



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Crime Hotspot Prediction, Risk-Aware Route Planning and Patrol Planning Using GIS and Machine Learning

Assistant Professor Smita Mangesh Junnarkar¹, Vineeth Ravichandran²

Department of DataScience, SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS), Sion(W), Mumbai.

¹Smita Mangesh Junnarkar, smitaj@sies.edu.in

Department of DataScience, SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS), Sion(W), Mumbai.

²[Vineethravichandran@gmail.com]

ABSTRACT :

Crime is not spread evenly across a city. A small number of streets, junctions and neighbourhoods account for a disproportionate share of incidents, and this regularity gives police departments a practical reason to look for forecasting tools. This paper presents an integrated framework that turns historical crime records into three connected outputs: a map of predicted crime probability, a set of risk-aware travel routes, and a patrol plan for available units. The study area is divided into a regular grid, and each cell is described by temporal, spatial, and contextual features. Spatial clustering and local statistics identify existing hotspots, while supervised learning models (Logistic Regression, Random Forest and Gradient Boosting) estimate the probability of at least one incident in a cell during the next time window. The predicted probabilities are then attached to the edges of a road network so that a shortest-path search can trade a small increase in travel distance for a lower exposure to risk. A greedy coverage procedure uses the same probabilities to allocate patrol units. The complete pipeline is implemented in Python and delivered through an interactive Streamlit dashboard. We describe the design, the evaluation protocol, and the ethical safeguards required when predictive tools are used in policing.

Keywords: *crime hotspot prediction, geographic information systems, machine learning, risk-aware routing, patrol planning, spatial analysis, predictive policing*

1. Introduction

Urban crime places a steady burden on residents, businesses and the agencies that serve them. Police departments rarely have enough personnel to watch every location at every hour, so decisions about where to send officers are, in practice, decisions about which risks to accept. For many years these decisions rested on experience and on simple counts of past incidents. The growth of digital record keeping, together with open-source geographic tools and accessible machine learning libraries, now makes it possible to support such decisions with systematic spatial and statistical analysis.

The empirical basis for place-focused analysis is strong. Sherman, Gartin and Buerger (1989) showed that predatory crime in Minneapolis clustered at a small number of addresses, and later work by Weisburd (2015) reported that about half of the crime in many cities occurs on roughly four to five per cent of street segments. The routine activity approach of Cohen and Felson (1979) offers a theoretical explanation: crime tends to appear where motivated offenders, suitable targets and weak guardianship coincide in time and space. If that convergence repeats at the same places, then the past contains information about the future.

Most published work stops at the prediction step. A map of probable hotspots is useful, but it does not tell a patrol supervisor how to distribute vehicles across a shift, and it does not tell a commuter or a field worker which of several roads is the safer choice at ten at night. This paper treats prediction as the first stage of a larger decision pipeline. The same probability surface is reused to plan routes that avoid high-risk segments and to plan patrols that cover as much expected risk as the available resources allow.

1.1 Contributions

The main contributions of this work are the following:

- A reproducible workflow that converts raw, geocoded crime records into a space-time grid of features and labels suitable for supervised learning.
- A comparison of three standard classifiers for next-period hotspot probability, evaluated with a time-ordered split and with metrics that remain meaningful when hotspot cells are rare.
- A risk-aware routing method that adds a tunable risk penalty to road-network edge costs and exposes the trade-off between distance and safety.
- A coverage-based patrol allocation procedure that uses predicted probabilities and a minimum spacing rule between units.
- A Windows-friendly implementation with an interactive Streamlit interface, together with a discussion of fairness, privacy and feedback-loop risks.

1.2 Organisation of the Paper

Section 2 reviews related work. Section 3 states the problem and objectives. Section 4 describes the data and its preparation. Section 5 sets out the methodology, including hotspot detection, prediction, routing and patrol planning. Section 6 explains the experimental design and Section 7 presents and discusses the results. Section 8 considers ethical issues, Section 9 lists limitations, and Section 10 concludes with directions for future work.

2. Related Work

2.1 Hotspot Mapping and Spatial Statistics

Early computational work on crime concentration relied on mapping. Kernel density estimation, introduced in a general statistical setting by Silverman (1986), became a common way to smooth point data into a continuous surface. Chainey, Thompson and Uhlig (2008) compared several mapping techniques and concluded that kernel density estimation was among the most reliable for predicting where crime would recur. Ratcliffe (2004) proposed the hotspot matrix, which links the spatial and temporal persistence of clusters to different operational responses.

Statistics that test for spatial association provide a more formal footing. The Getis-Ord statistic (Getis and Ord, 1992) detects locations where high values are surrounded by other high values, and local indicators of spatial association (Anselin, 1995) separate genuine clusters from spatial outliers. Density-based clustering, in particular DBSCAN (Ester et al., 1996), groups incident points without requiring the number of clusters to be fixed in advance, which suits crime data because the number of hotspots is not known beforehand.

2.2 Statistical and Machine Learning Models

Point-process models treat crime as a self-exciting process in which one event raises the short-term likelihood of nearby events. Mohler et al. (2011) adapted a model originally developed for earthquake aftershocks to burglary data and reported gains over conventional hotspot maps. In parallel, supervised learning has been applied to area-level prediction. Ensembles of decision trees, such as Random Forests (Breiman, 2001) and gradient boosting systems such as XGBoost (Chen and Guestrin, 2016), handle mixed numerical features and non-linear interactions well and are widely used on tabular crime data. Recurrent neural networks, including long short-term memory units (Hochreiter and Schmidhuber, 1997), have been tested for sequences of counts, although their advantage on short and noisy series is not consistent.

2.3 Routing and Patrol Allocation

Route planning on road networks is usually built on the shortest-path algorithm of Dijkstra (1959). Network data from OpenStreetMap can be downloaded, cleaned and analysed with the OSMnx library (Boeing, 2017), which has made network-based studies far easier to reproduce. Safety-aware routing replaces the pure distance objective with a combined cost that includes a risk term, so that the chosen path is not necessarily the shortest one. Patrol allocation, for its part, can be framed as a maximum coverage problem, for which greedy heuristics offer simple and fast solutions.

2.4 Criticism and Responsible Use

Predictive policing has been criticised on several grounds. Lum and Isaac (2016) demonstrated that models trained on police records learn where police have been, not necessarily where crime occurs, and may therefore reinforce existing patterns of enforcement. Ensign et al. (2018) analysed the feedback loop formally and showed that sending officers to predicted areas can inflate the data from those areas and distort later predictions. These findings shape the design decisions and the cautions discussed later in this paper.

3. Problem Statement and Objectives

Given a set of historical crime incidents, each with a location, a timestamp and a category, and given a road network of the same area, the aim is to support three decisions: where crime is likely to occur in the next period, which route between two points carries the lowest acceptable risk, and where a limited number of patrol units should be placed. The objectives are listed below.

- To detect and visualise existing crime hotspots using spatial clustering and local statistics.
- To predict the probability of crime in each grid cell for the next time window using machine learning.
- To compute routes whose cost combines distance with predicted risk.
- To generate patrol allocations that maximise covered risk under resource and spacing constraints.
- To package the full pipeline as an interactive tool and to evaluate it with appropriate and honest metrics.

4. Data and Preprocessing

4.1 Data Sources

The framework needs two types of input. The first is a table of crime incidents, and the second is the road network. For this study the incident records come from [name and source of your dataset, with years covered and number of records]. The road network is obtained from OpenStreetMap for the same study area [name of city or region]. Table 1 lists the attributes used from the incident data.

Table 1. Attributes used from the incident records

Attribute	Type	Use in the pipeline
Latitude, Longitude	Numeric	Locating the incident and assigning it to a grid cell
Date and time	Timestamp	Building time windows, hour, weekday and season features

Attribute	Type	Use in the pipeline
Crime category	Categorical	Filtering by offence type and computing category shares
Locality or beat	Categorical	Optional aggregation and result interpretation

4.2 Cleaning

Raw police data nearly always needs attention before analysis. Records with missing or implausible coordinates, such as points that fall outside the study boundary, were removed. Duplicate entries were identified by matching location, time and category. Timestamps were converted to a single time zone and parsed into date components. Offence labels were standardised so that spelling variants of the same category were merged. The number of rows dropped at each step is reported in the results so that the effect of cleaning can be audited.

4.3 Spatial Gridding and Time Windows

The study area was divided into square cells of [cell size, e.g., 500 m] on a side using a projected coordinate system, so that distances and areas are measured in metres rather than degrees. Each incident was assigned to the cell that contains it. Time was divided into windows of [window length, e.g., one day or one week]. For every cell and window a record was created, even when no crime occurred, because the absence of crime is an essential part of the training signal. The binary target is defined as:

$$y(c, t+1) = 1 \text{ if } N(c, t+1) \geq 1, \text{ otherwise } 0 \quad (1)$$

where $N(c, t)$ is the number of incidents in cell c during window t . Cells that lie outside the city boundary or have no road access were excluded.

4.4 Feature Engineering

Features were computed only from information available at or before window t , so that no future information leaks into the prediction of window $t+1$. Table 2 summarises the feature groups.

Table 2. Feature groups

Group	Examples
Temporal lags	Incident counts in the previous 1, 2, 4 and 8 windows; rolling mean and rolling maximum
Calendar	Month, weekday, hour band, weekend indicator, holiday indicator
Spatial neighbourhood	Mean count in the eight adjacent cells; distance to the nearest detected hotspot centre
Category mix	Share of property, violent and other offences in the cell over the past period
Contextual	Road density, number of intersections, distance to major roads [add any extra layers you used, such as points of interest]

4.5 Class Imbalance

In most cells and windows no crime is recorded, so positive examples are the minority. A model that predicts "no crime" everywhere would reach high accuracy and be of no use. To address this, class weights were applied during training, and the oversampling technique SMOTE (Chawla et al., 2002) was tested on the training portion only. Evaluation always used the original, untouched class distribution.

5. Methodology

The system has four stages: hotspot detection, probability prediction, risk-aware routing and patrol planning. The output of each stage is the input of the next. Figure 1 gives a description of the overall flow in words, and the subsections that follow explain each stage.

Figure 1 (description). Raw incident records and the road network are cleaned and projected, then aggregated to a space-time grid. Features are computed and passed to the trained model, which outputs a probability for each cell. The probability surface is then used in two ways: it weights the edges of the road graph for routing, and it feeds the coverage procedure for patrol placement. Results are displayed in the dashboard.

5.1 Hotspot Detection

Two complementary methods were used to describe where crime is concentrated. First, DBSCAN was run on incident coordinates with a neighbourhood radius of [eps value] metres and a minimum of [min points] points. Points that do not belong to any cluster are treated as noise, which prevents isolated incidents from being labelled as hotspots. Second, the Getis-Ord G_i^* statistic was calculated on the cell counts, and cells with a z-score above 1.96 (95 per cent confidence) were marked as statistically significant hot cells. Agreement between the two methods was examined, and the G_i^* output was also kept as a feature for the predictive model.

5.2 Crime Probability Prediction

Three classifiers were selected so that the comparison covers a transparent baseline and two stronger ensemble methods.

- Logistic Regression serves as an interpretable baseline. Its coefficients show the direction and size of each feature's effect after scaling.
- Random Forest (Breiman, 2001) averages many decision trees trained on bootstrap samples. It copes well with non-linear effects and gives feature importance scores.
- Gradient Boosting builds trees sequentially, each one correcting the errors of the previous ensemble. Implementations such as XGBoost (Chen and Guestrin, 2016) are efficient and often strong on tabular data.

All models were implemented with scikit-learn (Pedregosa et al., 2011) and its compatible libraries. Hyperparameters were tuned by time-series cross-validation on the training period only, using the area under the precision-recall curve as the selection criterion. Predicted probabilities were examined with a calibration curve, because the routing and patrol stages use the probabilities as numbers and not just as a ranking. Where calibration was weak, isotonic or Platt scaling was applied on a validation block.

5.3 Risk-Aware Route Planning

The road network is represented as a directed graph $G = (V, E)$, where nodes are intersections and edges are road segments. Each edge e has a length $d(e)$ in metres. The predicted probability of the cell or cells that the edge crosses is converted into an edge risk $r(e)$ between 0 and 1, using a length-weighted average when an edge spans more than one cell. The cost of travelling along an edge is then defined as:

$$w(e) = d(e) \times (1 + \lambda \cdot r(e)) \quad (2)$$

The parameter $\lambda \geq 0$ controls how strongly risk is penalised. When λ is zero the cost is simply distance, and the search returns the ordinary shortest path. Larger values of λ make the algorithm prefer longer roads that pass through safer areas. The minimum-cost route between an origin and a destination is found with Dijkstra's algorithm (Dijkstra, 1959) on the weighted graph. For every origin-destination pair the framework reports the shortest route and the risk-aware route side by side, and it computes the extra distance and the reduction in cumulative risk, defined as:

$$R(\text{path}) = \sum d(e) \cdot r(e), \text{ for all } e \text{ in path} \quad (3)$$

Reporting both numbers makes the trade-off visible. A route that is only a few per cent longer but lowers exposure noticeably is a good recommendation, whereas a route that doubles the distance for a small safety gain is not. The value of λ can be chosen by the user in the interface, and a sensitivity analysis over a range of values is included in the evaluation.

5.4 Patrol Planning

Patrol planning is posed as a coverage problem. Let K be the number of units available in a shift, and let $p(c)$ be the predicted probability for cell c in that shift. Each unit covers its own cell and, optionally, the cells within a radius ρ . The objective is to choose K positions that maximise the total covered probability:

$$\text{maximise } \sum p(c) \text{ over all cells } c \text{ covered by the chosen } K \text{ positions} \quad (4)$$

Because exact maximum coverage is computationally hard, a greedy procedure is used. At each step it selects the position that adds the largest amount of not-yet-covered probability, then marks the covered cells and repeats until K positions have been chosen. Greedy selection is known to give a solution that is guaranteed to be within a constant factor of the optimum for this class of problem, and in practice it runs in a fraction of a second on city-scale grids. A minimum spacing rule prevents two units from being placed in the same neighbourhood. Patrol sequences between the chosen positions can be generated with the same routing engine described above.

5.5 System Implementation

The system is written in Python. Pandas and NumPy handle tabular processing, GeoPandas and Shapely handle geometry and spatial joins, OSMnx and NetworkX provide the road graph and shortest-path search, and scikit-learn supplies the models. The interface is built with Streamlit, which turns a script into a web application without separate front-end code. The user can select the time window, offence category, risk weight λ and number of patrol units, and the dashboard returns the probability map, the comparison of routes and the proposed patrol points. The environment was prepared to run on Windows using a standard virtual environment, and all dependencies are listed in a requirements file to make the work easy to reproduce.

6. Experimental Setup and Evaluation

6.1 Train and Test Split

Because the task is forecasting, a random split would let the model learn from future data and would overstate performance. The data were therefore divided in time order: the earliest [percentage, e.g., 70] per cent of windows were used for training, the next [percentage] per cent for validation, and the most recent windows for testing. Cross-validation inside the training set also used expanding time windows.

6.2 Prediction Metrics

Accuracy alone is misleading when positives are rare, so the following measures were used: precision, recall, F1-score, the area under the ROC curve, and the area under the precision-recall curve. The Brier score was added to assess the quality of the probabilities themselves. Because the practical use is to rank cells for attention, a ranking measure was also computed. This is the hit rate at the top k per cent of cells, which answers a direct operational question: if patrols are limited to the k per cent of cells with the highest predicted probability, what share of actual crimes falls inside them?

Predictive accuracy was compared with two simple baselines. The first predicts that next period's hotspots are the cells with the highest counts in the previous period. The second ranks cells by their long-run historical average. A model is worth deploying only if it beats these baselines by a clear margin.

6.3 Routing and Patrol Metrics

For routing, a sample of [number] random origin-destination pairs was drawn. For each pair the study measured the added distance, the percentage reduction in risk exposure, and the computation time. For patrol planning, the main measure was the share of actual test-period incidents that fell within the covered cells, compared with a random placement and with a placement based on previous counts only.

7. Results and Discussion

This section presents the structure in which the results are reported. The highlighted entries must be replaced with the values produced by the project code on the test period. No numbers have been assumed here, because results written without running the experiments would not be valid.

7.1 Hotspot Detection

DBSCAN identified [number] clusters, and the Getis-Ord analysis marked [number] cells as significant hot cells. The overlap between the two was [percentage] per cent. [Describe where the main hotspots are located, for example around transport hubs, markets or commercial streets, and whether they change by time of day.]

7.2 Prediction Performance

Table 3. Comparison of models on the test period

Model	Precision	Recall	F1	ROC-AUC	PR-AUC	Top-10% hit rate
Previous-period baseline	[]	[]	[]	[]	[]	[]
Historical average	[]	[]	[]	[]	[]	[]
Logistic Regression	[]	[]	[]	[]	[]	[]
Random Forest	[]	[]	[]	[]	[]	[]
Gradient Boosting	[]	[]	[]	[]	[]	[]

The discussion of Table 3 should answer four questions. Which model ranked highest, and was the margin over the baselines meaningful? Did the more complex models justify their extra complexity compared with Logistic Regression? How well calibrated were the probabilities? Which features carried the most importance, and do they agree with criminological expectations? In most studies of this type, recent lag features and neighbourhood activity dominate, which is consistent with the persistence of hotspots described by Weisburd (2015). Report whether the same holds here, and explain any surprise.

7.3 Routing Results

Table 4. Effect of the risk weight on route quality (mean over sampled pairs)

Risk weight (λ)	Extra distance (%)	Risk reduction (%)	Mean runtime (ms)
0 (shortest path)	0	0	[]
[value]	[]	[]	[]
[value]	[]	[]	[]
[value]	[]	[]	[]

As λ increases, the risk-aware route should deviate more from the shortest path, so extra distance and risk reduction are expected to rise together. The most informative finding is usually the point at which additional distance stops buying much additional safety. This knee in the curve gives a practical default for λ . Include an example map of one origin-destination pair, showing both routes over the probability surface.

7.4 Patrol Planning Results

Table 5. Share of test-period incidents covered by different patrol placements

Placement method	Units (K)	Incidents covered (%)
Random placement	[K]	[]
Highest previous-period counts	[K]	[]
Proposed greedy coverage using model probabilities	[K]	[]

The comparison in Table 5 shows whether model-based placement adds value beyond a simple rule based on recent counts. If the improvement is small, this should be stated openly, since an honest negative or modest result is more useful to practitioners than an inflated one.

7.5 Discussion

Taken together, the three outputs show how one probability surface can support several decisions. The prediction stage provides the ranking, the routing stage translates it into advice for individuals, and the patrol stage translates it into advice for the organisation. Because errors in the first stage pass on to the others, model calibration matters more here than in a purely descriptive study. A model that ranks cells well but gives inflated probabilities would still produce reasonable patrols, yet it would distort the balance between distance and risk in routing. For this reason the calibration analysis in Section 5.2 is treated as part of the method and not as an optional extra.

8. Ethical Considerations

Any system that guides police resources or advises citizens on safe travel carries social risk, and this work does not ignore it. Four concerns deserve explicit mention.

First, the data record reported and recorded crime, which is not the same thing as crime that occurs. Neighbourhoods that are patrolled more heavily generate more records, and models trained on those records may send still more patrols there (Lum and Isaac, 2016; Ensign et al., 2018). To reduce this danger, the framework recommends using offence types that are less dependent on police initiative, auditing predictions by area, and treating the output as one input among several rather than as an order.

Second, route advice can stigmatise places. Labelling a locality as unsafe may lower investment and affect the people who live there. The tool is therefore intended for academic demonstration, and any public deployment would need wording, thresholds and governance agreed with local stakeholders.

Third, privacy must be protected. The framework works with aggregated cell-level counts and does not need personal identifiers about victims or suspects. Location precision should be reduced where needed so that individuals cannot be re-identified.

Fourth, the predictions describe places and times, not people. The system is not designed to score individuals and should not be used for that purpose.

9. Limitations

Several limitations should be kept in mind when interpreting the results. The quality of the output depends on the completeness and accuracy of the incident records, which may be affected by under-reporting and by delays in registration. The grid size and window length are design choices that influence results, a problem known in geography as the modifiable areal unit problem. The models use historical patterns and cannot anticipate sudden changes, such as large events, policy shifts or new infrastructure. The routing model considers risk and distance but ignores traffic, lighting, road closures and personal preferences. Finally, the evaluation covers one study area and one period, and generalisation to other cities should be tested and not assumed.

10. Conclusion and Future Work

This paper described a complete pipeline that begins with historical crime records and ends with practical outputs for decision makers: a probability map, safer routes and patrol plans. By combining spatial clustering, supervised learning, graph-based routing and coverage optimisation in one Python application, the work shows how GIS and machine learning can move from description to action. The emphasis on time-ordered evaluation, calibration and comparison against simple baselines is intended to keep the claims modest and verifiable.

Several directions are open for future work. Spatio-temporal deep learning models, such as graph neural networks, could capture the dependencies between neighbouring cells more directly. Real-time feeds for traffic and events could make the routing component more realistic. Fairness-aware training and regular audits could address the feedback problem discussed above. Lastly, a field study with practitioners would show whether the tool changes decisions in a useful way, which is ultimately the most important test of any predictive system.

REFERENCES

- [1] Anselin, L. (1995). Local indicators of spatial association—LISA. *Geographical Analysis*, 27(2), 93–115.
- [2] Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65, 126–139.
- [3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- [4] Chainey, S., Tompson, L., and Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21(1–2), 4–28.
- [5] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- [6] Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
- [7] Cohen, L. E., and Felson, M. (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review*, 44(4), 588–608.
- [8] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.

-
- [9] Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C., and Venkatasubramanian, S. (2018). Runaway feedback loops in predictive policing. *Proceedings of Machine Learning Research*, 81, 160–171.
- [10] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 226–231.
- [11] Getis, A., and Ord, J. K. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis*, 24(3), 189–206.
- [12] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- [13] Lum, K., and Isaac, W. (2016). To predict and serve? *Significance*, 13(5), 14–19.
- [14] Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493), 100–108.
- [15] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [16] Ratcliffe, J. H. (2004). The hotspot matrix: A framework for the spatio-temporal targeting of crime reduction. *Police Practice and Research*, 5(1), 5–23.
- [17] Sherman, L. W., Gartin, P. R., and Buerger, M. E. (1989). Hot spots of predatory crime: Routine activities and the criminology of place. *Criminology*, 27(1), 27–56.
- [18] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.
- [19] Weisburd, D. (2015). The law of crime concentration and the criminology of place. *Criminology*, 53(2), 133–157.