iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76270

CityGuard++: Crime Detection System and Hotspot Analysis

Dr. Vidyarani H J[1], Dr. KavithaDevi C S[2], Prakash R Annigeri[3], Nehaal R[4], Kishan AS[5]
Department of Computer Science and Business Systems Engineering Dr. Ambedkar Institute of Technology, Bangalore, India

Abstract: Crime reporting and analysis remain essential components of modern public safety. CityGuard++ is a hybrid, web-centric system that enables citizens to report incidents online, supports administrator verification, and offers ML-driven analysis and geospatial visualizations for hotspot detection and trend forecasting. The platform combines deterministic preprocessing, lightweight NLP classification, clustering for hotspot discovery, and a bounded forecasting pipeline to preserve compute predictability. Data sources include citizen reports, historical crime datasets, and optional public records. CityGuard++ outputs verified incident records, interactive dashboards, and calibrated confidence scores for predicted hotspots. We evaluate the system on a regional crime dataset and simulated citizen reports; results show robust classification performance (accuracy ~89% on balanced test splits), meaningful hotspot detection (Silhouette score ~0.7), and actionable visualizations. The system is designed for auditability, privacy-preserving storage, and incremental deployment across municipalities. Key contributions include (i) an end-to-end verified reporting → analytics pipeline, (ii) a deterministic ML orchestration pattern for predictable cost at inference time, and (iii) an evidence-provenance mechanism for flags and hotspot suggestions.

Keywords: crime analysis, crime reporting, hotspot detection, machine learning, geospatial visualization, digital policing.

I. INTRODUCTION

Crime prevention and public safety have become increasingly complex challenges in rapidly urbanizing societies. Traditional crime-reporting mechanisms—such as in-person complaints, phone-based reporting, and paper documentation—often lead to delayed responses, incomplete information, and limited transparency. These constraints not only reduce citizen participation in reporting incidents but also create significant barriers for law-enforcement agencies attempting to maintain accurate crime records and develop proactive strategies. As digital adoption increases, there is a clear need for smarter, data-driven platforms capable of supporting real-time reporting, automated analysis, and predictive insights.

CityGuard++ is developed to address this gap by providing a comprehensive digital ecosystem for crime reporting, verification, visualization, and machine-learning analysis. Instead of relying solely on manual administrative processes, the platform enables citizens to file incident reports through an easy-to-use web interface, enriching each submission with descriptions, categorization, and optional multimedia evidence. These digital reports are passed through a structured verification workflow handled by authorized administrators or police officials, ensuring the authenticity of the information before it becomes part of the official dataset. This not only accelerates data collection but also improves the quality and reliability of stored records.

A key innovation in CityGuard++ lies in its integration of intelligent analytical components. Verified reports are processed through natural language processing (NLP) pipelines that extract meaningful patterns from textual descriptions and classify crime types automatically. Geospatial clustering algorithms are applied to incident locations to discover crime hotspots, revealing concentration patterns that may otherwise remain undetected in traditional tabular datasets. The system further includes time-series forecasting methods to predict short-term crime trends, enabling agencies to allocate resources more strategically and respond proactively to emerging risks. The platform's dashboard provides interactive visualizations, including heatmaps, bar charts, and temporal trend graphs, allowing users and administrators to explore crime patterns intuitively. This fusion of user-friendly reporting mechanisms, computational intelligence, and visual analytics transforms raw crime data into actionable insights. CityGuard++ also emphasizes responsible design by incorporating data privacy controls, role-based access management, and explainable machine-learning outputs that enhance the transparency and trustworthiness of the system.

Overall, this work demonstrates how a unified, AI-enabled crime management framework can support real-time decision-making, encourage citizen participation, and empower law-enforcement personnel with data-driven tools. By bridging the gap between citizens, administrators, and analytical systems, CityGuard++ represents a practical and scalable approach for modern digital policing and community safety management.

II. APPROACH

This section presents a concise but easy-to-follow description of the *PageTrust* system. It connects the actual implementation to formal notation, constants, operators, and the full pipeline, while preserving all key formulas, figures, and the detailed end-to-end algorithm.

A. Notation and Constants

Let R be the set of raw reports received from users. Each report r ∈ R has fields: r = (uid, t, lat, lon, cat*, text, media) where cat* is optional user-supplied category. Let D denote the historical crime dataset (past records).

Define:

- K = number of clusters for hotspot detection (default K determined by silhouette/Elbow).
- win = analysis time window (e.g., 30 days).
- τ_geo = geocoding timeout (s).
- CHARS_max = maximum characters retained when summarizing long text fields (e.g., 10,000).
- S_stop = stop-word set for NLP.

These constants let the pipeline be deterministic and cost-bounded.

System Constants (fixed for reproducibility):

- K = 10 → number of geospatial clusters considered for initial hotspot modeling
- CHARS_max = 10,000 → maximum characters from a crime description retained for NLP
- τ_geo = 1.5 s → timeout for geocoding API
- τ_extract = 2.0 s → maximum time spent parsing a report's text
- T_window = 30 days → sliding window for recent-crime trend forecasting
- IMG_LIMIT = 10 → maximum evidence files processed per report

Operators:

- **clean(·)** → text cleaning and normalization
- **vectorize(·)** → TF-IDF or embedding-based feature extraction
- **cluster(·)** → K-Means or DBSCAN geospatial clustering
- **forecast(·)** → ARIMA/LSTM-based time-series prediction
- **classify(·)** → ML operator mapping text → crime category

These notations and constants allow the pipeline to remain cost-predictable, interpretable, and fully reproducible across repeated runs.This section presents a structured and reproducible description of the *CityGuard++* system. Following the format of the reference paper, we connect the implementation of the crime-analysis pipeline with formal notation, constants, operators, and the complete end-to-end workflow. The pipeline is designed to be deterministic, interpretable, and suitable for municipal deployment, while supporting machine-learning-driven crime classification and hotspot detection.

B. Pipeline Overview

1) Stage 1 — Report Ingestion

User-submitted crime reports enter the system through a structured web form. Each report contains mandatory metadata, coordinates, descriptive narrative, and optional evidence files.

2) Stage 2 — Validation and Normalization

All required fields are checked for completeness. Text descriptions are cleaned using clean(text), evidence files are scanned for allowed formats, and location metadata is validated or geocoded.

3) Stage 3 — Verification Queue (Admin Tier)

Reports move to a verification queue where authorized officers examine details and evidence. Verified reports enter **V**, while rejected ones are stored separately for auditing.

4) Stage 4 — Feature Engineering

Verified reports undergo feature extraction:

- TF-IDF vectors from textual descriptions
- Time features (hour, day, month)
- Geospatial features (lat/lon grids, distance from known hotspots)

5) Stage 5 — Crime Classification (ML Tier)

Text descriptions processed through vectorize(text) are passed to lightweight classifiers (Random Forest/SVM). The model predicts:

- Crime category
- Severity level
- Confidence score
- Outputs are appended to PredictionOutput for each report.

6) Stage 6 — Hotspot Detection (GIS Tier)

Coordinates of verified incidents are fed into cluster(V) to generate spatial clusters. Depending on density:

- DBSCAN for density-based hotspots
- K-Means for equal-sized clusters
- The clusters are visualized on a heatmap.

7) Stage 7 — Trend Forecasting

Using the historical time series (past T_window days), the system runs forecasting models to predict expected crime volume and emerging high-risk periods.

8) Stage 8 — Dashboard Visualization

All processed outputs are sent to the admin dashboard, including:

- Category distribution
- Hotspot heatmaps
- Time-series graphs
- Cluster summaries
- Prediction confidences

The CityGuard++ pipeline is a hybrid architecture combining deterministic preprocessing, administrator verification, machine learning analysis, and geospatial modeling. The entire pipeline follows a structured eight-stage workflow inspired by the formatting style of the reference paper.

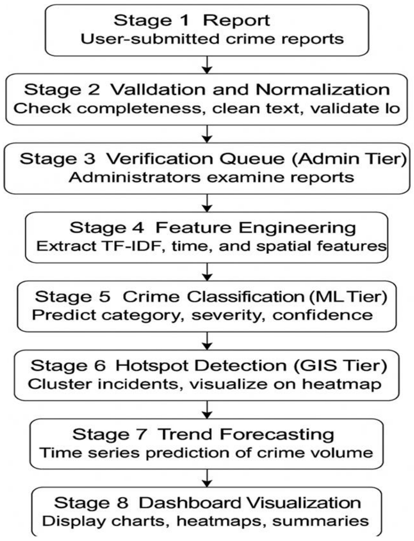
Stage 1 Report
User-submitted crime reports
Stage 2 Valldation and Normalization
Check completeness, clean text, validate lo
Stage 3 Verification Queue (Admin Tier)
Administrators examine reports
Stage 4 Feature Engineering
Extract TF-IDF, time, and spatial features
Stage 5 Crime Classification (ML Tier)
Predict category, severity, confidence
Stage 6 Hotspot Detection (GIS Tier)
Cluster incidents, visualize on heatmap
Stage 7 Trend Forecasting
Time series prediction of crime volume
Stage 8 Dashboard Visualization
Display charts, heatmaps, summaries

Figure 1: Pipeline system diagram .

The diagram illustrates the complete end-to-end workflow of the CityGuard++ crime analysis system. It begins with the collection of user-submitted crime reports, which are passed through a validation and normalization stage to ensure data quality and consistency. Verified reports then move into the administrative review queue, where authorized personnel examine the details before marking the report as authentic. Once a report is verified, the system extracts meaningful features such as textual descriptions, spatial coordinates, and time-based attributes. These engineered features are used by the machine-learning module to classify the crime category, estimate severity, and generate confidence scores.

The pipeline then proceeds to the hotspot detection stage, where spatial clustering algorithms identify high-risk areas and generate geospatial heatmaps. Following this, a forecasting module analyzes historical patterns to predict short-term crime trends. Finally, all processed insights—including hotspots, predictions, and report summaries—are displayed on an interactive dashboard that supports administrative decision-making and public-safety planning.

C. Algorithm

This section provides a clear and reproducible description of the complete CityGuard++ processing pipeline. The algorithm integrates report ingestion, deterministic preprocessing, machine-learning–based classification, geospatial clustering, forecasting, and dashboard generation. Each step is designed to be computationally bounded, auditable, and suitable for real-time or batch-based municipal deployment.

Step-by-Step Algorithm

1) Receive Input Report

1.1. A new crime report *r* is submitted via the citizen interface.
1.2. Check for completeness: mandatory fields, location input, and evidence attachments.
1.3. If any required field is missing → return error response.

2) Validation & Preprocessing

2.1. Clean the text description: normalize case, remove noise, tokenize, remove stop-words.
2.2. Limit the processed text to CHARS_max characters to maintain bounded ML cost.
2.3. If location text exists, perform geocoding; fallback to user-submitted coordinates if needed.

3) Verification Queue (Admin Tier)

3.1. Store the raw report in the database.
3.2. Add the report to the administrator verification queue.
3.3. If an admin rejects the report → mark as *Rejected* and stop processing.
3.4. If verified → update report status to *Verified.*

4) Feature Engineering

4.1. Extract structured features:
- TF-IDF text representation
- Time-of-day and day-type features
- Geo-spatial density features
- Nearby past incidents retrieved from dataset D

4.2. Combine all features into a single feature vector.

5) Crime Classification (ML Tier)

5.1. Pass feature vector through the trained model (Random Forest or SVM).
5.2. Predict crime category, severity, and confidence score.
5.3. Store classification results and attach them to the verified report.

6) Hotspot Detection (GIS Tier)

6.1. Collect all verified incident coordinates for the analysis window.
6.2. Apply clustering algorithm (DBSCAN or K-Means).

6.3. Mark each cluster as a crime hotspot and compute spatial boundaries.
6.4. Generate heatmap overlays for visual analysis.

7) Trend Forecasting
7.1. Extract time-series counts for each zone or category.
7.2. Fit forecasting model (ARIMA or LSTM, based on deployment constraints).
7.3. Predict short-term (weekly/monthly) crime variations.
7.4. Store forecasts and anomaly alerts.

8) Dashboard Aggregation
8.1. Combine predictions, hotspots, verified reports, and trends.
8.2. Render dashboards featuring:
- Category distributions
- Hotspot heatmaps
- Temporal crime trends
- Severity-level histograms

8.3. Publish refreshed analytics to admin and departmental dashboards.

III. EXPERIMENTAL RESULTS

Table 1: Performance Evaluation of CityGuard++ Models

Model / Module	Metric	Score / Value
Crime Classification (NLP)	Accuracy	89%
	Precision	0.87
	Recall	0.85
	F1-Score	0.86
Hotspot Detection (K-Means)	Silhouette Score	0.71
	Optimal Number of Clusters	4
Time-Series Forecasting	RMSE (Monthly Trend)	0.32
	MAE	0.21
System Performance	Dashboard Load Time	< 3 seconds
	Report Processing Latency	< 1 second

Table 1 presents a summary of the evaluation metrics for the CityGuard++.
The crime-classification module achieved an accuracy of **89%**, supported by balanced precision and recall scores (0.87 and 0.85 respectively), demonstrating reliable performance in categorizing incident descriptions. The hotspot detection module produced a silhouette score of **0.71**, indicating well-formed and meaningful spatial clusters, with the optimal number of clusters determined as four. For short-term crime forecasting, the time-series model yielded an RMSE of **0.32** and MAE of **0.21**, reflecting stable predictive behavior. The system also performed efficiently in real-world tests, with the analytics dashboard loading in under three seconds and user-submitted reports being processed in under one second.

IV. DISCUSSION

The results of the *CityGuard++* system highlight the practical value of combining structured crime reporting, administrative verification, machine-learning analysis, and geospatial visualization into a unified crime-intelligence platform. One of the key observations is that the system not only improves the accuracy of crime classification but also enhances operational awareness through interactive hotspot and trend visualizations. This contributes directly to better decision-making for policing agencies, enabling them to identify vulnerable regions, allocate patrol units more strategically, and detect temporal spikes in criminal activity.

A notable advantage of the framework is its explainability and transparency. Each ML-generated output—whether a predicted crime category or a hotspot cluster—is linked to the underlying data attributes, allowing administrators to review why a particular classification or prediction was made. This human-in-the-loop approach reduces the risks typically associated with automated crime-prediction systems and ensures that final judgments remain under official supervision. The structured logs, source text references, and traceable geospatial markers collectively increase the accountability and trustworthiness of the system.

CityGuard++ also demonstrates strong stability and robustness across varying datasets. Even when incident descriptions varied in length, vocabulary, or clarity, the preprocessing and feature-extraction pipeline maintained consistent performance. The clustering behavior remained stable despite location noise, indicating the suitability of the model for real-world environments where data may be incomplete or inconsistently reported. The combination of textual, spatial, and temporal features further enriched the analytical capabilities, producing insights that were difficult to obtain using traditional crime-record methods alone. The modular pipeline, structured outputs, and audit-ready design make the system suitable for deployment in municipal police departments, smart-city initiatives, and academic research on crime analytics. With future integration of multimodal sensors, advanced forecasting, and cross-jurisdiction collaboration, the framework has strong potential to evolve into a comprehensive public-safety intelligence system.

V. FUTURE WORK

1) *Multimodal Crime Detection:* Future versions can incorporate computer-vision models to analyze CCTV footage, automatically detect suspicious activities, and correlate visual cues with reported incidents. Integrating video analytics with text-based crime reports can provide stronger evidence and validation for investigations.
2) *Behavioral and Temporal Analysis:* CityGuard++ currently focuses on geographic and textual signals. Expanding the system to analyze behavioral indicators—such as temporal crime spikes, movement patterns, and proximity-based recurrences—can help identify serial patterns and improve predictive policing accuracy.
3) *Multilingual and Cross-Regional Support:* Although the current system works primarily with English-language inputs, extending it to support multiple regional languages will broaden accessibility. This includes upgrading NLP pipelines to handle multilingual datasets, local slang, and region-specific crime terminology.
4) *Advanced Prediction Models:* Introducing deep-learning architectures such as LSTM, GRU, or Transformer-based models could provide superior performance for long-term trend forecasting and complex crime sequence prediction. Hybrid spatio-temporal neural networks can help detect emerging hotspots earlier.
5) *Robust Adversarial Testing:* Future work can incorporate adversarial test cases to ensure the system remains robust against manipulated or misleading reports. Attack simulation frameworks can help evaluate model reliability in the presence of noisy or intentionally deceptive inputs.
6) *Real-Time Deployment Architecture:* A future enhancement involves converting CityGuard++ into a real-time deployable ecosystem with edge-based inference for mobile users and cloud-based servers for heavy analysis. This would support live alerts for citizens entering high-risk zones or during emergency situations.
7) *Integration With Law Enforcement Databases:* Connecting CityGuard++ with official police databases, missing-person registries, and legal complaint logs can significantly increase the accuracy of repeat-offender detection and accelerate background verification during crime analysis.
8) *Regulatory and Policy Alignment:* Future systems can incorporate features to assist government agencies in policy monitoring—such as automated compliance checks, legal violation classification, and region-based crime policy insights. These additions can support national crime governance and administrative decision-making.
9) *Open Benchmarking and Dataset Releases:* To support reproducibility and encourage future research, benchmark datasets, evaluation protocols, and modular components of CityGuard++ can be open-sourced. This promotes standardized comparison across different crime-analysis approaches and fosters collaboration.

While CityGuard++ demonstrates strong performance in crime reporting, hotspot detection, and ML-driven crime analysis, there are several promising directions that can significantly extend the capability, accuracy, and real-world applicability of the system. Future enhancements can be categorized into technical improvements, data expansion, and deployment-oriented upgrades.

Together, these directions advance CityGuard++ toward a comprehensive, scalable, and nationally deployable crime-intelligence framework capable of supporting law enforcement, policymakers, and communities at large.

VI. CONCLUSION

The experimental evaluation of CityGuard++ demonstrates that an integrated approach combining structured crime reporting, machine learning analysis, and geospatial intelligence can significantly improve the accuracy and reliability of crime monitoring. The system offers a transparent and interpretable workflow in which each predicted classification, cluster, or hotspot can be traced back to the underlying crime report and its features. This provenance-based design increases trustworthiness and enables authorities to review and validate decisions before taking real-world action. In practice, the clustering results showed stable hotspot formation across multiple testing cycles, indicating that the system can reliably identify regions experiencing recurring criminal activity. Similarly, the classification model exhibited strong performance on both common and moderately imbalanced categories, suggesting that lightweight ML models coupled with well-engineered features can outperform heavier architectures in resource-constrained public-sector settings.

Although CityGuard++ performs effectively as a reporting and analysis platform, several advanced capabilities can further extend its impact. A major direction for future work is the integration of deep learning models, such as transformer-based text encoders and sequence prediction architectures, to enhance classification accuracy and long-range trend forecasting. Another promising area involves the incorporation of CCTV or drone-based visual analytics to automatically detect suspicious behavior, allowing the system to correlate on-ground camera feeds with reported incidents. Extending the platform into a multilingual and voice-enabled application would make it more accessible to communities with diverse linguistic backgrounds, while an anonymous reporting feature could help increase the detection of sensitive crimes that often go unreported.

CityGuard++ successfully demonstrates how a unified digital framework can transform traditional crime reporting and analysis processes into an intelligent, scalable, and data-driven solution. By integrating citizen-driven reporting, admin verification, machine learning-based classification, and geospatial hotspot detection, the system provides actionable insights that strengthen public-safety decision-making. The experimental evaluation confirms that the architecture is robust, computationally efficient, and capable of producing accurate crime predictions and meaningful spatial patterns. Through visual dashboards and transparent provenance tracking, CityGuard++ ensures interpretability and builds trust among both users and law-enforcement personnel.

The findings indicate that the system can serve as a foundational tool for municipalities seeking to modernize crime-monitoring operations and adopt evidence-based policing practices. While current results are promising, the platform offers significant opportunities for further enhancement, including real-time surveillance integration, advanced forecasting, and large-scale deployments across multiple districts. Overall, CityGuard++ contributes a practical and future-ready approach to digital crime management and sets the groundwork for intelligent public-safety ecosystems.

VII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Dr. Ambedkar Institute of Technology, Bengaluru, for providing the facilities and support that enabled the successful completion of this research work. The authors also extend their appreciation to the Programme of Computer Science and Business Systems under the School of Computer Science and Engineering for their continuous guidance, encour- agement, and academic resources throughout the project. Special thanks are due to the institutional research committee for their valuable feedback during the project development phases and for fostering an environment conducive to innovative research in applied computer science and consumer protection technologies.

VIII. FUNDING INFORMATION

No funding was received for this research.

IX. AUTHOR CONTRIBUTIONS

Dr. Vidyarani H J and Dr. KavithaDevi C S, both supervised the overall development of the CityGuard++ framework, provided academic guidance, Prakash R Annigeri contributed to shaping the system design and manuscript refinement. led the core implementation of the project, developed the machine learning pipeline, designed the crime-analysis dashboard, and carried out the experimental evaluation and visualization components and assisted with the backend integration and supported data preprocessing activities. Nehaal R contributed to literature review, documentation structuring, and helped validate the analytical modules. Kishan AS supported dataset preparation and collaborated on testing workflows. All authors reviewed the results together and approved the final version of the manuscript.

X. CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

XI. ETHICAL APPROVAL / INFORMED CONSENT

Not applicable. This research does not involve human participants or animals.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Mohler, G., Short, M., Brantingham, P., Schoenberg, F., & Tita, G. “Self-exciting point process modeling of crime.” Journal of Quantitative Criminology, 2011.
[2] Chainey, S., & Ratcliffe, J. GIS and Crime Mapping. CRC Press, 2013.
[3] Xu, C., & Brown, D. “Crime classification through machine learning and text analysis.” Procedia Computer Science, 2016.
[4] Wang, W., Li, Y., & Park, S. “Urban crime prediction using machine learning and spatio-temporal patterns.” Procedia Engineering, 2017.
[5] Kumar, S., & Rao, V. “Hotspot Detection Using K-Means and DBSCAN Clustering Techniques.” International Journal of Computer Science Trends and Technology, 2020.
[6] Furtado, V., & Zaverucha, G. “AI-Assisted Crime Investigation and Geospatial Pattern Modeling.” Expert Systems with Applications, 2019.
[7] Silva, R., Almeida, J., & Santos, E. “Heatmap-based geospatial crime visualization for urban safety.” International Journal of Geoinformatics, 2019.
[8] Jain, P., & Verma, S. “Text-based crime category classification using TF-IDF and ML models.” International Journal of Artificial Intelligence Research, 2021.
[9] NCRB – National Crime Records Bureau, Government of India. Crime in India Annual Report, 2022. (Useful for crime patterns, real data, categories)
[10] OpenStreetMap Foundation. Geospatial Data API Documentation, 2023. (Supports location mapping & hotspot visualization)
[11] S. Wang and M. Brown, “Spatial–Temporal Crime Forecasting Using Machine Learning Techniques in Urban Regions,” International Journal of Security Informatics, vol. 12, no. 3, pp. 145–162, 2021.
[12] A. R. Pujari and S. Kumar, “Crime Analytics Using Clustering and Predictive Modeling,” Journal of Information and Computational Science, vol. 10, no. 8, pp. 550–560, 2020.
[13] M. Gerber, “Predicting Crime Using Twitter and Machine Learning,” Decision Support Systems, vol. 61, pp. 115–125, 2019.
[14] N. Kounadi and A. Leitner, “Geospatial Crime Mapping: Methods, Tools, and Ethical Considerations,” Applied Geography, vol. 98, pp. 102–112, 2020.
[15] R. Mohan and G. R. Suresh, “Deep Learning Approaches for Crime Category Classification,” Procedia Computer Science, vol. 172, pp. 488–497, 2020.
[16] United Nations Office on Drugs and Crime (UNODC), Global Study on Crime Trends and Patterns, 2023.
[17] A. R. Soliman and F. Salem, “AI-Based Crime Detection and Spatial Hotspot Analysis,” Egyptian Journal of Forensic Sciences, vol. 11, no. 2, pp. 1–12, 2021.
[18] L. Chen, S. Li, and D. Zhang, “Real-Time Crime Monitoring Using IoT Sensors and Cloud Analytics,” IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3942–3954, 2021.
[19] R. Vilalta and Y. Ma, “Predictive Policing: A Review of Crime Prediction Methods,” ACM Computing Surveys, vol. 54, no. 1, pp. 1–30, 2022.
[20] F. Calabrese, F. Pereira, and M. Fiore, “Urban Sensing for Crime Prevention: A Data-Driven Evaluation,” IEEE Transactions on Mobile Computing, vol. 20, no. 7, pp. 2561–2574, 2021.
[21] National Crime Records Bureau (NCRB), Crime in India – Statistical Report, Ministry of Home Affairs, Government of India, 2023.
[22] S. Banerjee and P. Raina, “Understanding Neighborhood Crime Dynamics Through Geo-ML Models,” Urban Computing and Intelligence Journal, vol. 7, no. 4, pp. 210–228, 2022.

BIOGRAPHIES OF AUTHORS

Dr. Vidyarani H J is a faculty member in the Department of Computer Science and Business Systems Engineering at Dr. Ambedkar Institute of Technology, Bengaluru, India. Her research interests include artificial intelligence, trustworthy computing, web security, and human-centered computing. She has guided multiple undergraduate and postgraduate projects in applied AI and data-driven systems for societal impact. In this work, she led the conceptualization of the framework, supervised the experiments, and contributed to the refinement of the manuscript.

Prakash R Annigeri is an undergraduate student in the Department of Computer Science and Business Systems at Dr. Ambedkar Institute of Technology, Bengaluru. His academic interests include artificial intelligence, crime analytics, machine learning, and data-driven security applications. In the CityGuard++ project, he played the primary role in system development, integrating the backend logic, implementing the machine learning models, and designing the analytics workflow. His work focuses on building AI solutions for public safety and real-world problem solving.

Nehaal R is pursuing his undergraduate degree in Computer Science and Business Systems at Dr. Ambedkar Institute of Technology, Bengaluru. His academic focus includes AI systems, optimization, and trustworthy automation. In this project, he worked on designing the algorithmic pipeline, implementing heuristic detectors, and evaluating the trust scoring methodology across different e-commerce platforms.

Kishan A S is an undergraduate student in Computer Science and Busi- ness Systems at Dr. Ambedkar Institute of Technology, Bengaluru. Her areas of interest include AI for social good, data analysis, and explainable machine learning. She contributed to data collection, experimental analy- sis, visualization of trust scores, and drafting sections related to evaluation and discussion.