

CDAPS – Crime Data Analysis & Prediction System Using Time Series Forecasting

Harshad Ade¹, Prof. A. H. Auti², Anish Bansod³, Samyak Bhaisare⁴, Sourav Dutta⁵

Department of Computer Engineering, Sinhgad Academy of Engineering, Kondhwa (BK), Pune

Abstract - Traditional study planning and scheduling systems are mostly static. They often struggle to keep up with changing academic needs, learning speeds, and student performance. This paper presents an AI-Based Dynamic Study Management System to support efficient learning and improve academic productivity. It achieves this through intelligent scheduling, NLP-based text summarization, and adaptive quiz creation. The system treats study topics as tasks that can be scheduled dynamically. It assigns them to available time slots based on their duration, priority, deadlines, and user performance. AI models help summarize content automatically, and adaptive quizzes are created using AI-driven prompt techniques. The architecture is scalable and responsive, built with React, FastAPI, TensorFlow, and MongoDB. The system also uses reinforcement learning to reschedule adaptively and provide personalized study suggestions. The implementation is analyzed and compared to current AI systems, highlighting its strengths and weaknesses in speed, privacy, adaptability, and efficiency. This solution aims to lighten the load for students, improve retention rates, and provide a smarter alternative to traditional study planning methods.

Key Words: Artificial Intelligence, NLP, Prompt Engineering Techniques, FastAPI

1. INTRODUCTION

Crime is one of the biggest challenges facing urban growth, economic stability, and public safety around the world. Rapid urbanization, population growth, economic differences, and technological changes have greatly affected the nature and spread of crime. As cities grow, law enforcement agencies must handle larger amounts of crime-related information while making quick operational decisions. Traditional crime analysis methods rely on past reports and manual interpretation, making it tough to spot new patterns and predict future incidents. Recent progress in artificial intelligence, machine learning, and data analysis has opened new ways to understand crime trends using data-driven methods. Predictive analysis helps uncover hidden relationships, time trends, and repeated behaviors in large crime datasets. This enables law enforcement to shift from reactive policing to proactive strategies focused on preventing crime and reducing risks. Time-series forecasting techniques have delivered positive results in various fields, including finance, traffic control, healthcare analysis, and environmental monitoring. We can apply similar methods to crime data to predict future incident rates and identify times of higher criminal risk. Moreover, interactive visualization

systems can convert complex analytical results into easy-to-understand graphics, aiding decision-making for policymakers and public safety officials. The growing access to digital crime records and computing power sets a solid base for creating smart crime forecasting systems that can provide accurate predictions and useful insights for urban areas.

2. LITERATURE SURVEY

Recent advancements in machine learning and data analytics have led to the creation of intelligent crime prediction systems. Researchers have applied algorithms like Decision Trees, Random Forest, KNN, Naïve Bayes, and XGBoost to predict crime categories and find areas prone to crime. Ensemble models usually provide higher accuracy and better generalization than traditional classifiers. Spatio-temporal analysis has become common in studying crime patterns by looking at both geographical and timing factors. Techniques such as hotspot detection and geospatial mapping help identify high-risk locations and assist law enforcement agencies in allocating resources more efficiently. Time-series forecasting methods such as ARIMA, SARIMA, Prophet, and LSTM have been used to predict future crime trends. SARIMA is particularly effective with seasonal crime data and often delivers reliable results, even with limited historical records. Deep learning approaches have further enhanced crime analytics by automatically extracting features and recognizing patterns. However, these methods need large datasets and significant computational resources, which can limit their practical use. Recent advances in artificial intelligence, machine learning, and data analytics have opened new opportunities for understanding crime dynamics through data-driven methods. Predictive analytics helps uncover hidden relationships, trends over time, and recurring behavior patterns in large crime datasets. These capabilities enable authorities to shift from reactive policing to proactive strategies aimed at preventing crime and managing risks.

Time-series forecasting techniques have shown positive results in various fields, including financial forecasting, traffic management, healthcare analytics, and environmental monitoring. Similar methods can be applied to crime data to estimate future incident rates and pinpoint times of increased criminal risk. Additionally, interactive visualization systems can change complex analytical results into clear graphical representations that aid decision-making among policymakers and public safety officials.

The growing availability of digital crime records and computing resources provides a strong basis for creating

intelligent crime forecasting systems that can generate accurate predictions and helpful insights for urban areas.

3. DATASET AND PREPROCESSING

The performance of any crime prediction system depends significantly on the quality and reliability of the dataset used for model development. In this research, historical crime records collected from multiple Indian cities between 2020 and 2024 were utilized to analyze crime patterns and forecast future incidents. The dataset consists of approximately 40,160 records containing information such as city name, crime category, date of occurrence, and monthly crime count. The data covers multiple categories of crimes, including violent crimes, property crimes, financial crimes, and public order offences, providing sufficient diversity for machine learning and time-series analysis.

Before model development, the dataset underwent a comprehensive preprocessing stage to improve data quality and ensure consistency. Initially, duplicate records and incomplete entries were identified and removed to eliminate redundancy and prevent biased predictions. Missing values in categorical attributes were replaced using appropriate labels, while numerical inconsistencies were corrected through validation against neighbouring observations. Date fields were converted into a standard datetime format, allowing the extraction of temporal attributes such as year, month, quarter, and season. Since crime forecasting relies on time-series analysis, individual crime records were aggregated into monthly crime counts for each city, creating a structured dataset suitable for forecasting models.

To improve predictive performance, several feature engineering techniques were applied. Temporal features including month, quarter, and year were extracted to capture seasonal crime behaviour. Cyclical encoding using sine and cosine transformations was employed for the month attribute to preserve its periodic nature. Historical dependency was incorporated through lag features representing previous monthly crime counts, while rolling mean and rolling standard deviation were calculated over three-month and six-month windows to capture short-term and long-term trends. These engineered features enable the forecasting models to learn recurring crime patterns more effectively.

Exploratory Data Analysis (EDA) was performed to understand the statistical characteristics of the processed dataset before model training. City-wise crime distribution, category-wise frequency, monthly trends, and correlation analysis were examined to identify hidden relationships among variables. The analysis revealed clear seasonal fluctuations and recurring crime patterns across several cities, supporting the selection of SARIMA and other forecasting models. The final processed dataset was divided into training and testing subsets using an 80:20 ratio, ensuring that chronological order was preserved to avoid data leakage during forecasting experiments. This preprocessing pipeline provides a reliable foundation for machine learning, anomaly detection, and crime forecasting while improving the robustness and accuracy of the proposed framework.

Table 3.1 Dataset Overview

Parameter	Description
Dataset Size	Approximately 40,160 Records
Time Period	January 2020 – December 2024
Cities Covered	29 Indian Cities
Forecast Interval	Monthly
Target Variable	Monthly Crime Count
Data Type	Structured Time-Series Dataset

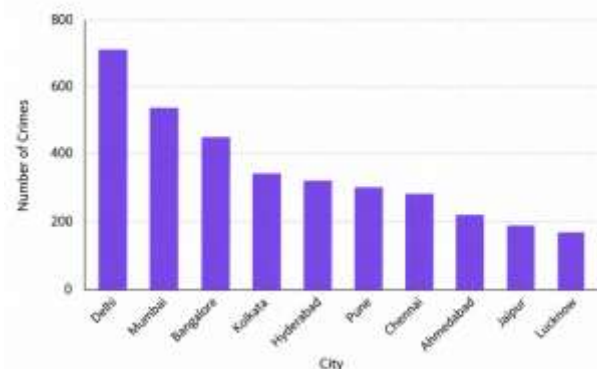
Figure 3.1 Dataset Preprocessing Pipeline

Figure 3.1 Dataset Preprocessing Pipeline



Figure 3.2 City-wise Crime Distribution

Top Cities by Crime Volume (2024)



4. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed Crime Analytics and Forecasting Framework is designed to transform historical crime records into meaningful predictions and analytical insights. The system consists of six major modules: data collection, preprocessing, feature engineering, forecasting, anomaly detection, and visualization. The overall workflow begins with collecting crime records from structured datasets containing information such as city, crime category, date, and incident count. These records are stored and processed to create a clean and consistent analytical dataset.

During preprocessing, missing values, duplicate records, and inconsistent entries are removed. The cleaned data is then aggregated on a monthly basis to support time-series forecasting. Several engineered features including month encoding, lag variables, and rolling statistical measures are generated to improve predictive performance. These features help the models capture seasonality, recurring trends, and temporal dependencies present in crime data.

The machine learning module consists of Random Forest, Gradient Boosting, XGBoost, and LightGBM algorithms. These models are trained to learn relationships between historical crime patterns and future crime occurrences. Their performance is evaluated using standard regression metrics to identify the most suitable model for crime estimation tasks.

For forecasting future crime trends, SARIMA, Prophet, and LSTM models are implemented and compared. SARIMA is particularly effective in capturing seasonal crime behavior, while Prophet handles trend decomposition and LSTM learns temporal dependencies through recurrent neural networks. The forecasting module generates monthly crime predictions for individual cities and crime categories.

An anomaly detection module is integrated into the framework to identify unusual crime spikes and abnormal behavior patterns. Statistical techniques such as Z-Score and Interquartile Range (IQR) are used to detect observations that significantly deviate from historical trends. These anomalies may indicate special events, sudden crime outbreaks, or unusual regional activities that require further investigation.

The final component of the framework is the visualization and reporting module. A FastAPI-based backend communicates with an interactive dashboard that presents crime trends, forecasting results, anomaly alerts, and comparative city-level analysis. The dashboard enables users to explore historical records, monitor future predictions, and generate analytical reports for decision-making purposes.

Figure 4.1 Overall System Architecture

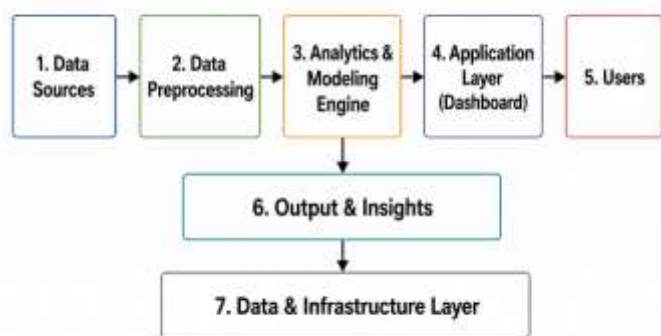


Figure 4.1 Overall System Architecture

5. ALGORITHMIC DESIGN AND IMPLEMENTATION

5.1 Feature Engineering and Regression Module

Each monthly, city-level crime record is represented as a feature vector x_i that combines temporal, categorical, and

autoregressive attributes, paired with a target variable y_i denoting the actual crime count for that period. To preserve the cyclical nature of time, the month attribute is encoded through sine and cosine transformations rather than a simple integer label, which prevents the model from misinterpreting December and January as distant values. Beyond this temporal encoding, the feature set is enriched with a one-month lag term, a rolling mean, and a rolling standard deviation, all of which help the model pick up on short-term momentum and recent volatility in crime activity.

Using this combined feature set, four ensemble-based regressors — Random Forest, Gradient Boosting, XGBoost, and LightGBM — are trained under a common preprocessing pipeline and evaluated on an 80:20 train-test split, ensuring a fair comparison across models. Performance is assessed using **MAE, MSE, RMSE, and R^2** , the standard metrics for regression accuracy.

5.2 Time-Series Forecasting Module

Given a city-level monthly crime sequence $D = \{s_1, s_2, \dots, s_n\}$, three forecasting approaches — SARIMA, Prophet, and LSTM — are trained independently on the historical data. The SARIMA model's order parameters are tuned via grid search to identify the best-fitting seasonal and non-seasonal components, while Prophet decomposes the series into trend and seasonality terms to model longer-range patterns. The LSTM network, by contrast, learns from sliding windows of past observations and applies early stopping during training to guard against overfitting on a relatively short crime series. Each model's forecasts are then compared across both short-term and long-term horizons to determine which technique generalizes better as the prediction window extends.

5.3 Anomaly Detection Module

To flag unusual fluctuations in crime activity, each city-month observation is evaluated against its own rolling mean and standard deviation using a Z-score calculation. Records whose Z-score exceeds a defined threshold are marked as anomalous. Despite its simplicity, this statistical approach is effective at surfacing irregular spikes — whether driven by seasonal effects, one-off events, or unexpected disturbances — while remaining computationally lightweight enough for routine, large-scale monitoring.

5.4 Integration and Control Flow

The overall system is built around a service-oriented architecture: incoming requests are first validated and then routed to the relevant module — prediction, forecasting, or analytics — depending on the nature of the query. Once processed, the backend returns the result as a structured JSON payload to the frontend visualization layer. Because each model's implementation is encapsulated within its own service, individual models can be updated, retrained, or replaced without disrupting the broader API or the user-facing interface.

6. MATHEMATICAL FORMULATION

The proposed Crime Analytics and Forecasting Framework combines machine learning, time-series forecasting, and anomaly detection to predict crime trends and flag unusual patterns in crime activity. The framework relies on careful

feature engineering, predictive modeling, and statistical validation, each described below.

Feature Engineering

To help the models capture seasonal patterns in crime data, the *month* feature is encoded using a cyclical (sine-cosine) transformation rather than treated as a simple numeric or categorical variable. This avoids the artificial discontinuity that would otherwise occur between December and January:

$$\begin{aligned} Month_{sin} &= \sin\left(\frac{2\pi \times Month}{12}\right), Month_{cos} \\ &= \cos\left(\frac{2\pi \times Month}{12}\right) \end{aligned}$$

In addition, historical crime behavior is captured through **lag features**, which represent the crime count observed k periods before the current time step:

$$Lag_k = CrimeCount_{t-k}$$

and **rolling mean features**, which smooth short-term fluctuations by averaging crime counts over a window of n preceding periods:

$$RollingMean_n = \frac{1}{n} \sum_{i=1}^n CrimeCount_i$$

Together, these engineered features allow the models to learn both seasonal cycles and short-term momentum in crime trends.

Predictive Modeling

Crime counts are predicted using ensemble learning methods, namely Random Forest, Gradient Boosting, XGBoost, and LightGBM. These were chosen for their ability to model non-linear relationships and interactions among engineered features without requiring strict distributional assumptions.

For Random Forest, the final prediction is obtained by averaging the outputs of N individual decision trees, $T_i(x)$:

$$\hat{Y} = \frac{1}{N} \sum_{i=1}^N T_i(x)$$

To capture explicit temporal and seasonal structure, the **Seasonal Autoregressive Integrated Moving Average (SARIMA)** model is also applied, denoted as $SARIMA(p, d, q)(P, D, Q)_s$, where (p, d, q) represent the non-seasonal autoregressive, differencing, and moving-average orders, (P, D, Q) represent their seasonal counterparts, and s is the length of the seasonal cycle. SARIMA's forecasts are benchmarked against two additional forecasting approaches — **Prophet** and **Long Short-Term Memory (LSTM)** networks — to assess which method best captures the underlying temporal dynamics of crime data.

Anomaly Detection

Unusual spikes or drops in crime activity are identified using the **Z-score** method, which measures how many standard deviations an observation lies from the mean:

$$Z = \frac{x - \mu}{\sigma}$$

where x is the observed crime count, μ is the mean crime count, and σ is the standard deviation. Observations satisfying $|Z| > 3\sigma$ are flagged as statistical outliers, signaling potentially anomalous crime activity that warrants further investigation.

Model Evaluation

Model performance is assessed using a combination of standard regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2).

RMSE, which penalizes larger errors more heavily, is computed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

while R^2 , which indicates the proportion of variance in crime counts explained by the model, is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where y_i is the actual crime count, $\hat{y}_i$ is the predicted crime count, and $\bar{y}$ is the mean of the actual values.

Collectively, these mathematical formulations — spanning feature construction, model architecture, anomaly thresholds, and evaluation criteria — provide the theoretical foundation underpinning the proposed framework's forecasting and anomaly detection capabilities.

7. EXPERIMENTAL SETUP AND IMPLEMENTATION ENVIRONMENT

The proposed Crime Analytics and Forecasting Framework was implemented and evaluated using a standardized experimental setup to ensure fair comparison among machine learning and forecasting models. All experiments were conducted on a system equipped with an AMD Ryzen 5 5500U processor, 8 GB RAM, SSD storage, and Windows 11 operating system. The framework was developed using Python 3.11 with FastAPI as the backend framework. Data processing and analysis were performed using Pandas and NumPy, while machine learning models were implemented using Scikit-Learn. Time-series forecasting was carried out using Statsmodels, deep learning models were developed with TensorFlow/Keras, and interactive visualizations were created using Plotly and Streamlit. PostgreSQL was used for data storage, while Git and GitHub facilitated version control.

The crime dataset underwent a comprehensive preprocessing pipeline involving data cleaning, missing value handling, duplicate removal, temporal aggregation, feature engineering, and normalization. The processed dataset was transformed into monthly city-level crime records suitable for predictive

analysis. To evaluate model performance, the dataset was divided chronologically into 80% training data and 20% testing data, ensuring preservation of temporal dependencies and realistic forecasting conditions.

Multiple machine learning models, including Random Forest, Gradient Boosting, XGBoost, and LightGBM, were trained using optimized hyperparameters obtained through iterative experimentation. Time-series forecasting was primarily performed using the SARIMA model configured to capture yearly seasonality, while Prophet and LSTM models were implemented for comparative analysis. The LSTM architecture consisted of an input layer, a hidden layer with 64 units, a dropout rate of 0.2, a dense layer of 32 neurons, and a single output neuron for forecasting future crime counts.

The implementation workflow comprised six stages: data collection, preprocessing and feature engineering, model training, forecast generation, anomaly detection, and visualization. A FastAPI-based backend was developed to support model loading, prediction services, data retrieval, and dashboard communication. The visualization module provided crime trend analysis, city-wise comparisons, forecast dashboards, anomaly alerts, and analytical reporting features. Model performance was evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). This experimental environment ensures reliable, scalable, and reproducible evaluation of the proposed crime analytics framework.

8. RESULTS AND DISCUSSION

The performance of the proposed Crime Analytics and Forecasting Framework was evaluated using multiple machine learning and forecasting algorithms. Experimental analysis was conducted on the processed crime dataset to assess prediction accuracy, forecasting capability, and anomaly detection effectiveness. Ensemble learning algorithms including Random Forest, Gradient Boosting, XGBoost, and LightGBM were compared using standard regression metrics. Similarly, SARIMA, Prophet, and LSTM models were evaluated for monthly crime forecasting.

Among the regression models, Gradient Boosting achieved the most consistent performance due to its ability to iteratively minimize prediction errors. Random Forest produced stable results while XGBoost and LightGBM demonstrated competitive performance with faster training times. The comparative analysis indicates that ensemble learning techniques effectively capture complex relationships present in crime datasets.

The forecasting experiments revealed that SARIMA outperformed Prophet and LSTM under the available dataset conditions. Since crime records exhibit strong seasonal behavior, SARIMA successfully captured recurring monthly patterns and generated more reliable forecasts. Prophet accurately modeled long-term trends but occasionally

struggled with sudden fluctuations. LSTM showed promising results but required larger datasets for optimal performance.

The anomaly detection module successfully identified abnormal crime spikes and unusual activity patterns. Several anomalies corresponded to periods with significantly higher crime counts compared to historical averages. The integration of anomaly detection with forecasting enhances the practical usefulness of the framework by providing both predictive insights and early warning capabilities.

9. CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

The growing availability of crime-related data has opened up significant opportunities for creating smart analytical systems that support proactive law enforcement strategies. However, many traditional crime management systems still rely mostly on historical reports and descriptive statistics, which provide limited help in predicting future crime trends or spotting new criminal activities. To tackle these issues, this research proposed a Crime Analytics and Forecasting Framework that combines machine learning, time-series forecasting, anomaly detection, and interactive visualization into a single decision-support platform.

This framework was designed to turn raw historical crime records into useful predictive insights through a systematic process that includes data collection, preprocessing, feature engineering, model training, forecasting, anomaly detection, and visualization. Historical crime data collected from various Indian cities were cleaned, combined, and turned into a structured dataset for predictive analysis. Different feature engineering techniques, such as temporal feature extraction, cyclical encoding, lag variables, and rolling statistical measures, were used to capture seasonal behavior and historical dependencies in the dataset. These preprocessing steps significantly improved the dataset's quality and boosted the learning ability of the prediction models used.

To assess the effectiveness of different predictive methods, four ensemble machine learning algorithms—Random Forest, Gradient Boosting, XGBoost, and LightGBM—were implemented and compared for crime estimation tasks. Additionally, three forecasting techniques, SARIMA, Prophet, and Long Short-Term Memory (LSTM), were tested to find the best model for predicting future crime incidents. The comparative analysis showed that the SARIMA model consistently delivered reliable forecasting results by effectively capturing seasonal trends and repeating crime patterns found in the historical data. Among the machine learning models, Gradient Boosting performed best due to its ability to iteratively reduce prediction errors and model complex nonlinear relationships within structured crime data.

Along with forecasting, using statistical anomaly detection methods further improved the framework's capabilities.

Implementing the Z-Score and Interquartile Range (IQR) methods allowed the system to identify unusual fluctuations and unexpected spikes in crime activity that may require immediate action from law enforcement agencies. Recognizing such anomalies can help authorities identify strange events, investigate possible security threats, and take timely preventive measures. The visualization module, built with a FastAPI-based backend and an interactive dashboard, successfully converted complex analytical outputs into easy-to-understand graphs, charts, and comparative reports, making the forecasting results available to both technical users and decision-makers.

In summary, the proposed framework shows that combining machine learning, statistical forecasting, anomaly detection, and visual analytics can significantly enhance crime intelligence and support data-driven decision-making. Instead of just being a crime prediction model, the framework acts as a complete analytical platform that can help law enforcement agencies, policymakers, researchers, and smart city administrators understand historical crime behavior, anticipate future trends, and optimize resource allocation. The system's modular design also ensures scalability, allowing for the addition of more analytical components and data sources with minimal changes. Therefore, this framework represents a practical step toward building intelligent, data-driven public safety systems that can support proactive policing and urban crime management.



Figure 8.1 Dashboard

9.2 Future Scope

Although the proposed framework shows promising results in crime forecasting and data visualization, there are several ways to improve its predictive ability and practical use. The current system mainly relies on organized historical crime records. Future versions could include real-time crime data from police information systems, emergency reporting platforms, and smart city surveillance networks. Adding live data would enable dynamic forecasting, allowing authorities to track changing crime patterns and respond better to new incidents.

Another promising direction is to expand the framework by integrating Geographic Information System (GIS) technology. By merging predictive analytics with geospatial visualization,

the system could create crime hotspot maps at street or neighborhood levels. This would help in accurately identifying high-risk areas and support smart patrol planning, resource use, and crime prevention strategies tailored to specific locations. Adding more context, like weather data, demographic details, economic indicators, population density, traffic trends, and public event schedules, could improve forecasting accuracy by considering outside factors that affect criminal behavior.

Future research might also explore using advanced deep learning and artificial intelligence techniques. This includes Transformer-based forecasting models, Temporal Fusion Transformers (TFT), Graph Neural Networks (GNNs), and blended forecasting models that mix statistical and neural network methods. These approaches have shown good results in complex prediction tasks and may yield better accuracy as larger datasets become available. Additionally, using Explainable Artificial Intelligence (XAI) methods like SHAP and LIME could clarify prediction results by pinpointing the factors behind forecasted crime trends, which would build trust among law enforcement.

From an implementation standpoint, using secure cloud infrastructure for the framework would allow for scalable multi-user access and integration with government systems. Developing mobile apps and web portals would also enhance accessibility for field officers and administrative staff. Furthermore, applying Natural Language Processing (NLP) techniques to analyze police reports, online news, emergency call logs, and social media could provide additional insights, improving forecasting and helping catch emerging criminal activities early. Together, these advancements would turn the proposed framework into a complete intelligent policing platform that supports real-time crime monitoring, efficient resource allocation, strategic planning, and evidence-based public safety management in future smart city settings.

10. REFERENCES

- [1] M.-S. Baek, W. Park, J. Park, K.-H. Jang, and Y.-T. Lee, "Smart Policing Technique With Crime Type and Risk Score Prediction Based on Machine Learning for Early Awareness of Risk Situation," *IEEE Access*, vol. 9, pp. 131906–131921, 2021.
- [2] S. Hossain, A. Abtahee, I. Kashem, M. M. Hoque, and I. H. Sarker, "Crime Prediction Using Spatio-Temporal Data," in *Communications in Computer and Information Science (CCIS)*, Springer, vol. 1235, pp. 277–289, 2020.
- [3] S. Sridharan, S. N., V. S., and P. Santhi, "Crime Prediction Using Machine Learning," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024.
- [4] G. García-Zanabria, M. Angelini, N. Ferreira, and C. Turkay, "CriPAV: Street-Level Crime Patterns Analysis and Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4000–4016, 2022.

- [5] S. Tuarob, S. Chotiprapa, and M. H. Ashaduzzaman, "Beyond Administrative Reports: A Deep Learning Framework for Classifying and Monitoring Crime and Accidents Leveraging Large-Scale Online News," *Neural Computing and Applications*, vol. 37, pp. 7183–7205, 2025.
- [6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.
- [7] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015.
- [8] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," *PLOS ONE*, vol. 13, no. 3, 2018.
- [9] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [10] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.