Boldt *Crime Science* (2026) 15:7
https://doi.org/10.1186/s40163-026-00269-x

Crime Science

RESEARCH Open Access

A globally optimal algorithm for hotspot detection and ranking

Martin Boldt[1*]

Abstract

Objectives Crime prevention strategies often rely on the small set of micro-places where crime is most concentrated, the so-called hotspots, yet it has remained unclear how close existing hotspot detection methods come to the maximum coverage theoretically possible. This study introduces GraphVenn, the first algorithm that identifies the globally optimal placement of *N* fixed-radius hotspots directly from the empirical crime distribution, without relying on heuristic or approximate approaches.

Methods GraphVenn was evaluated on three years of crime data from Malmö, Boston, and New York City (in total 1.75 million crimes) and compared against kernel density estimation (KDE), greedy PAI maximization (PAI-Max), and GraphTrace. Both the globally optimal and the greedy (fast approximation) modes of GraphVenn were evaluated across different spatial resolutions, demonstrating scalability to large urban datasets.

Results In optimal mode, GraphVenn identified the absolute maximum coverage of incidents achievable under fixed-radius constraints. The greedy variant reached within 0.1–1.9% of this optimum while reducing runtimes by up to two orders of magnitude. By contrast, existing methods consistently fell short, e.g., in New York City the optimal GraphVenn captured 51,522 crimes within its top-100 hotspots compared to 35,098 with KDE and 28,241 with GraphTrace, while PAI-Max was excluded due to its runtimes. In practical terms, the baselines therefore missed between 16,000 and 23,000 crime incidents that could have been covered.

Conclusions Globally optimal detection of fixed-radius hotspots that maximize the distinct crime count is now computationally feasible at city scale. GraphVenn offers (i) a practical tool for researchers, law enforcement, and crime analysts to identify the most effective fixed-radius hotspot locations with confidence that no better configuration exists, and (ii) a benchmark for evaluating approximate methods against the true maximum crime count. Open-source code is provided to support replication and further research.

Keywords Global crime hotspot optimization, Graph-based crime analysis, Crime hotspot detection, Computational criminology

*Correspondence:
Martin Boldt
martin.boldt@bth.se
[1]Department of Computer Science, Blekinge Institute of Technology, Karlskrona, Blekinge, Sweden

Introduction

A well-established finding in criminology is that crime is highly concentrated in space, since a small number of micro-places account for a disproportionate share of incidents (Weisburd, 2015). This observation forms the basis for hotspot policing, resource allocation, and situational prevention strategies (Braga et al., 2019). A central challenge, however, is to determine exactly *which locations* should be selected as hotspots to maximize crime coverage, given practical constraints such as the number of hotspots that can be targeted and the spatial radius within which incidents are counted.

To date, researchers and crime analysts have relied on approximate approaches such as kernel density estimation (KDE), greedy optimization, or graph-based clustering (Martínez et al., 2023; Mohler et al., 2020; Rosenblatt, 1956). While useful in practice, these methods make simplifying assumptions, e.g., imposing artificial spatial grid cells that split dense concentrations of crime between adjacent cells, thereby underestimating their intensity. Moreover, they cannot guarantee that the selected hotspots are truly the best possible. In other words, we have not known how close our hotspot maps come to the maximum achievable coverage of crime incidents.

This paper introduces GraphVenn[1], an algorithm that, for the first time, enables crime analysts to detect the true global optimum for fixed-radius hotspots. Unlike grid-based or heuristic methods, GraphVenn adapts dynamically to the observed crime patterns by modeling the spatial relationships between incidents directly using graphs. In doing so, it can combine graph-theoretic representations with spatial optimization to identify the strongest concentrations of crime without reliance on arbitrary grids.

Given the number of hotspots to detect (N), a hotspot radius, and a spatial resolution, GraphVenn computes the configuration of N hotspots that covers the maximum possible number of distinct crime incidents within the specified radius and resolution. It operates in two modes. In *optimal mode*, it establishes the ground truth by guaranteeing maximum coverage under the chosen parameters. In *greedy mode*, it achieves near-identical results in a fraction of the time, making it practical for routine use by crime analysts and practitioners. Together, these capabilities highlight two key contributions:

1. *Practical value:* crime analysts can use GraphVenn to identify the most effective hotspot locations for interventions such as CCTV placement, confident that no better configuration exists under the chosen constraints.
2. *Scientific value:* For the first time, researchers can observe the actual upper bound of crime concentration given the fixed-radius hotspot constraint, providing a benchmark for evaluating approximate approaches.

The remainder of this paper presents the related work, then introduces GraphVenn, evaluates it on crime data from three cities, and discusses implications for crime mapping and prevention practice.

Related work

Research on why certain micro-places experience disproportionately high crime levels draws from several criminological theories. Opportunity-based perspectives argue that crime concentrates where offending opportunities are abundant, including routine activity (Cohen & Felson, 1979), rational choice (Cornish & Clarke, 2016), and crime pattern theory (Brantingham & Brantingham, 1995), all emphasizing the convergence of offenders, targets, and absent guardians.

Empirically, much work has focused on identifying and analyzing crime hotspots (Chainey, 2014). The "law of crime concentration" (Weisburd, 2015) shows that a small share of places account for most recorded crime, a finding validated across time scales (Haberman et al., 2017) and in diverse national contexts (Hardyns et al., 2019),(Stanković 2022).

Computational approaches to spatial hotspot detection

A broad set of computational methods has been developed for detecting high-crime locations. In criminology, theory-driven models such as risk Terrain modeling (RTM) (Caplan et al., 2011) incorporate environmental risk factors, while data-driven approaches rely more directly on historical point-pattern data. Kernel density estimation (KDE) remains the most widely applied technique vPorter12,Gorr15, and many variants incorporate temporal dynamics using sliding windows or spatio-temporal smoothing (Hu et al., 2018). Spatio-temporal point process models (Johnson et al., 2007; Piza & Carter, 2018) also capture repeat and near-repeat effects, while generalized linear and additive models (Kennedy et al., 2011; Wang et al., 2016, 2012), Random Forests (Berk et al., 2008), and deep learning methods (Stec & Klabjan, 2018; Wang et al., 2019) are increasingly used for hotspot prediction and offender risk assessment.

A substantial body of work in statistics approaches hotspot detection using *scan statistics*, beginning with Kulldorff's spatial scan statistic (Kulldorff, 1997). This framework has since been expanded in several directions, including irregular or variable-size scanning windows, extensions to non-circular shapes, and hypothesis-testing procedures that evaluate the statistical significance

[1] The source code for GraphVenn is publicly available on GitHub: https://github.com/boldten/GraphVenn/.

of detected clusters (Abolhassani & Prates, 2021; Patil & Taillie, 2004). Other related contributions include the support vector subset scan for anomalous cluster detection (Fitzpatrick et al., 2021), and learning-to-rank frameworks for prioritizing high-risk locations (Mohler et al., 2020).

Within applied criminology, the U.S. Department of Justice's *CrimeStat* software implements many of these techniques, including KDE, spatial scan statistics, nearest-neighbor clustering, and flexible cluster shapes (National Institute of Justice, 2019). These approaches typically aim to identify a single (or small number of) statistically significant clusters of elevated intensity for analytic or investigative purposes. In contrast, they are generally not designed to optimize the *joint* selection of a larger set of disjoint hotspots of fixed radius.

Graph-based spatial analysis has also gained traction for modeling street network structure and spatial accessibility. Martínez et al. (2023) introduced graph-constrained clustering for linear infrastructure, while Rosser et al. (2017) demonstrated that network-constrained representations better capture offender and victim mobility than grid-based models. Our own prior work, GraphTrace (Boldt et al. 2025), extends this line of research by imposing explicit geometric constraints on coverage regions. However, like most graph-based techniques, these approaches do not address the problem of optimally selecting many fixed-radius hotspots under a distinct-coverage objective.

Research gap and aim

Despite extensive work on spatial hotspot detection, two gaps remain. First, scan-statistic and KDE-based methods rarely optimize the *joint* selection of many non-overlapping hotspots; instead, they typically identify a small number of statistically significant clusters, sometimes permitting overlapping coverage. Second, many approaches depend on grid discretization or does not allow distance guarantees from cluster centers, whereas operational contexts frequently rely on fixed-radius coverage constraints, e.g., CCTV placement.

To address this gap, we propose the *GraphVenn* algorithm, a fully data-driven method that identifies high-crime locations within a predefined radius and guarantees that the top-N hotspots are *jointly* optimal under this spatial resolution. In contrast to the many heuristic approaches used in prior work (most notably greedy ranking schemes that select hotspots one at a time and therefore do not explore the full combinatorial space of possible N-hotspot configurations) GraphVenn evaluates all parallel combinations implicitly through a combination of exact pruning and an ILP-based optimizer. This ensures that the selected N hotspots maximize the number of *distinct* crime incidents covered across the entire city. GraphVenn therefore complements rather than replaces existing scan-statistic methods: while those approaches provide statistical inference and support for flexible geometries, GraphVenn offers a computationally tractable solution to the large-N, fixed-radius, distinct-coverage problem that arises in many tactical and resource-allocation settings.

Introducing graphVenn

This section introduces GraphVenn, beginning with a formal definition of the hotspot detection problem, followed by a stepwise description of the algorithm in three phases: (i) graph construction, (ii) grid sampling with Venn voting, and (iii) the optimization engine.

Problem formalization

Let $I = \{i_1, i_2, \ldots, i_n\}$ denote the set of recorded crime incidents during some chosen time period, and let $U = \{u_1, u_2, \ldots, u_m\}$ represent the set of *unique crime locations* as latitude–longitude coordinate pairs, e.g., (55.606, 13.002), here shown at three-decimal precision. By definition, $n > m$, as multiple incidents occur and/or are registered at the same location, which is consistent with Weisburd's law of crime concentration (Weisburd, 2015).

Further, let d be the maximum effective coverage radius (in meters), representing the spatial extent of a hotspot. The objective is to identify the spatial points x such that the circular regions $C(x, d)$, centered at x with radius d, contain the maximum number of distinct crime incidents from I. The analyst specifies three parameters:

- `d`: hotspot radius in meters (e.g., 50 or 100; default = 100 m),
- `N`: the number of hotspot positions to return (default = 100),
- `p`: the spatial resolution given as the number of decimals in latitude–longitude coordinates, e.g., `p=5` is equivalent to approx. 1 m between candidate points, `p=4` $\approx$ 11 m, `p=3` $\approx$ 111 m, `p=2` $\approx$ 1,113 m (default = 3).

Phase 1: graph construction

The first phase constructs an undirected weighted graph $G = (V, E)$ over the set of unique crime locations U (Trudeau, 1994). Each node $v_i \in V$ corresponds to a location u_i, and an edge $(v_i, v_j) \in E$ is created if the distance between u_i and u_j is within distance `d`. Edges are weighted by exact distances, and nodes are annotated with `crime_count` representing the number of incidents recorded at that location.

This graph encodes both the intensity and spatial connectivity of crime distributions. For more detail on the graph construction methodology, see our earlier work in (*Anonymized reference, 2025*).

Figure 1 illustrates that GraphVenn considers the `d`-neighborhood around each unique crime location *x*, i.e., the circular coverage area of radius `d`, as part of the search space. Any candidate hotspot must lie within at least one such neighborhood to capture incidents, and intermediate points such as *y* can combine coverage from multiple nearby locations (e.g., both *x* and *z*). This observation directly motivates the confined search strategy used in Phase 2.

A second important property is the principle of 2`d`-*disjointness*. If two locations are more than 2×`d` apart, their coverage areas are non-overlapping, and no crime incident can potentially be in both hotspots. This eliminates the need for allocation decisions about shared incidents. Exploiting this property allows GraphVenn to decompose the global optimization into independent subproblems, thereby reducing computational complexity while still guaranteeing exact solutions.

Phase 2: grid sampling with venn voting

The second phase discretizes the continuous search space into a *finite* set of candidate points that can be explored. Rather than scanning the entire city, GraphVenn restricts the search to the union of `d`-neighborhoods around all unique crime locations. In other words, every potential hotspot must lie within at least one circle *C*(*x*, *d*), where $x \in U$ is a recorded crime location and *C*(*x*, *d*) denotes the area within radius `d` of *x*. Any point outside this union cannot cover a single crime incident since it is further away than distance `d` from all incidents, and therefore cannot be part of an optimal solution either.

Next, each neighborhood is discretized into a grid defined by the user-selected precision `p`. Figure 2 shows an example for $\texttt{p} \in \{3, 4, 5\}$. Every candidate point within the radius-`d` mask is evaluated as a potential hotspot center. Coverage is scored using a Venn-based voting mechanism where all crime locations within radius `d` from each candidate point contributes its `crime_count` as a vote. Then the votes accumulate across overlapping neighborhoods, as shown in Fig. 3. This ensures that intermediate points between observed prior crime locations are tested, often resulting in superior coverage by combining multiple prior crime locations. Thus, placing a circle with radius `d` centered on any of the 1st ranked hotspot positions in Fig. 3 would include all three crime locations within distance `d`, i.e., summing up $50 + 80 + 120 = 250$ crime incidents.

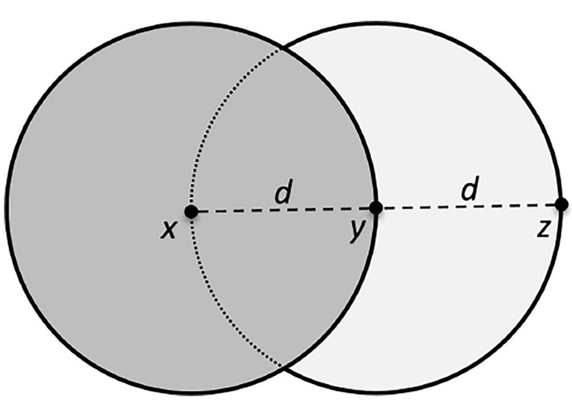

Fig. 1 Illustration of GraphVenn's spatial search space, where *x* represents a prior crime location in *U*. Any better hotspot position must lie within distance `d` of such a point, like *y*, to potentially access new crime incidents (like *z*) while still covering *x*

This exhaustive yet bounded search guarantees that all plausible hotspots are considered (given the selected `d`, `N` and `p`), while at the same time avoiding unnecessary exploration of empty space.

Effect of radius and resolution

The number of candidate points that GraphVenn must evaluate grows rapidly with both the hotspot radius `d` and the spatial resolution `p`. Table 1 illustrates this scaling effect, at `p`=4 and `d`= 100 m, 448 candidate points are tested for each unique crime location in U, whereas at `d`=1,000 m the number exceeds 44,000. At the finest resolution (`p`=5), this rises to almost 4.5 million candidate points per location, underscoring the computational cost of very fine sampling. This scaling highlights a key trade-off, that finer resolution improves spatial precision but at a sharply rising runtime. Note that the inter-candidate point distances in Table 1 reflect the maximum spacing at the equator, where one degree of longitude is longest. At higher latitudes, longitudinal spacing shrinks while the number of grid cells remains constant, resulting in denser effective sampling.

When selecting radius and spatial resolution, a practical guideline is to start by choosing a radius suitable for the problem at hand. Then, to match the spacing between candidate points to roughly 10–30 % of the hotspot radius. For instance, at `d`=100 m a resolution of `p`=4 is a good balance between coverage and efficiency since it has an inter-point distance of 11 m which corresponds to $\frac{11}{100} = 11\,\%$ of the chosen radius, i.e., within 10-30 %. At larger radii such as `d`=500 m, `p`=3 (about 111 m distance, i.e. 22%) is typically sufficient. Finer resolutions can be used if greater placement precision is essential, but they quickly become computationally expensive. Conversely, overly coarse settings (e.g., `p`=2 for `d`=100 m) leaves many locations with no candidate points at all, making hotspot detection impossible at those locations.

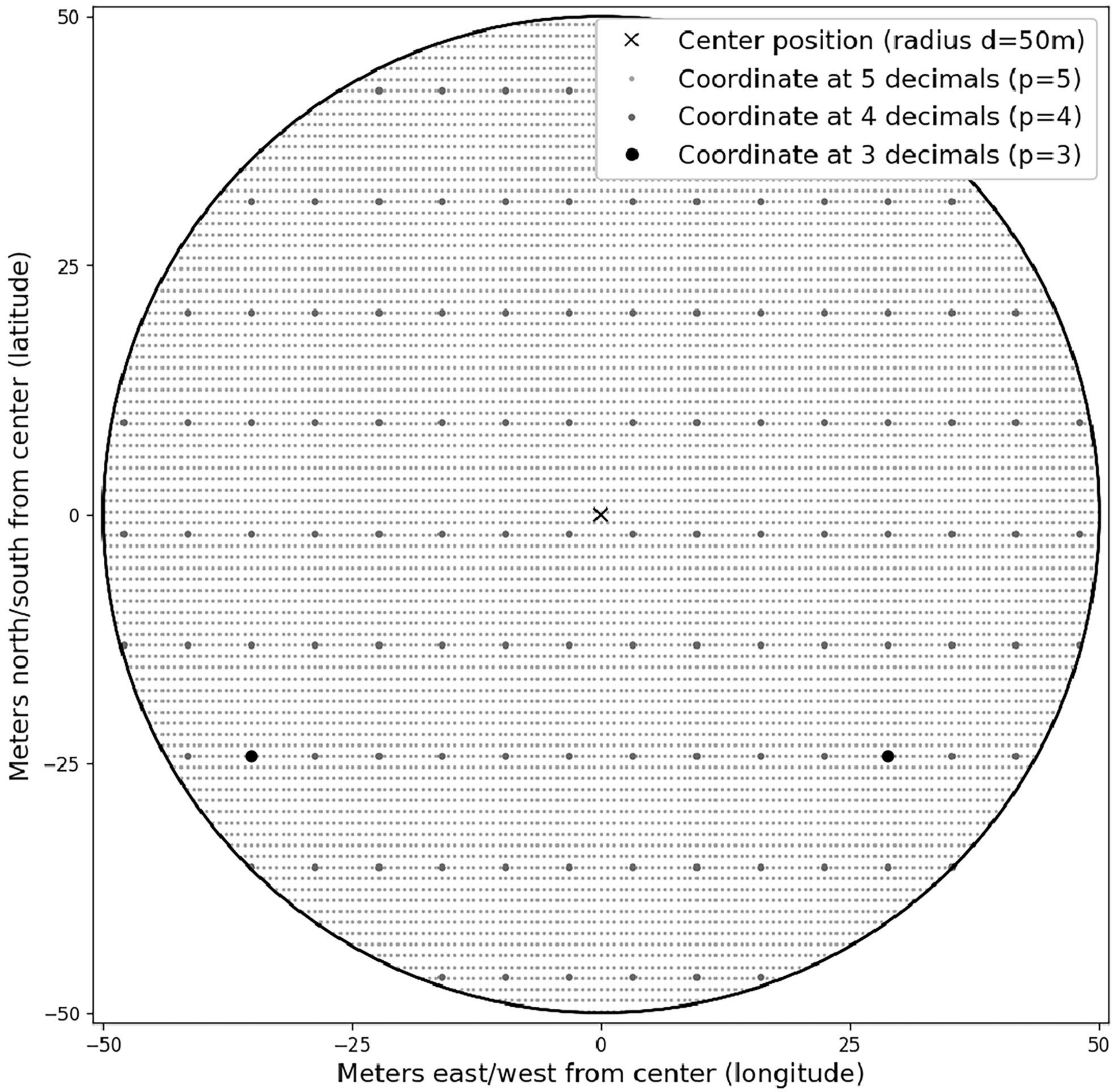

Fig. 2 Grid sampling around a crime location at three spatial resolutions ($\mathtt{p}=\{3,4,5\}$), clipped to radius $\mathtt{d}=50$ m. Higher $\mathtt{p}$ increases candidate point density

Phase 3: optimization engine

The final phase selects the best $\mathtt{N}$ hotspots from the hotspot candidates produced in Phase 2. GraphVenn offers two strategies:

- **Optimal strategy:** candidates are safely pruned using dominance checks and upper bounds, then partitioned into $2\mathtt{d}$-connected components. Each component is solved exactly by integer linear programming (ILP), framed as a maximum coverage problem (Wolsey, 2020). A dynamic program allocates the $\mathtt{N}$ hotspots across components, ensuring a globally optimal configuration.
- **Greedy strategy:** a lazy-greedy routine incrementally selects hotspots by marginal gain, avoiding double counting. This yields near-optimal coverage in a fraction of the time (Leskovec et al., 2007).**Guarantee.** At the specified resolution $\mathtt{p}$ and radius $\mathtt{d}$, GraphVenn exhaustively enumerates and scores all feasible hotspot centers. In optimal mode, the ILP optimization ensures that the returned $\mathtt{N}$ hotspots are *globally optimal*, i.e., no alternative selection can cover more unique incidents. However,

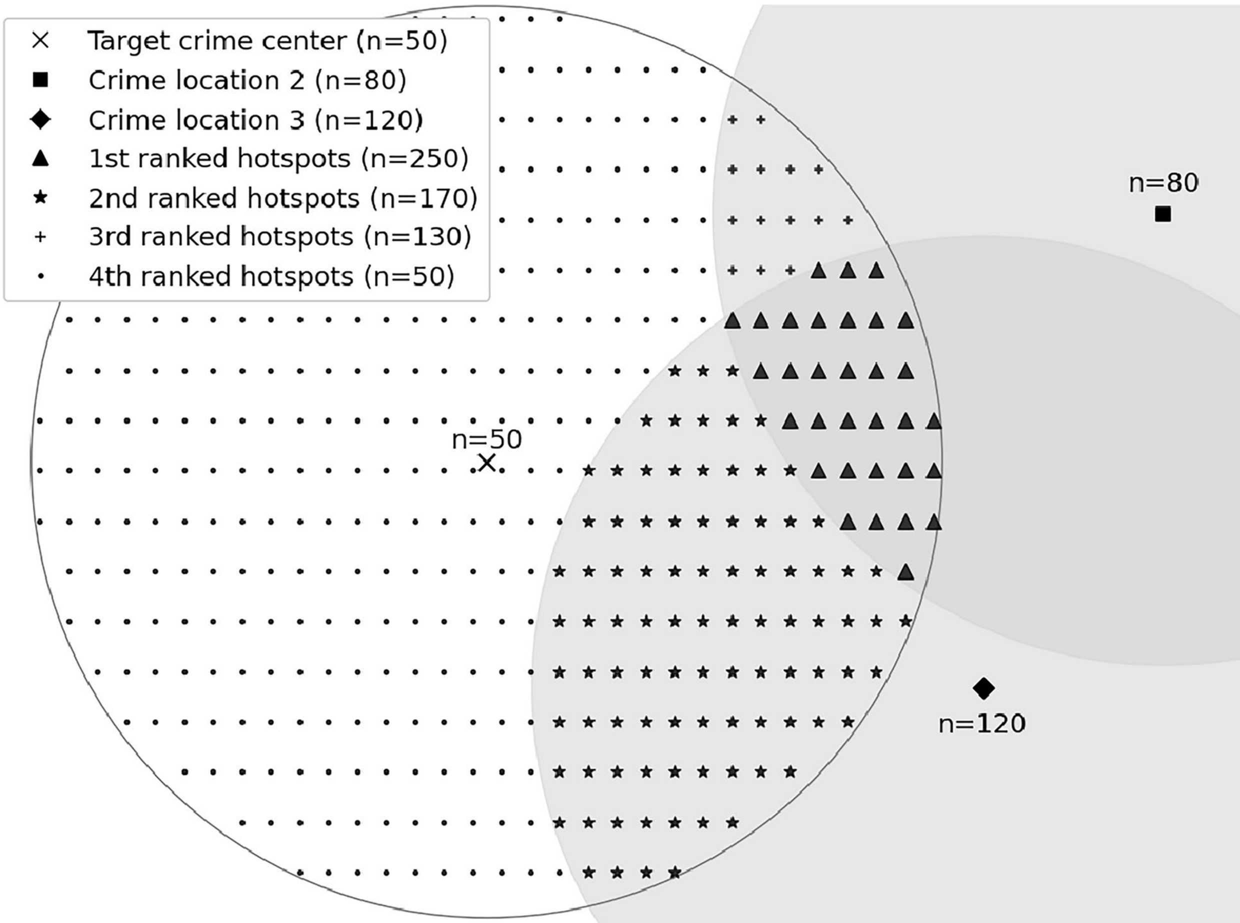

Fig. 3 Venn-based voting at precision p= 4, d= 100 m. By identifying overlapping neighborhoods, candidate hotspots can accumulate incidents from multiple locations, here boosting crime counts from 50 to 250, i.e., a 5-fold increase

Table 1 Number of candidate points evaluated, and distance between points, for three radii (d= {100, 500, 1000}m) and four spatial resolutions at p= {2,3,4,5}

Radius `d` (in meters)	Resolution `p`	Inter-point distance (in meters)	Expected number of candidate points
100 m	2 decimals	1,113 m	0
100 m	3 decimals	111.3 m	4
100 m	4 decimals	11.13 m	448
100 m	5 decimals	1.113 m	44,826
500 m	2 decimals	1,113 m	1
500 m	3 decimals	111.3 m	112
500 m	4 decimals	11.13 m	11,206
500 m	5 decimals	1.113 m	1,120,650
1,000 m	2 decimals	1,113 m	4
1,000 m	3 decimals	111.3 m	448
1,000 m	4 decimals	11.13 m	44,826
1,000 m	5 decimals	1.113 m	4,482,599

there can be other configurations of `N` hotspots with the exact same crime count since GraphVenn selects the first optimal solution. Crucially, none of the other potential configurations can give a higher crime count. In greedy mode, the result is *locally optimal*, typically within a few percent of the maximum, but without a global guarantee.

When the optimal strategy is selected, the ILP is solved using the open-source `CBC` (Coin-OR Branch-and-Cut) solver, which was chosen because it is freely available and ensures full reproducibility of the results. All ILP computations in this study were executed in single-process mode for consistency when compared with other hotspot detection methods. More information about the ILP formulation is available in the Appendix.

Method

This section outlines the methodological framework adopted to evaluate the GraphVenn algorithm.

Evaluation strategy

Unlike existing methods such as KDE, scan statistics, GraphTrace, or greedy PAI maximization, GraphVenn identifies globally optimal hotspots at a user-defined spatial resolution and radius. This property shifts the evaluation focus to: (i) computational efficiency at city scale, and (ii) practical value compared to available baselines. Accordingly, we assess GraphVenn as both a benchmark and a viable tool for applied hotspot detection through two experiments:

(i) *Computational feasibility.* Tests runtime and memory usage across three cities (Malmö, Boston, New York City), varying spatial resolution (p={2,3,4,5}) and strategy (optimal, greedy), applied year-by-year and results reported as averages.

(ii) *Comparative hotspot performance.* Compares crime coverage and execution time for both GraphVenn's greedy and optimal versions (at precisions p={4,5}) against the three baselines: KDE, PAI-Max, and GraphTrace. Coverage is measured as the sum of distinct crimes covered across the top-100 hotspots (radius d=100 m), ensuring each crime is counted only once. All methods are evaluated yearly, and results are reported as the average over the three years per city. Importantly, all methods generate the same number of hotspots with identical area. This ensures that differences in crime coverage reflect differences in spatial prioritization rather than differences in total area examined.

All experiments were conducted on a MacBook Pro (2023) with an Apple M2 Max chip and 32 GB RAM to reflect realistic hardware constraints in applied crime analysis.

Evaluation on historical crime data

We analyze crime data from Malmö, Sweden, and two U.S. cities, Boston and New York City (NYC), summarized in Table 2. Malmö provides 72,443 incidents (2018–2020) from Swedish law enforcement, retained at full spatial resolution without anonymization. The ratio of crime incidents to unique geographic locations is about $2.7 - 2.8$. For example, in 2020, 25,386 reported crimes were registered at approximately $9,123 = \frac{25,386}{2.783}$ unique locations. This dataset was also used in prior work that inspired the development of GraphVenn (Boldt et al. 2025).

For Boston (2019–2021) and NYC (2016–2018), we use the Crime Open Database (CODE)[2] to retrieve the latest three years of data available, yielding 221,677 and 1,452,797 incidents, respectively. NYC provides the largest available CODE dataset and thus serves as a scalability benchmark, while Boston represents an intermediate case with roughly double Malmö's population. In total, the study covers 1,746,917 crime incidents across the three cities.

As Table 2 shows, Boston and NYC have much higher crime-to-location ratios than Malmö, largely due to CODE's anonymization (spatial aggregation to street intersections), which reduces apparent spatial granularity. The Malmö dataset was approved for research use by the Swedish Ethical Review Authority on 15 January 2024 (decision DNR-2023-07467-01).

Investigated hotspot detection methods

Below is an introduction to the hotspot detection methods, including their configurations, included in the study.

1. `KDE`: Kernel Density Estimation (KDE) is one of the most widely used hotspot mapping techniques in criminology. It estimates a continuous crime-intensity surface by smoothing incident locations with a kernel function (typically Gaussian), where the bandwidth controls the degree of spatial smoothing (Parzen, 1962; Rosenblatt, 1956). In this study, KDE was implemented using a bandwidth of 50 m. This value was selected by evaluating bandwidths between 5 and 200 m and choosing the one that maximized the average crime count within each city. To convert the continuous KDE surface into discrete hotspots that are comparable

Table 2 Number of crimes and the ratio of crimes to unique positions for each of the three cities and year

Year	Malmö		Boston		New York City	
	Crimes	Ratio	Crimes	Ratio	Crimes	Ratio
2016	–	–	–	–	489,288	6.693
2017	–	–	–	–	479,482	6.177
2018	23,476	2.671	–	–	484,027	6.630
2019	23,581	2.719	83,700	5.040	–	–
2020	25,386	2.783	69,146	5.663	–	–
2021	–	–	68,831	5.464	–	–
$\sum$	72,443	–	221,677	–	1,452,797	–

[2] The Crime Open Database (CODE): https://osf.io/zyaqn/

to the circular hotspots used by the other methods, the surface was rasterized into grid cells whose area matches that of a circle with radius d=100 m (approximately 177 × 177 meters). Hotspots were then defined by ranking grid cells according to their KDE intensity and selecting the top N, each visualized as a circle of radius 100 m centered on the cell centroid. This ensures that KDE, GraphTrace, and GraphVenn are evaluated using equal-area hotspot units and identical rules for counting distinct crime incidents.

2. `PAI-Max`: The PAI-Max baseline implements the greedy PAI maximization step from the forecasting framework of Mohler et al. (2020), adapted here for retrospective hotspot selection. At each iteration, the method evaluates all grid-cell centers as candidate hotspot locations, selects the circular region (radius 100 m) that maximizes the Predicted Accuracy Index (PAI), removes the crimes covered by that hotspot to prevent double-counting across overlapping circles, and then repeats until the top-K ($K = 100$) hotspots are obtained. Thus, in the present study, PAI-Max functions as a greedy selection procedure applied directly to historical crime data rather than as a predictive forecasting model.
3. `GraphTrace`: GraphTrace is a graph-based method that models crime events as nodes and connects them with edges if they fall within distance d of one another. By tracing overlapping neighborhoods in this graph, GraphTrace identifies dense clusters of crimes that can serve as hotspots (Boldt et al. 2025). The method adapts dynamically to the true spatial distribution of crime incidents, rather than relying on fixed grids or kernel functions. The configuration of the method was kept to its default values, with $N = 100$ hotspots, 5m cell spacing between grid points.
4. `Greedy GraphVenn(p=4,5)`: The greedy variant of GraphVenn sequentially selects hotspot centers that maximize additional crime coverage given previously chosen hotspots, using a grid of candidate locations at resolution p. Unlike grid-based baselines that evaluate the full study area, Greedy GraphVenn restricts its search to the union of all distance-d neighbourhoods around prior crime locations, allowing it to focus its analysis on those specific areas. This, in turn, allows much finer spatial sampling (e.g., p=4 or 5) exactly where it matters. The greedy heuristic approach reduces computational complexity while still yielding near-optimal solutions in practice. Parameter p was changed according to the experimental setup, while N was kept at 100 hotspots, and radius d=100 m.
5. `Optimal GraphVenn(p=4,5)`: The optimal variant of GraphVenn formulates the hotspot detection task as an integer linear program and computes the globally optimal selection of hotspots given the candidate grid at resolution p. While computationally more demanding, it provides an upper bound on achievable coverage (given N fixed-radius circular hotspots with radius d) against which other methods can be benchmarked. Parameter p was changed in the experiment similarly as for the greedy GraphVenn.

Limitations and validity threats

This study has several limitations. First, the spatial precision of crime geocoding, which is often coarse (Gerell, 2018). Analyses finer than roughly p=4 decimals are of limited value, though improved geocoding could increase utility, and tools like GraphVenn may also motivate better recording practices. This study fixed d=100 m and N=100 hotspots, consistent with prior work (Boldt, et al. 2025), in order to ensure that the baseline method PAI-Max remain within computational limits, whose runtime increases sharply as these parameters expand.

The study was diagnostic rather than predictive, focusing on historical detection rather than forecasting. Further, only circular fixed-radius hotspots were studied, while there obviously are other forms of hotspots, e.g., arbitrary hotspot areas (captured by polygons) and hotspots bound to street segments. This simplification of using fixed-radius hotspots aids computation, but future extensions could support adaptive shapes and this is further discussed in the discussion section.

The evaluation was limited to large cities with rich data and to full calendar years. Smaller jurisdictions, rural areas, or shorter temporal scales (weeks or months) were not assessed but are in principle compatible with the method. Finally, while the experimental design reduces internal validity threats by controlling inputs, external validity is more limited. Results are based on three cities across two contexts and three years of data for each city; generalizability to other places or periods is uncertain and requires further testing.

Results

The results are organized into the two experiments. The first examines the computational feasibility of GraphVenn in terms of runtime and memory usage. The second evaluates its analytical usefulness in terms of crime coverage.

Experiment 1: computational feasibility

We measured runtimes for both versions of GraphVenn: the greedy approximation, intended for fast analysis, and the globally optimal variant, which guarantees maximum coverage.

Runtimes for greedy graphvenn

Figure 4 reports runtimes for selecting the top-100 hotspots using the greedy optimization approach and a radius of d=100 m. At precision p=5, Malmö and Boston required 26 and 36 min, while NYC took about 1.5 h. These values remain feasible for offline or batch analyses. More importantly, reducing the resolution to p=4 again yielded dramatic speedups, where all cities completed within one minute, nearly two orders of magnitude faster than p=5. This makes real-time or near-real-time hotspot analysis entirely practical.

As a brief note, we also tested two larger radii: d=500 m and d=1,000 m. At p=3, corresponding to a grid spacing of about 22% of the radius (within the recommended 10–30% interval), runtimes were only a few seconds for Malmö, and under a minute for Boston and NYC. At p=4, runtimes rose to several minutes but remained far below the p=5 results for d=100 m. These findings confirm that GraphVenn maintains practical runtimes even at larger hotspot radii, offering flexibility for different analytical scales.

Runtimes for optimal GraphVenn

The globally optimal mode, which guarantees maximum unique crime coverage, required longer runtimes than the greedy version but remained computationally feasible. For d=100 m, runtimes grew substantially. At p=5, Malmö and Boston each required about 2.5 h, while NYC took just under 5.5 h, see Fig. 5. These values represent the computational upper bound for full-resolution optimal analysis on large urban datasets at 1-meter precision. Crucially, reducing the resolution to p=4 again produced major improvements, as runtimes dropped to under 13 min for Malmö, 8 min for Boston, and 16 min for NYC, nearly two orders of magnitude faster than p=5. Thus, even for the largest city tested, globally exact solutions can be computed within time frames compatible with overnight or batch analysis.

Memory usage

Memory requirements followed a similar scaling pattern. At p=2, RAM usage was about 4.1 GB for Malmö and Boston and 4.2 GB for NYC for the optimal version of GraphVenn. At p=5, usage increased to 13.8 GB, 14.9 GB, and 15.2 GB, respectively (greedy variants required significantly less memory). These values remain well within

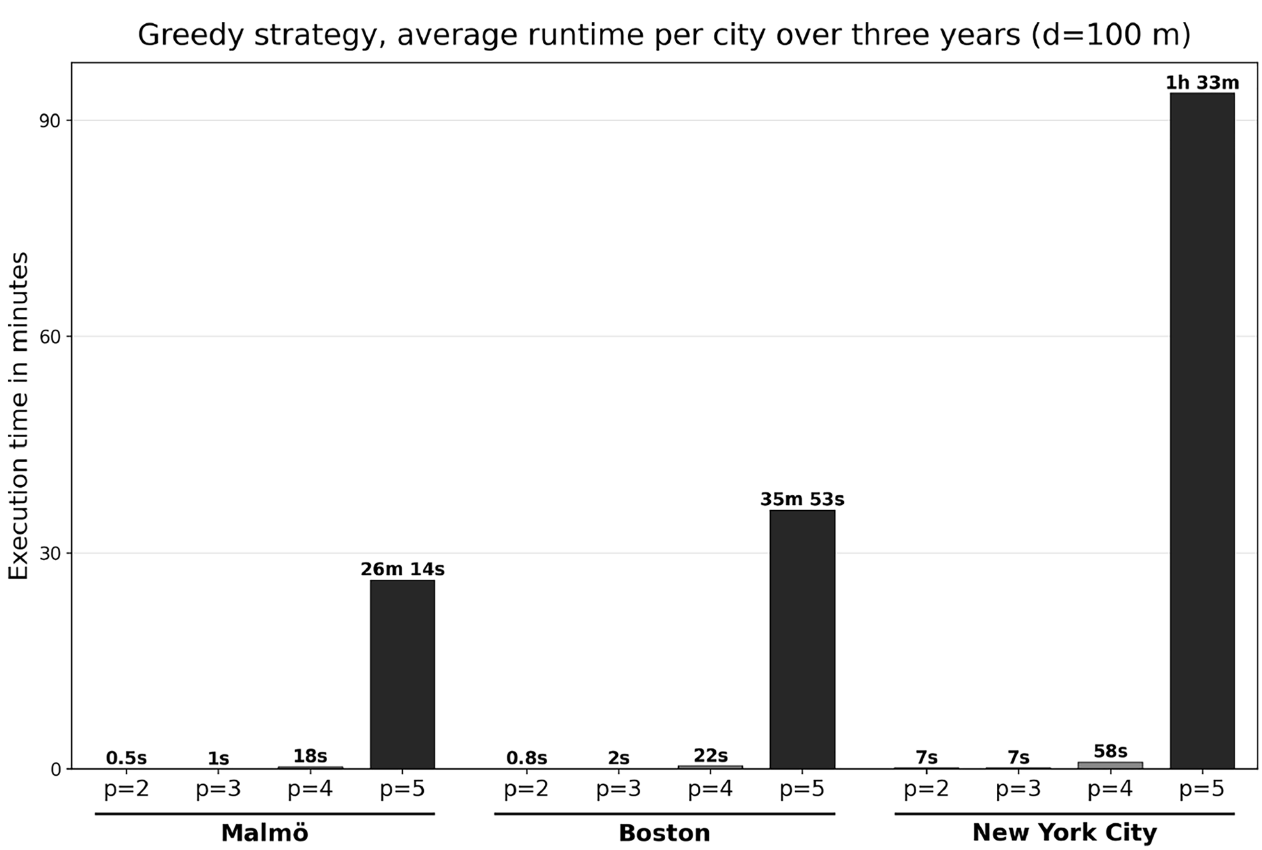

Fig. 4 Average runtime for greedy GraphVenn across three years in Malmö, Boston, and NYC, using hotspot radius d=100 m and resolutions p={2,3,4,5}

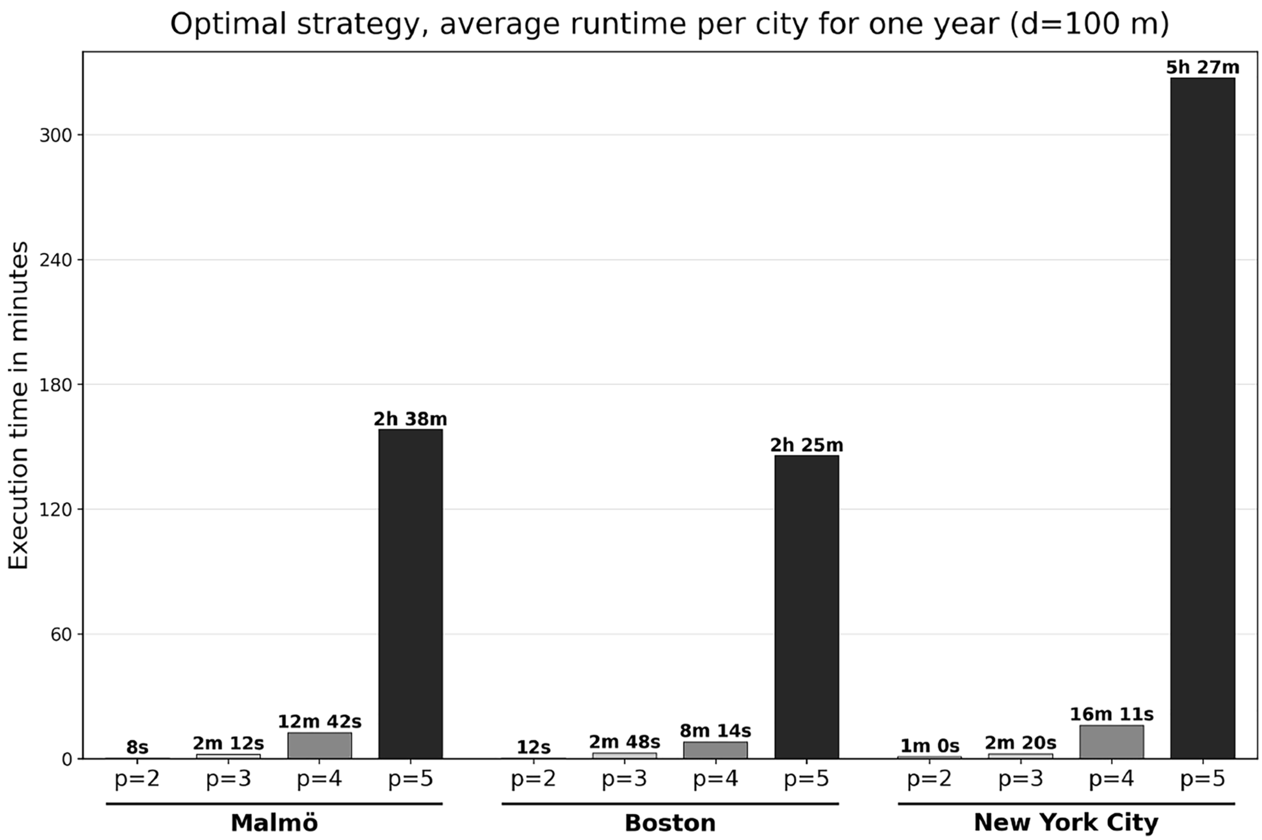

Fig. 5 Average runtime for optimal GraphVenn across three years in Malmö, Boston, and NYC, using hotspot radius d=100 m and resolutions p={2,3,4,5}

the capabilities of modern consumer-grade desktop and laptop hardware.

Experiment 2: analytical usefulness

Having established that GraphVenn is computationally feasible at city scale, we now turn to its analytical usefulness by comparing hotspot detection performance against established baseline methods, in terms of crime coverage. Figure 6 compares GraphVenn's optimal variant at precision p=5 with its greedy variants (p=4 and p=5), and the three state-of-the-art baselines (note the logarithmic Y axis). Across all three cities, GraphVenn consistently identifies more crimes, with higher medians and upper quartiles, although GraphTrace is not far behind. The difference between the optimal GraphVenn and its greedy versions p=4 and p=5 is small. Notably, in Malmö and NYC, GraphVenn's greedy variants, and in one instance GraphTrace, record the single highest hotspot counts, even slightly surpassing optimal GraphVenn. This outcome is expected and is explained further in the discussion section.

In Malmö, Optimal(p=5) achieves slightly higher counts than the greedy variants, reflecting the advantage of exact optimization at full spatial resolution. In Boston and NYC, however, the greedy and optimal versions perform almost identically, with only minor differences between p=4 and p=5. GraphTrace remains competitive in Malmö but deteriorates sharply under the data aggregation for the U.S. cities, while KDE and PAI-Max, though less sensitive to aggregation due to their grid-based nature, still fall well behind GraphVenn. These results highlight that greedy GraphVenn attains nearly optimal coverage at a fraction of the runtime, combining efficiency with robustness across both full-resolution and aggregated datasets.

Table 3 shows that greedy GraphVenn(p=5) achieves within 0.3–1.9% of the optimal solution across cities. GraphTrace remains competitive in Malmö but again deteriorates under aggregation, dropping to 32% in Boston and 55% in NYC. PAI-Max showed moderate performance in Malmö and Boston, but was infeasible for NYC due to prohibitive runtimes. KDE consistently lags behind, reaching only 68–79% of the optimal.

Baseline runtimes revealed substantial performance differences between methods. On average, PAI-Max required 3 h 34 min in Malmö, and 4 h 11 min in Boston, while processing had to be terminated for NYC because runtime was projected to exceed two weeks. KDE, by contrast, was the fastest method, completing in approximately 3 s, 10 s, and 5 min for Malmö, Boston, and NYC,

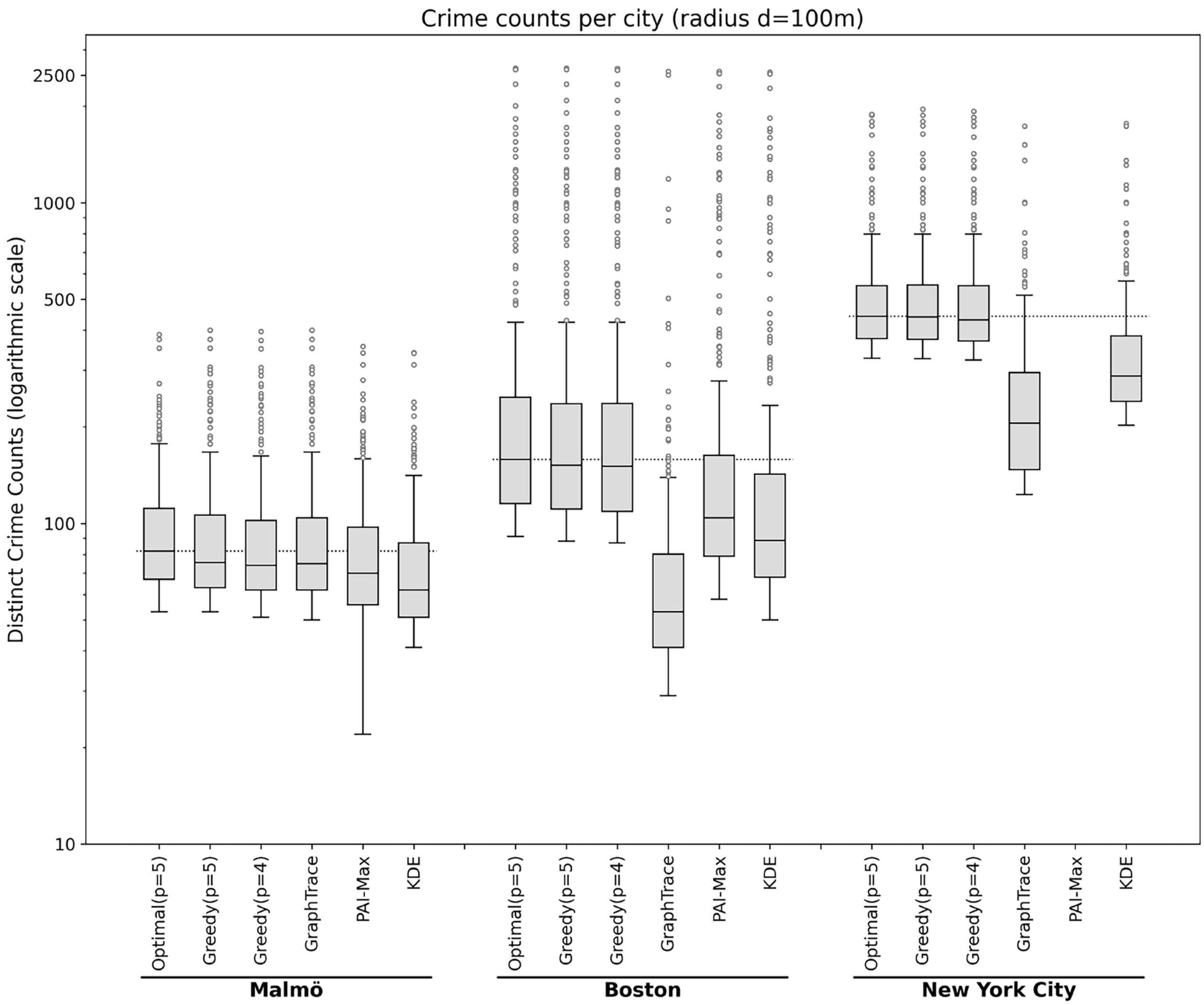

Fig. 6 Logarithmic box plot showing the unique crime counts per city for the top-100 hotspots with radius d=100 m for: GraphVenn's Optimal(p=5) and Greedy versions p=5 and p=4, GraphTrace, PAI-Max, and KDE

Table 3 Mean distinct crime counts per city, averaged across three years, with relative performance (in %) compared to the global optimal GraphVenn at p= 5 (d= 100 m). The candidate method closest to the optimal count in bold font

Method	Malmö		Boston		New York City	
	Mean	**Rel. (%)**	**Mean**	**Rel. (%)**	**Mean**	**Rel. (%)**
GraphVenn Optimal(p=5) (*max. coverage possible*)	9,852	100.0%	28,832	100.0%	51,522	100.0%
GraphVenn Optimal(p=4)	9,628	97.7%	28,555	99.0%	50,992	99.0%
GraphVenn Greedy(p=5)	9,660	**98.1%**	28,592	**99.2%**	51,385	**99.7%**
GraphVenn Greedy(p=4)	9,452	95.9%	28,332	98.3%	50,884	98.8%
GraphTrace	9,543	96.9%	9,289	32.2%	28,241	54.8%
PAI-Max	9,021	91.6%	23,278	80.7%	n/a	n/a
KDE	7,795	79.1%	20,908	72.5%	35,098	68.1%

respectively. GraphTrace occupied the middle ground, with runtimes of 5 min, 23 min, and 26 min across the same cities.

Overall, these results show that greedy GraphVenn delivers near-optimal crime coverage across different spatial scales and datasets, while the globally optimal variant remains computationally feasible for full-city analyses.

Figure 7 presents a zoom-in of the top hotspots identified in New York City (2018) by both the optimal and

Optimal GraphVenn(p=5) Greedy GraphVenn(p=5)

Fig. 7 Area surrounding the highest-ranked hotspot identified by both the optimal and greedy versions of GraphVenn in New York City (2018). Hotspots ranked within the top 10 for each method are shown in yellow; all others are shown in red. Transparency emphasizes the relative hotspot rank

the greedy version of GraphVenn(p=5). Although the two variants produce broadly similar patterns, there are notable differences in their highest-ranked selections. The greedy version assigns rank 1 to a location east of Penn Station covering Herald Square (an area that the optimal GraphVenn rank 17), while the area that the optimal GraphVenn identifies as its top hotspot is ranked 21 by the greedy algorithm. In contrast, the 2nd ranked hotspot is practically identical across the two variants. Overall, within the region shown in the figure, the optimal GraphVenn places eight hotspots, whereas the greedy version allocates seven, reflecting the small, but meaningful differences in how the two strategies distribute hotspots under the distinct-coverage objective. This difference in hotspot ranking strategies is discussed further in the discussion section.

Figure 8 shows the KDE and GraphTrace hotspots within the same region. Both methods position their highest-ranked hotspots roughly 200 m east of optimal GraphVenn's top hotspot, and each highlights several additional locations along the major corridors leading toward Times Square. KDE places three hotspots in horizontally adjacent grid cells stretching from east of the Port Authority Bus Terminal toward Broadway, including one hotspot directly north of the bus terminal. GraphTrace, in contrast, places a hotspot just northeast of the bus terminal and distributes four additional hotspots along Broadway, where at least one partially covers Times Square. In total, KDE identifies eight hotspots in this area, GraphTrace identifies seven, and GraphVenn (optimal and greedy) identifies eight and seven respectively. Thus, although the methods detect a similar number of hotspots, they allocate them to different micro-locations and emphasize local crime concentrations in different ways. Notably, all methods except optimal GraphVenn place their rank-1 hotspot in approximately the same location, while their rankings diverge from rank 2 onward. The greedy variant of GraphVenn nevertheless places its second-ranked hotspot in the same location as the optimal solution.

Discussion

Crime hotspot detection is, at its core, a maximum coverage problem belonging to the class of NP-hard problems (Garey & Johnson, 1979). In general, such problems exhibit an exponential growth in the number of possible combinations as the search space increases, making brute-force enumeration computationally infeasible for realistic datasets. However, human criminal behavior tends to manifest in geographic space with structural regularities that can be exploited to reduce this complexity to tractable levels. Urban crime data exhibit key spatial properties: (i) crime is highly concentrated, allowing many candidate locations to be pruned early; and (ii) clusters are spatially separable, enabling decomposition into independent components. By restricting the analysis to the bounded search space defined by the d-neighborhood around prior crime locations, the effective complexity can be reduced even further. Together,

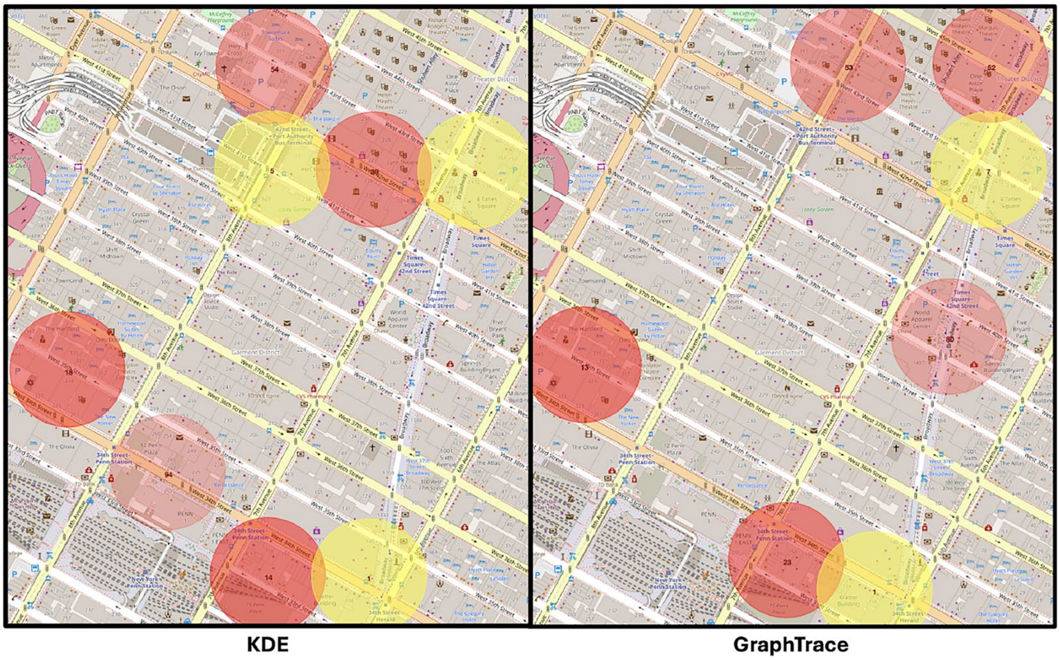

Fig. 8 Same area as in the previous figure, but for the hotspots detected by KDE and GraphTrace

these properties make global optimization feasible even for large, real-world crime datasets.

Given these properties, crime analysts can further trade runtime against accuracy by adjusting the spatial resolution `p`, achieving near-interactive runtimes for greedy analyses and overnight runtimes for globally optimal ones. This makes exact hotspot detection feasible in criminological research and practical crime prevention without prohibitive computational cost. By more precisely identifying high-crime locations, GraphVenn enables analytical focus on the most crime-dense areas, increasing the likelihood of detecting statistically significant relationships between experimental variables and spatial crime patterns when compared to conventional control area selection.

Beyond spatial resolution, we also examined GraphVenn's sensitivity to the hotspot radius `d` and the number of hotspots `N`. Internal tests show that, when `p` is chosen in proportion to `d` (as recommended in Sect. 3.3.1), greedy GraphVenn maintains essentially stable runtimes across the radii tested (50 m to 1,000 m) and scales smoothly with increasing `N` (evaluated up to 1,000 hotspots in the NYC dataset). In contrast, the optimal ILP variant becomes substantially more expensive as `N` increases, making large (`N`,`d`)-sensitivity sweeps less practical for the present study. These observations suggest that a more systematic exploration of parameter sensitivity is a promising direction for future work, particularly for understanding operational trade-offs in different deployment contexts. Finally, all methods were executed in a single-process (non-parallel) configuration to ensure fair runtime comparisons. In practice, however, both the hotspot-candidate evaluation phase and the component-level optimization steps are naturally parallelizable. Leveraging multiprocessing or distributed execution, especially when analyses are run separately within districts rather than citywide, could further reduce runtimes and make larger values of `N` feasible also for the optimal GraphVenn formulation.

Previous hotspot detection methods rely on heuristic approximations to make the crime-mapping problem computationally manageable. GraphVenn challenges this assumption by demonstrating that global optimality is practically attainable, at least within the scope of this study. This allows criminological analyses to be grounded in provably optimal solutions, reducing reliance on approximation error. Consequently, researchers and practitioners can assess how closely heuristic methods approach the optimum, fostering more transparent evaluation, methodological refinement, and calibrated expectations about attainable performance.

Table 4 illustrates the difference between the greedy and globally optimal strategies using results from Malmö. In the 10 best hotspots out of the total top-100, several competing methods (including Greedy (`p=5`), GraphTrace, and in one case PAI-Max) identify hotspots with more crimes than those chosen by the optimal solution. However, this does not mean the optimal solution "missed" these hotspots. Instead, it reflects a deliberate choice by the optimization engine, that some high-count candidates are set aside because selecting them would overlap with other hotspots, reducing the total distinct

Table 4 Top-10 and bottom-10 hotspot crime counts for each method in Malmö (d=100 m)

Method	Ranks 1–10 out of total 100									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Optimal (p=5)	372	252	238	218	212	195	187	182	181	169
Greedy (p=5)	**376**	**290**	**281**	**248**	**234**	**219**	**207**	**187**	164	159
GraphTrace	**375**	**271**	**269**	**245**	**235**	**234**	**223**	**209**	160	159
PAI-Max	336	**254**	224	214	198	188	182	170	165	159
KDE	331	222	210	182	176	173	165	156	142	133
	Ranks 91–100 out of total 100									
	91	92	93	94	95	96	97	98	99	100
Optimal (p=5)	**60**	**60**	**59**	**58**	**58**	**58**	**57**	**56**	**56**	**56**
Greedy (p=5)	58	57	57	57	56	56	56	**56**	55	55
GraphTrace	54	54	54	53	53	53	53	52	51	50
PAI-Max	54	54	53	53	53	53	52	52	52	52
KDE	46	46	45	45	45	45	45	44	44	44

In upper section of the table any crime count by the candidate methods with higher counts than Optimal(p=5) is marked with bold font. In the lower section, the highest count is highlighted for each rank

crimes covered across all N selections. By slightly adjusting hotspot placements, some "redistribution" of crime incidents across lower-ranked hotspots is possibile. As a result, the optimal solution secures a higher overall coverage, even if it sacrifices peak counts in the very top ranks. This trade-off becomes more apparent toward the end of the top-10 list, where none of the competing methods continue to match or exceed the optimal solution.

The pattern is even clearer in the bottom ranks (91–100), where Optimal (p=5) consistently maintains higher counts than all alternatives. Greedy (p=5) matches the optimum in one isolated instance (rank 98), while GraphTrace, PAI-Max, and KDE fall behind. Taken together, the comparison highlights how the globally optimal solution prioritizes total distinct coverage across the full set of hotspots, while greedy and heuristic approaches risk exhausting overlapping crimes too early.

However, GraphVenn comes with several limitations and opportunities for extension, and next we discuss five such cases. First, GraphVenn currently restricts hotspots to fixed-radius circular regions, since each evaluation point is analyzed independently. However, these evaluation points could in principle be extended by merging nearby evaluation points (with equally high or higher crime counts) into multi-point irregular shaped hotspots. Provided that none of the included points is already covered by a higher-ranked hotspot. Such a simple cluster-extension rule would allow GraphVenn to generate multi-point hotspots of arbitrary shape while still preserving the global optimum in total crime coverage. As an illustrative example, we refer to the set of evaluation points with a crime count of 250 in Fig. 3, which could form a single merged hotspot.

Second, crime data geocoding is often coarse, making very fine spatial resolutions (e.g., p= 5 or higher) of limited practical value. At the same time, this coarseness reduces the effective computational burden, enabling GraphVenn to solve city-scale problems such as New York City within minutes at p= 4. Future work could explore dynamic spatial resolutions, for example by adapting p to local data density or snapping nearby incidents to coarser grids to preserve tractability as datasets become increasingly precise.

Third, GraphVenn currently requires that users specify both the number of hotspots and the radius; an alternative formulation using an explicit area-budget constraint (e.g., limiting coverage to at most $X\,\%$ of a jurisdiction) may offer greater flexibility in some operational settings. A further practical consideration is that many police agencies allocate resources at the level of districts or beats rather than citywide. In such cases, GraphVenn can be run independently within each jurisdictional unit, ensuring that each district receives its own optimal hotspots while also substantially reducing complexity of the problem, and potentially lowering runtimes by orders of magnitude.

Four, although GraphVenn is designed as a descriptive tool, its underlying crime-impact values (i.e., the standalone counts for each evaluated location) provide a high-resolution measure of local crime intensity that may be useful as an input to predictive models. These scores could, for example, serve as covariates or priors in spatio-temporal forecasting frameworks, or help seed predictive hotspot-identification algorithms by identifying stable micro-places with persistently elevated crime levels. Integrating such descriptive high-resolution information into forecasting models represents a promising direction for future research.

Five, and finally, extending the current framework to incorporate environmental constraints such as rivers, major roads, building geometries, or line-of-sight considerations could yield hotspot representations that better reflect real operational constraints.

Overall, the experimental results demonstrate that globally optimal hotspot solutions can be computed for large cities using full-year datasets. Even at 1 m resolution, runtimes remain practical with about 2.5 h for Malmö and Boston, and about 5.5 h for NYC due to its larger geographical area and higher number of crime incidents. These runtimes are well within reasonable limits for practical use by crime analysts. If computational speed is prioritized over exact optimality, the greedy GraphVenn(p=5) solutions achieve results within only 0.3 %–1.9 % of the global optimum while completing in roughly 0.5 h for Malmö and Boston, and in 1.5 h for NYC. If this is considered too long for NYC, however, reducing to p=4 provides a more practical balance, completing in less than a minute while still remaining close to the optimal solution at 98.8%. Compared to the commonly used baseline KDE, the greedy GraphVenn method is 5.5 $\times$ faster while at the same time achieving results that are 30.7 percentage points more accurate (98.8 % instead of 68.1 %).

Conclusions

The experimental results in this work indicate that globally optimal hotspot detection is feasible for both criminological research and applied crime prevention. By exploiting structural properties of crime data, such as spatial concentration, clustered organization, and a bounded search space, GraphVenn computes exact top-N hotspots at city scale. Across Malmö, Boston, and New York City, GraphVenn consistently outperformed established methods such as KDE, greedy PAI maximization, and GraphTrace, often by wide margins.

Two implications follow. First, law enforcement practitioners gain a tool for identifying intervention sites in situations where fixed-radius hotspots are operationally relevant (e.g., CCTV placement), with confidence that no better configuration exists under the chosen constraints. Second, researchers can quantify the *true upper bound* of crime concentration under explicitly defined hotspot constraints (i.e., fixed radius and spatial resolution), providing a clear benchmark against which approximate or predictive methods can be evaluated. In this sense, GraphVenn establishes an optimal benchmark within a clearly defined operational problem, rather than aiming to replace statistical hotspot detection or predictive approaches.

Methodologically, GraphVenn bridges the gap between optimality and feasibility. Its optimal mode establishes the theoretical maximum crime coverage, while its greedy variant produces near-identical results at runtimes suitable for routine use. Together, these capabilities enable both rigorous benchmarking and practical deployment. To facilitate adoption and further development, the algorithm and source code are released under an open-source license.

Future work

Several directions for future research emerge from this study. First, extending GraphVenn beyond fixed-radius circular hotspots to support irregular, network-aligned, or composite shapes would help capture crime patterns shaped in other forms than circles, e.g., polygonal areas taking into account environmental barriers, while preserving the benefits of global optimization. Second, GraphVenn could be adapted to alternative constraint formulations, such as limiting the total hotspot area (e.g., at most X% of a jurisdiction) or running the method independently within districts or beats, which may better reflect operational resource boundaries and can substantially reduce computational complexity. Third, the method's underlying crime-impact values offer high-resolution descriptive information that could be integrated into spatio-temporal forecasting models, providing a bridge between descriptive and predictive hotspot analysis. Finally, systematic sensitivity analyses (including variation in the hotspot radius, number of hotspots, and spatial resolution) represent natural extensions, as initial tests indicate that greedy GraphVenn scales well across a wide parameter range, whereas the optimal ILP variant becomes more demanding for large N. Exploring these extensions would deepen understanding of how GraphVenn can support both research and practical crime prevention.

Appendix

Integer linear program for optimal hotspot selection

The globally optimal hotspot configuration in GraphVenn is obtained by solving a weighted *maximum-coverage problem* under a no-double-counting constraint. Let $\mathcal{H} = \{1, \ldots, n\}$ denote the set of candidate hotspot centers, and let $\mathcal{U} = \{1, \ldots, m\}$ be the set of unique crime locations. Each location $j \in \mathcal{U}$ has an associated weight w_j equal to the number of crime incidents recorded at that location. For each candidate hotspot *i*, let $C_i \subseteq \mathcal{U}$ denote the subset of locations that fall within the fixed radius *d* of hotspot *i*.

We introduce a binary decision variable x_i for each hotspot candidate and a binary variable y_j for each crime location. The variable x_i indicates whether hotspot *i* is selected, and y_j indicates whether location *j* is covered by at least one selected hotspot. The ILP formulation is:

$$\max_{\{x_i, y_j\}} \sum_{j \in \mathcal{U}} w_j y_j$$
$$\text{s.t.} \;\; y_j \leq \sum_{i: j \in C_i} x_i, \forall j \in \mathcal{U},$$
$$\sum_{i \in \mathcal{H}} x_i \leq N,$$
$$x_i \in \{0,1\}, \;\; y_j \in \{0,1\}, \;\; \forall i \in \mathcal{H}, j \in \mathcal{U}.$$

The first set of constraints ensures that a location may be counted as covered only if at least one selected hotspot includes it. The second constraint restricts the number of selected hotspots to the user-specified budget N. The objective maximizes the total number of *distinct* crime incidents covered across all hotspots.

This ILP is solved using the open-source `CBC` (Coin-OR Branch-and-Cut) solver accessed through the `PuLP` Python interface. CBC was chosen because it is freely available and ensures full reproducibility of all reported experiments. All ILP computations in this study were executed in single-threaded mode for consistency when compared against other hotspot detection methods.

Acknowledgements
I am grateful to Dr. Manne Gerell for managing the ethical review process, leading the research grant application that funded this work, and sharing his extensive criminological expertise, particularly in place-based studies and methods. I thank my friend and colleague Anton Borg for many stimulating discussions on criminology-related data science and methodology, and Karl Kronkvist for his criminological expertise and highly valued skeptical questioning in methodological discussions. Finally, I wish to thank my Ph.D. student Kenneth Lewenhagen for his contributions on CCTV placement and interactive web visualizations.

Author contributions
All authors read and approved the final manuscript.

Funding
Open access funding provided by Blekinge Institute of Technology, Sweden. The current study has been funded by the Swedish Research Council under Grant 2022–05442. Open access funding provided by Blekinge Institute of Technology, Sweden.

Data availability
The source code for GraphVenn is released under an open-source license and is publicly available at https://github.com/boldten/GraphVenn/. The datasets for Boston and New York City are publicly available and may be used for replication.

Declarations

Conflict of interest
The author(s) declare no Conflict of interest.

Received: 10 October 2025 / Accepted: 12 January 2026
Published online: 21 February 2026

References

Abolhassani, A., & Prates, M. O. (2021). An up-to-date review of scan statistics. *Statistics Surveys, 15*, 111–153. https://doi.org/10.1214/21-SS132

Berk, R., Sherman, L., Barnes, G., Kurtz, E., & Ahlman, L. (2008). Forecasting murder within a population of probationers and parolees: A high stakes application of statistical learning. *Journal of the Royal Statistical Society Series A: Statistics in Society, 172*(1), 191–211. https://doi.org/10.1111/j.1467-985X.2008.00556.xhttps://academic.oup.com/jrsssa/article-pdf/172/1/191/49601145/jrsssa_172_1_191.pdf.

Boldt, M and Lewenhagen, K and Borg, A and Kronkvist, K and Gerell, M, (2025). GraphTrace: A Graph-Guided Hotspot Detection Method for CCTV Placement. *Journal of Quantitative Criminology* 1–32. https://link.springer.com/article/10.1007/s10940-025-09623-9

Braga, A., Turchan, B., Papachristos, A., & Hureau, D. (2019). Hot spots policing and crime reduction: An update of an ongoing systematic review and meta-analysis. *Journal of Experimental Criminology, 15*(3), 289–311. https://doi.org/10.1007/s11292-019-09372-3

Brantingham, P., & Brantingham, P. (1995). Criminality of place: Crime generators and crime attractors. *European Journal on Criminal Policy and Research, 3*, 5–26.

Caplan, J. M., Kennedy, L. W., & J.M. and. (2011). Risk terrain modeling: Brokering criminological theory and gis methods for crime forecasting. *Just Quart, 28*(2), 360–381. https://doi.org/10.1080/07418825.2010.486037

Chainey, S. (2014). *Examining the extent to which hotspot analysis can support spatial predictions of crime*. Ph. D. thesis, University College London.

Cohen, L. E., & Felson, M. (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review, 44*(4), 588–608.

Cornish, D.B., & Clarke, R.V. (2016). *Environmental criminology and crime analysis*, Chapter The rational choice perspective, pp. 48–80. Londin, UK: Routledge.

Fitzpatrick, D., Ni, Y., & Neill, D. B. (2021). Support vector subset scan for spatial pattern detection. *Computational Statistics & Data Analysis, 157*, 107149. https://doi.org/10.1016/j.csda.2020.107149

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-Completeness*. San Francisco: W. H. Freeman and Company.

Gerell, M. (2018). Quantifying the geographical (un)reliability of police data. *Nordisk Politiforskning, 5*(2), 157–171. https://doi.org/10.18261/issn.1894-8693-2018-02-05

Gorr, W. L., & Lee, J. (2015). Early warning system for temporary crime hot spots. *Journal of Quantitative Criminology, 31*, 25–47. https://doi.org/10.1007/s10940-014-9223-8

Haberman, C. P., Sorg, E. T., & Ratcliffe, J. H. (2017). Assessing the validity of the law of crime concentration across different temporal scales. *Journal of Quantitative Criminology, 33*, 547–567.

Hardyns, W., Snaphaan, T., & Pauwels, L. (2019). Crime concentrations and micro places: An empirical test of the "law of crime concentration at places" in belgium. *Australian & New Zealand Journal of Criminology, 52*(3), 390–410. https://doi.org/10.1177/0004865818807243

Hu, Y., Wang, F., Guin, C., & Zhu, H. (2018). A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation. *Applied Geography, 99*, 89–97. https://doi.org/10.1016/j.apgeog.2018.08.001

Johnson, S. D., Bernasco, W., Bowers, K. J., Elffers, H., Ratcliffe, J., Rengert, G., & Townsley, M. (2007). Time patterns of risk: A cross national assessment of residential burglary victimization. *Journal of Quantitative Criminology, 23*, 201–219. https://doi.org/10.1007/s10940-007-9025-3

Kennedy, L. W., Caplan, J. M., Piza, E., & J.M. and. (2011). Risk clusters, hotspots, and spatial intelligence: Risk terrain modeling as an algorithm for police resource allocation strategies. *Journal of Quantitative Criminology, 27*, 339–362. https://doi.org/10.1007/s10940-010-9126-2

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics - Theory and Methods, 26*(6), 1481–1496. https://doi.org/10.1080/03610929708831995

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '07, New York, NY, USA, pp. 420–429. Association for Computing Machinery.

Martínez, A. F., Chaudhuri, S., Díaz-Avalos, C., Juan, P., Mateu, J., & Mena, R. H. (2023). Clustering constrained on linear networks. *Stochastic Environmental Research and Risk Assessment, 37*, 569–594. https://doi.org/10.1007/s00477-022-02376-y

Mohler, G., Porter, M., Carter, J., & LaFree, G. (2020). Learning to rank spatio-temporal event hotspots. *Crime Science, 9*(1), 1–12.

National Institute of Justice. (2019). Crimestat: Spatial statistics program for the analysis of crime incident locations. https://nij.ojp.gov/topics/articles/crimestat-spatial-statistics-program-analysis-crime-incident-locations. Accessed: YYYY-MM-DD.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics, 33*(3), 1065–1076. https://doi.org/10.1214/aoms/1177704472

Patil, G. P., & Taillie, C. (2004). Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environmental and Ecological Statistics, 11*(2), 183–197.

Piza, E. L., & Carter, J. G. (2018). Predicting initiator and near repeat events in spatiotemporal crime patterns: An analysis of residential burglary and motor vehicle theft. *Justice Quarterly, 35*(5), 842–870. https://doi.org/10.1080/07418825.2017.1342854

Porter, M. D., & Reich, B. J. (2012). Evaluating temporally weighted kernel density methods for predicting the next event location in a series. *Annals of GIS, 18*(3), 225–240. https://doi.org/10.1080/19475683.2012.691904

Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics, 27*(3), 832–837. https://doi.org/10.1214/aoms/1177728190

Rosser, G., Davies, T., Bowers, K. J., Johnson, S. D., & Cheng, T. (2017). Predictive crime mapping: Arbitrary grids or street networks? *Journal of Quantitative Criminology, 33*, 1983–1995. https://doi.org/10.1007/s10940-016-9321-x

Stanković, D. (2022). Crime concentration and temporal stability in spatial patterns of crime in niš, serbia. *European Journal on Criminal Policy and Research, 28*, 597–616.

Stec, A., & Klabjan, D. (2018). Forecasting crime with deep learning. arXiv:1806.01486 .

Trudeau, R. J. (1994). *Introduction to graph theory* (2nd ed., p. 224). New York, US: Dover Publications.

Wang, B., Yin, P., Bertozzi, A. L., Brantingham, J. P., Osher, S. J., & Xin, J. (2019). Deep learning for real-time crime forecasting and its ternarization. *Chinese Annals of Mathematics, 40*, 949–966. https://doi.org/10.1007/s11401-019-0168-y

Wang, H., Kifer, D., Graif, C., & Li, Z. (2016). Crime rate inference with big data. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16, New York, NY, USA, (pp. 635–644). Association for Computing Machinery.

Wang, X., Brown, D.E., & Gerber, M.S. (2012). Spatio-temporal modeling of criminal incidents using geographic, demographic, and twitter-derived information.*2012 IEEE International conference on intelligence and security informatics*, (pp. 36–41).

Weisburd, D. (2015). The law of crime concentration and the criminology of place. *Criminology, 53*(2), 133–157. https://doi.org/10.1111/1745-9125.12070

Wolsey, L. A. (2020). *Integer programming* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Publisher's Note