International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Crime Prediction and Analysis using Machine Learning

[1]A.S. Arunachalam , [2]Hemanathan D and [3]Dinesh S
[1]Professor, [2]Email: arunachalam1976@gmail.com
[2&3]U.G. Scholar, [2]Email: hemanadhan546d@gmail.com [3] Email:ddinesh9337@gmail.com
Department of Computer Science and Information Technology,
Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, India.

https://doi.org/10.55041/ijsmt.v2i5.040

Cite this Article: D, H. & S, D. (2026). Crime Prediction and Analysis using Machine Learning. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.040

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract

Crime prediction has become an important application of artificial intelligence because public safety agencies need faster and more reliable ways to identify crime trends. This project presents CrimeCast, a web-based crime prediction and analysis system developed with Python and Flask, trained on crime data from Tamil Nadu, India. The system uses a Random Forest Classifier to predict the most likely crime type from inputs such as state, city, latitude, longitude, year, and domestic status. The model is designed to classify six major crime categories: Assault, Burglary, Cyber Crime, Domestic Violence, Robbery, and Theft. The application combines machine learning with a secure, responsive web interface built with Bootstrap and an SQLite database for storing user accounts and prediction history. It also includes analytics dashboards that show crime distribution, yearly trends, city-wise frequency, and domestic versus non-domestic comparisons. An interactive heatmap built with Leaflet.js provides a geographic visualization of crime density across Tamil Nadu. With its prediction engine, history tracking, and visual analytics, CrimeCast acts as a decision-support tool for law enforcement agencies, planners, and researchers.

Key words: Machine Learning, Artificial intelligence, Data Analytics, and Random Forest Classifier.

1. Introduction

Crime is a persistent concern in modern society, especially in rapidly urbanizing regions where population density, economic disparity, and technological change influence criminal activity. Traditional crime prevention methods often depend on manual investigation and retrospective reporting, which are reactive and may not provide timely guidance for prevention. The growth of Machine Learning and Data Analytics offers a more proactive way to understand crime patterns. By analyzing historical records, a system can learn relationships between location, time, and crime type and then use those patterns to make predictions about future incidents. This makes it possible to support preventive action instead of only responding after a crime occurs.

CrimeCast is designed as a web-based crime prediction and analysis platform that brings together machine learning, secure authentication, data storage, and visualization in a single application. The system is intended to be easy to use while still delivering reliable predictions and useful insights through charts, dashboards, and maps.

2. Methodology

The project follows a structured development methodology that begins with data collection. Crime records specific to Tamil Nadu are gathered and organized into a dataset containing important attributes such as state, city, latitude, longitude, year, and domestic status.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

The next stage is data preprocessing. Missing values are handled, duplicate records are removed, and inconsistent entries are cleaned. Categorical fields such as state and city are encoded into numerical form so that they can be used by the machine learning model. Feature selection is then performed to focus on the variables most relevant to crime prediction.

After preprocessing, the dataset is split into training and testing sets. A Random Forest Classifier is trained on the training data and evaluated using standard metrics such as accuracy, precision, recall, and F1-score. This ensures that the model is tested on unseen records before deployment.

Once the model performs satisfactorily, it is integrated into a Flask-based web application. The backend receives user inputs, processes them, and passes them to the model for prediction. The frontend is built with HTML, CSS, and Bootstrap to provide a responsive and intuitive interface. Predictions, confidence scores, and historical records are stored in SQLite for later analysis.

3. System Design and Modules

CrimeCast is organized as a three-layer web system. The presentation layer handles user interaction through the web interface. The application layer, powered by Flask, manages request processing, authentication, preprocessing, and model inference. The data layer stores user details, prediction history, and crime-related records using SQLite.

The system contains multiple modules that work together. The User Authentication Module manages registration, login, session control, and password hashing. The Data Input Module accepts crime-related parameters from the user with validation checks to reduce errors. The Data Preprocessing Module transforms raw input into a model-ready format.

The Prediction Module is the core intelligence of the system. It uses the Random Forest model to classify the likely crime type and generates a confidence score. The Database Management Module stores user accounts and prediction history, while the Prediction History Module lets users review previous results and compare them over time.

The Data Analytics Module presents crime patterns through charts and graphs, helping users understand overall crime distribution, yearly variations, and city-wise comparisons. The Geographic Visualization Module extends this by displaying crime density on an interactive heatmap, making high-risk areas easy to identify.

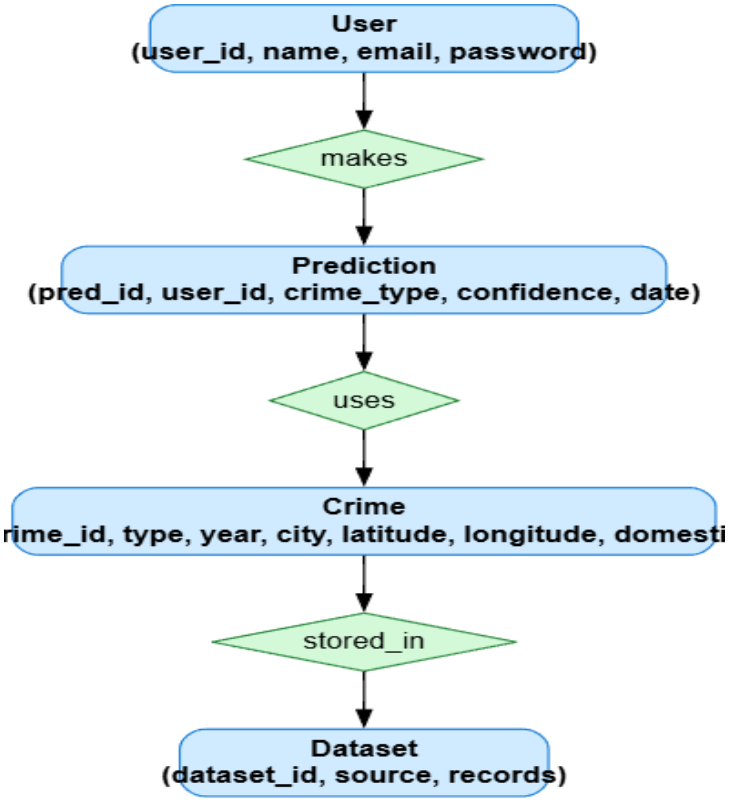

Figure 1Modules Structure

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

4. Advantages and Limitations

The main advantages of the system are its high prediction accuracy, real-time response, secure login mechanism, interactive analytics, and geographic visualization. It reduces manual effort and supports proactive decision-making. At the same time, the system depends on the quality and completeness of the training data. Predictions are limited to the categories present in the dataset, so sudden social changes or unusual events may not be captured fully.

5. Future Scope

Future versions of CrimeCast can incorporate additional machine learning techniques, including gradient boosting and deep learning models, to further improve accuracy. The system can also be expanded to handle real-time feeds from official crime records or other updated sources. A mobile interface, richer GIS mapping, and role-based access control would improve accessibility and security. Continuous retraining will help the model stay relevant as crime patterns evolve.

6. Key Implementation Highlights

The implementation of CrimeCast is centered on simplicity and maintainability. The training script reads the crime dataset, encodes categorical values, and fits a Random Forest model that can be reused by the web application. Saving the model with joblib makes deployment practical because the trained classifier can be loaded instantly without retraining at runtime. The Flask application handles the full user journey: registration, login, dashboard access, prediction submission, history review, analytics display, and heatmap visualization. Each route is kept focused on a single responsibility, which makes the code easier to understand and extend.

Security is handled through password hashing and session-based authentication. This is important because the application stores user login details and prediction history. In addition, database operations are centralized, which supports consistent storage and retrieval of records. The prediction interface accepts location and year information and converts them into the format required by the model. Confidence scores provide a basic indicator of certainty, making the output more informative for users who need a quick risk assessment.

7. Testing and Validation

Testing was an essential part of the development process. The model was validated using a train-test split so that its accuracy could be measured on unseen data. This gave a realistic estimate of how well the system would perform in practice.

The web application was also checked for user flow, input handling, and database interaction. Error handling was important because invalid or missing values could otherwise interrupt the prediction process. The application therefore includes safeguards that redirect the user when login or form validation fails. The analytics and heatmap features were reviewed to ensure that stored data is converted into meaningful visual output. These visual components strengthen the system by helping users identify patterns rather than only seeing a single predicted label.

8. Overall Contribution

CrimeCast contributes a practical framework for combining prediction, visualization, and secure web interaction in one application. Rather than treating crime analysis as a static reporting problem, the system shows how machine learning can be used to support dynamic decision-making with measurable outputs and interactive insights. The design is valuable because it connects technical prediction with usable presentation. A model alone may be accurate, but a full system becomes more useful when the results are shown in dashboards, histories, and geographic maps. This increases transparency and helps users understand how crime patterns vary across time and location.

In an academic context, the project demonstrates end-to-end software development skills, including dataset preparation, model training, Flask integration, database design, authentication, and visualization. In a practical context, it provides a foundation that can be extended for future public safety applications.

9. Results and Discussion

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

The model trained in CrimeCast achieves strong predictive performance and provides useful crime classification for the selected categories. Based on the uploaded project report, the system reaches an overall accuracy of approximately 92 percent, which demonstrates that the selected features and Random Forest approach are effective for this dataset. Beyond prediction accuracy, the project adds value through explainable presentation. Users can view the predicted crime type, inspect confidence levels, and explore supporting analytics. This combination of prediction and visual analysis makes the system more practical than a standalone classifier because it helps users interpret the results.

The prediction history feature is especially useful for repeated analysis. It enables the user to review past inputs, identify recurring locations, and compare outputs over time. The heatmap also supports spatial understanding by showing where incidents are concentrated, which can guide targeted patrol planning and resource allocation.

The system is lightweight and suitable for deployment on standard hardware. Since it uses Python, Flask, and SQLite, it can be run locally or hosted on a small server without requiring expensive infrastructure. This makes it a useful prototype for educational, research, and decision-support purposes.

10. Conclusion

CrimeCast demonstrates how machine learning can be applied to a real-world public safety problem. By combining data preprocessing, a Random Forest prediction engine, secure web access, and visual analytics, the system converts raw crime data into actionable information.

The project shows that a well-structured, data-driven system can support smarter planning and more proactive crime prevention. With future enhancements such as real-time data integration, advanced learning models, and broader geographic coverage, the system can become even more powerful and practical for law enforcement and policy analysis.

References

[1]. N. Shah, N. Bhagat, and M. Shah, “Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention,” Visual Computing for Industry, Biomedicine, and Art, vol. 4, no. 1, pp. 1–14, 2021.

[2]. S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha,

[3]. “An empirical analysis of machine learning algorithms for crime prediction using stacked generalization,” IEEE Access, vol. 9, pp. 67488–67500, 2021.

[4]. W. Safat, S. Asghar, and S. A. Gillani,“Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques,” IEEE Access, vol. 9,
pp. 70080–70094, 2021.

[5]. N. Kanimozhi, N. V. Keerthana, G. S. Pavithra, G. Ranjitha, and S. Yuvarani,“Crime type and occurrence prediction using machine learning algorithm,”in Proc. Int. Conf. Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 266–273.

[6]. M. Sabarish and A. S. Arunachalam, "A Trust Secure Attacker Detection with Upgraded Deep Learning-Assistance for SDN Networks," *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, India, 2023, pp. 121-125, doi: 10.1109/SMART59791.2023.10428532.

[7]. H. Al-Ghushami, D. Syed, J. Sessa, and A. Zainab, “Intelligent automation of crime prediction using data mining, “in Proc. IEEE 31st Int. Symp. Industrial Electronics (ISIE), 2022.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.