for development and testing. The recommended hardware includes an Intel Core i5-class processor with a minimum clock speed of 2.4 GHz, 8 GB of RAM as a baseline with 16 GB recommended for faster training, and a dedicated GPU such as an NVIDIA GeForce GTX 1650 to accelerate computation for larger models. An SSD with at least 50 GB of free storage is recommended for dataset and model storage, and a webcam is listed as a supporting device for testing the real-time functionality described below.

On the software side, the report describes a Windows or Linux (Ubuntu-preferred) environment running Python 3.8 or later, with Jupyter Notebook and Google Colab used as the primary development environments because of their interactive features and access to cloud-based GPUs. Core libraries include TensorFlow and Keras for model development, OpenCV for image and video handling, and NumPy, Pandas, and Matplotlib for data manipulation and visualization. This combination of accessible hardware and open-source software keeps the overall system reproducible and inexpensive to deploy at a small scale.

D. Real-Time Pipeline and Clustering Analysis

The report documents a frame-based real-time processing pipeline in which video frames are captured from a surveillance camera, preprocessed through resizing, grayscale conversion, and normalization, passed through the trained KNN or SVM model for classification, and the resulting label is displayed on a monitoring interface. This pipeline is intended to give continuous feedback rather than relying on periodic manual review.

Beyond classification, the appendix code also includes a K-means clustering routine that groups states by total crime volume and by their most frequent crime categories, using the elbow method to select an



appropriate number of clusters. This clustering view complements the classification and regression components by grouping regions with similar crime profiles, which can support resource-allocation decisions independently of any single crime-type prediction. A separate interactive dashboard, built with dropdown and slider widgets, allows a user to filter the dataset by state, crime type, and year range and to plot the resulting trend directly from the underlying records.

E. Route-Risk and Dashboard Prototypes

The appendix code implements a small graph-based prototype in which cities are represented as nodes and pairwise crime-risk scores are represented as weighted edges. Given a source and destination city selected from dropdown widgets, the prototype applies Dijkstra's shortest-path algorithm to the weighted graph to identify the route with the lowest accumulated risk score rather than the shortest physical distance, and renders the resulting network with the selected path highlighted. While the underlying risk values in the appendix are illustrative rather than derived from the full dataset, the prototype demonstrates how a graph-based safety metric could be layered on top of the crime-classification and forecasting components described earlier in this paper.

A second prototype provides an interactive filtering dashboard built with dropdown menus for state and crime type and a slider for year range. Selecting a combination of filters returns the matching subset of records and plots the corresponding trend line, giving an analyst a quick way to inspect a specific state-crime-year combination without writing new queries. Together with the linear-regression forecasting component described in Section V, these widgets illustrate a lightweight analyst-facing interface that sits on top of the underlying classification and regression models.

F. Reproducibility Notes

Because the appendix code loads the dataset directly from a shared Google Drive link and mixes several modeling approaches -- a Random Forest classifier applied to a regression-style target, a separate Random Forest regressor over synthetic date-derived features, K-means clustering, and the two widget-based prototypes described above -- reproducing the reported figures requires first separating these components into a single, consistently defined train/test pipeline. Fixing a random seed, documenting the exact feature set used for each model, and logging metrics for each component independently would allow the accuracy figures summarized in Table II to be verified and compared on a like-for-like basis.

DISCUSSION

The project demonstrates that machine learning can provide a practical framework for organizing and analyzing crime records. Classification can support categorization of crime patterns, regression can estimate aggregate crime levels, and clustering can identify groups of locations or crime profiles with similar characteristics. Interactive visualization can make these outputs more accessible to analysts and decision makers.

The main practical strengths are the modular architecture, use of established machine-learning algorithms, visualization support, and the ability to combine historical analysis with interactive exploration. The project also recognizes that predictive outputs are estimates derived from historical records rather than direct evidence of future criminal activity.

A. Limitations

Several limitations are identified in the source report. Dataset coverage may not represent all crime types or demographic and geographic variations. Imbalanced categories can reduce minority-class performance. Real-time implementation can be constrained by computational cost, data quality, and system scalability.



In addition, the report contains different descriptions of datasets, models, and performance values; this inconsistency should be resolved before the system is presented as a reproducible benchmark.

B. Ethical and Security Considerations

Crime analytics requires strong safeguards because model outputs can affect individuals and communities. The source report emphasizes privacy, transparency, accountability, anonymization, and secure storage. It also proposes blockchain-based record protection to improve data integrity and resistance to tampering [7].

Predictive systems should not be treated as autonomous authorities for intervention. Historical crime data may contain reporting bias or unequal representation, and a model can reproduce such patterns. Human oversight, auditability, access controls, documented limitations, and periodic evaluation are therefore necessary.

C. Practical Deployment

A responsible deployment architecture should separate data ingestion, preprocessing, model inference, visualization, and human review. Real-time camera, social-media, weather, or external data sources should only be incorporated after validation of their accuracy, legal basis, privacy implications, and operational necessity. Model performance should be monitored after deployment because crime patterns and reporting practices can change over time.

D. Comparative Summary

Table III contrasts conventional, manually driven crime reporting with the Safe City Vision analytical framework across several practical dimensions. The comparison highlights why a machine-learning-based pipeline is expected to offer advantages in consistency and responsiveness, while also showing that these advantages depend on the same data-quality and governance safeguards discussed in the limitations and ethical-considerations subsections.

TABLE III. TRADITIONAL VERSUS SAFE CITY VISION APPROACH

Dimension	Traditional Approach	Safe City Vision
Data handling	Manual, fragmented records	Centralized structured pipeline
Update frequency	Periodic, delayed	Supports frequent re-analysis
Pattern detection	Analyst-dependent	ML-assisted classification
Visualization	Static reports	Interactive dashboards
Predictive insight	Limited	KNN/SVM/RF-based prediction
Oversight need	Institutional review	Human-in-the-loop plus audits



CONCLUSION

Safe City Vision provides a multidisciplinary approach to crime detection and analysis using machine learning, visualization, and interactive decision support. The project demonstrates the feasibility of applying KNN, SVM, Random Forest, regression, and clustering techniques to crime-related data. The reported results indicate useful predictive performance, although the inconsistent accuracy figures in the source document require final experimental verification.

The work provides a foundation for future research involving larger validated datasets, improved temporal modeling, spatial hotspot analysis, calibrated predictions, fairness assessment, and real-time data integration. Such development should prioritize transparency, privacy, human oversight, and reproducibility.

FUTURE SCOPE

Future work can extend the project in several directions. First, the dataset and train/test protocol should be formally documented and independently reproducible. Time-aware evaluation can be used to prevent leakage between historical and future observations. Second, additional models such as gradient boosting and LSTM networks can be compared with KNN, SVM, and Random Forest using precision, recall, F1-score, calibration, and uncertainty measures rather than accuracy alone.

Third, geographic information can be integrated with hotspot mapping to identify spatial concentrations and support resource planning. Fourth, external contextual variables such as weather, mobility, or appropriately governed public information may improve forecasting. Finally, secure data management, fairness evaluation, human-in-the-loop review, and continuous monitoring should be included before operational deployment.

THREATS TO VALIDITY

Internal validity is affected by the inconsistent accuracy figures reported across different sections of the source document, since it is not possible to determine from the available material which value reflects the final, correctly evaluated model. Construct validity depends on whether the labeled crime categories in the dataset correspond consistently to the legal or operational definitions used by law-enforcement agencies; ambiguity here could affect both training and interpretation of results.

External validity is limited by the geographic and temporal scope of the underlying crime records; patterns learned from one region or period may not generalize to another without re-validation. Finally, conclusion validity depends on the use of appropriate evaluation protocols, such as held-out test sets and time-aware splits, which should be explicitly documented before the reported performance is used to support operational decisions.

REFERENCES

- [1] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011, doi: 10.1198/jasa.2011.ap09546.
- [2] T. Wang, C. Rudin, D. Wagner, and R. Sevieri, "Learning to detect patterns of crime," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2013.
- [3] W. L. Perry, B. McInnis, C. C. Price, S. C. Smith, and J. S. Hollywood, *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations*. RAND Corporation, 2013.
- [4] M. Akshaya and A. Suruliandi, "Crime prediction and classification using K-means clustering and SVM," *International Journal of Pure and Applied Mathematics*, vol. 115, no. 6, pp. 1–6, 2017.



-
- [5] S. Chainey, L. Tompson, and S. Uhlig, "The utility of hotspot mapping for predicting spatial patterns of crime," *Security Journal*, vol. 21, nos. 1–2, pp. 4–28, 2008, doi: 10.1057/palgrave.sj.8350066.
- [6] M. Kaur and R. Sharma, "Crime forecasting using LSTM neural networks," in *Proc. 2021 Int. Conf. Computing, Communication and Automation (ICCCA)*, IEEE, 2021, doi: 10.1109/ICCCA49541.2021.9302647.
- [7] S. Sharma and R. Singh, "Blockchain-based secure storage of crime records," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 3, 2020.