Received: 26 July 2022 | Revised: 7 March 2023 | Accepted: 8 March 2023

DOI: 10.1111/tgis.13050

RESEARCH ARTICLE

Transactions in GIS WILEY

Is your neighbor your friend? Scan methods for spatial social network hotspot detection

Xiaofan Liang[1] | Joshua Baker[2] | Daniel DellaPosta[3] | Clio Andris[1,2]

[1]School of City and Regional Planning, Georgia Institute of Technology, Atlanta, Georgia 30313, USA

[2]College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30313, USA

[3]Department of Sociology and Criminology, Pennsylvania State University, University Park, Pennsylvania 16801, USA

Correspondence
Clio Andris, School of City and Regional Planning, Georgia Institute of Technology, Atlanta, GA 30313, USA.
Email: clio@gatech.edu

Funding information
National Science Foundation

Abstract

GIS analyses use moving window methods and hotspot detection to identify point patterns within a given area. Such methods can detect clusters of point events such as crime or disease incidences. Yet, these methods do not account for *connections* between entities, and thus, areas with relatively sparse event concentrations but high network connectivity may go undetected. We develop two scan methods (i.e., moving window or focal processes), EdgeScan and NDScan, for detecting local spatial-social connections. These methods capture edges and network density, respectively, for each node in a given focal area. We apply methods to a social network of Mafia members in New York City in the 1960s and to a 2019 spatial network of home-to-restaurant visits in Atlanta, Georgia. These methods successfully capture focal areas where Mafia members are highly connected and where restaurant visitors are highly local; these results differ from those derived using traditional spatial hotspot analysis using the Getis–Ord Gi* statistic. Finally, we describe how these methods can be adapted to weighted, directed, and bipartite networks and suggest future improvements.

1 | INTRODUCTION

In spatial statistical analysis, hotspot or cluster detection can find statistically significant concentrations of an event or phenomenon, such as incidences of cancer, locations of similar tree species, or clusters of restaurants across a city. Traditional point pattern analysis methods, such as Getis–Ord GI and Gi* statistics or Ripley's K-function, are often used to find these clusters. For example, given the locations of individuals who belong to a

specific religious group, we can infer whether individuals cluster, whether they live in certain neighborhoods, and how far they live from the institution. These results can lead to useful findings on what may attract individuals to certain locations, whether there are benefits of living in proximity, and what resources a group may have access to.

Here, we extend point pattern analysis to capture theoretical *propinquity* effects, wherein proximal people are more likely to form a connection (see Bossard, 1932; Fischer, 1982). For instance, we may want to find groups of species that not only live in a place with similar ecological properties, but also signal to each other; individuals who not only live nearby, but also call one another; or organizations that exchange ideas and technologies, in addition to simply locating in the same office park.

To detect social networks of *connected* entities, we require methods that can capture areas where sets of points are both clustered in geographic proximity and connected in network space. Here, we develop two methods for detecting not just clusters of points, but clusters of connected points (local networks) for nonplanar networks (e.g., telecommunications flows, points of interest visits, and social networks). We present two scan methods (i.e., moving window/focal processes), *EdgeScan* and *NDScan* (Network Density Scan), which detect the number of nonplanar edges (or sum of edge weights in a weighted graph) and network density, respectively, of a network contained within a focal window. We also conduct sensitivity tests by varying the window size (in kilometers) and distance metric: Euclidean distance, Manhattan distance, walking distance, and K-nearest-neighbor (KNN) neighborhoods.

The objective of this research is to detect locations where dense networks occur. We conduct case studies of social connections between geolocated members of the Mafia in New York City and a case study of points of interest (POIs) visits between the visitors' home census block group and destination restaurants in Atlanta, Georgia. A focal research objective for the Mafia case study is to discover if and where Mafia members have local social connections. A parallel research objective for the Atlanta POI case study is to detect if and where restaurants serve (primarily) local residents.

We find significant clusters of connected Mafia networks in areas such as Dyker Heights, Bushwick, and Flushing neighborhoods of New York City that are not detected using a traditional spatial hotspot detection method, Getis–Ord Gi*. Regarding visits to restaurants in Atlanta, areas around Midtown and Downtown have many clustered restaurants, but travelers do not form network clusters; conversely, neighborhoods, such as East Chastain Park, have dense connections, despite low point density.

Our findings show that areas with notable social interaction may go undetected with traditional analysis methods. More broadly, adding a connectivity metric to typical point pattern hotspot detection helps disambiguate dense social–spatial networks from (disconnected) point clusters. In addition, this method results in scalar numbers at the node level, and thus, nodes can be ranked, compared, and symbolized by these values on a map, leading to more facile visualization of spatial social networks (SSNs) (an emergent issue in GIScience (Giordano et al., 2022; Sarkar & Yadav, 2021)). EdgeScan and NDScan methods are shared as a R package and an online tutorial (See Data Availability Statement).

This research responds to a growing need for social network analysis (SNA) tools within GISystems (Andris et al., 2018; Edwards, 2020; Sarkar et al., 2016). Location-based social network (LBSN) data analysis has helped describe spatial aspects of very large network datasets and advanced methods (examples abound), and the research described here helps leverage more canonical GIS techniques to help researchers learn about localized systems of human connectivity.

This manuscript proceeds as follows. We next review related work on hotspot detection broadly and related to networks and social networks. We then describe the data and data processing steps for our case studies and outline the EdgeScan and NDScan algorithms. Following, we report on descriptive analysis results and the results of our sensitivity analysis, and discuss our findings and conclusions.

2 | RELATED WORK

2.1 | Hotspot detection

In GIS, hotspot analysis is used to capture spatially concentrated events, such as crimes (Chainey et al., 2008), species (Hurlbert & Jetz, 2007), and diseases (Das et al., 2021). There are many techniques that can reveal hotspots in point patterns. Density-based methods identify hotspots by detecting a high concentration of events in an area. For example, moving window methods, such as local K and local L statistics derived from Ripley's K function (Getis & Franklin, 1987; Ripley, 1976), create a fixed-distance user-defined neighborhood and go through every spatial unit (i.e., point, grid, or census unit) to report the density statistic and significance. The statistic is significant if it exceeds the expected density generated based on the complete spatial randomness (CSR) assumption (Baddeley et al., 2015). Other spatial scan methods count events within the moving window as the radius increases toward a threshold; this threshold might be a given number of events (Besag & Newell, 1991) or population (Kulldorff & Nagarwalla, 1995). In contrast, kernel density estimation (KDE) is not constrained by a neighborhood: it generates a smooth surface of predicted density based on Gaussian distribution (Silverman, 1998).

Correlation-based methods (or similarity-based methods) identify hotspots through spatial autocorrelation. For example, local Moran's *I* calculates the degree of which the value of the spatial unit is similar to its neighbors (Anselin, 1995). Similarly, the Getis–Ord Gi* statistic indicates where high values cluster: values are high if the weighted average of neighbors (including the focal event) is close to the sum of all values (Getis & Ord, 1992; Ord & Getis, 1995). Low average nearest-neighbor distances can also indicate a cluster (Cressie, 2015). More recently, machine learning based clustering algorithms have been used to capture hotspots (i.e., clusters) with more diverse shapes. For example, Density-based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised clustering that first omits areas with noise (i.e., low density of events) and then scans through each event (i.e., focal event) in a user-defined window (Ester et al., 1996), while events are a user-defined threshold. In terms of visualization, hotspots can be mapped by values through thematic mapping or be represented by generalized shapes such as KDE surface or spatial ellipses (Chainey et al., 2008; Maciejewski et al., 2009).

2.2 | Hotspot detection on networks

Spatial network hotspot detection is a newer technique that accounts for point distribution along networks embedded on a plane (Ishioka et al., 2019; Okabe et al., 2006; Okabe & Yamada, 2001). These methods improve upon the circular neighborhoods used by the traditional scan methods by concentrating the potential area of point events to the network's geometry (e.g., omitting areas where traffic accidents do not occur) (Shiode & Shiode, 2013). Network (or cost) distance is used instead of Euclidean distance to create a more realistic model of point dispersion and distance between points (Borruso, 2008; Shiode, 2011). As such, nearby points that are not connected in the networks may not be counted together. The visualization of network-based hotspots is also different, as the values can be visualized on lines (e.g., road networks) rather than point clouds (Oliver, 2016). Yet, these studies still focus on point distributions.

More recent work has used spatial scan statistics to analyze movement and flows on networks (Gao et al., 2018), Moran's *I* (Liu et al., 2015), Ripley's K (Ripley, 1976), and density-based clustering (Shivanasab et al., 2021; Tao & Thill, 2016). In these studies, one or many of the following characteristics, such as density (or volume), the proximity of the origins and destinations, the magnitude, direction, shape, and even the timestamp of flows, are used to identify representative flow patterns in a haystack of data. Consequently, the outputs find outstanding global patterns or clusters that are independent of scale (in distance), but cannot specify local insights or generate hotspot statistics.

2.3 | Hotspot detection on spatial social networks

Few studies have examined *hotspots* of SSNs. We define a SSN as a specific type of network where the nodes are geolocated and edges are nonplanar, representing social connections between individuals or places. Prior work on creating heat maps of point-to-point migration flows across England has captured nonplanar network density by counting the number of edges that cross a grid cell and creating a raster surface of results (Rae, 2009, 2011). This method is subject to fly-over edges that may not be pertinent to research questions on local connections. Hotspots in social networks can also be detected by calculating the sum of degrees as a scan statistic (Wang & Phoa, 2016) but this is typically conducted without a geographic context. A more recent study on social connections characterizes SSN hotspots as the percentage of neighbors who are also connected within walking distance, thus solely focusing on the short-range ties (Andris et al., 2021). However, this study did not test how results might change with window size and definition such as K-nearest neighbors.

2.4 | Community detection in networks

Community detection (also known as network partitioning) algorithms are used to detect parts of a network with interconnected nodes using divisive, agglomerative, or hierarchical processes (Fortunato & Hric, 2016). One popular set of community detection methods is referred to as "modularity" algorithms (Newman, 2006). When nodes have geolocation information, each node is assigned a community, and then the node is subsequently mapped onto geographic space to show regions where communities are located, and how far they spread (as in Comber et al. (2012) and Hu et al. (2018)). These methods have been used on social networks: a cell phone call network in Jiamusi, China showed that personal network communities can be geographically tight or can span the whole city (Wang et al., 2019).

Community detection has been used to divide geographic space into contiguous regions based on origin–destination or adjacency matrices, extending decades of past research uncovering "functional" geographic regions (Masser & Scheurwater, 1980). An agglomerative partitioning method using a minimum spanning tree was able to create U.S. regions with homogeneous presidential election outcomes (Guo, 2008). Twitter ties have been used to detect functional boundaries of cities, and in one case study showed that social communities tend to follow Borough boundaries in New York City (Brelsford et al., 2019). At the neighborhood level, community detection was used to find isolated and well-connected neighborhoods in Milwaukee, Wisconsin (Prestby et al., 2020). Furthermore, community detection methods have been applied to social media data from Twitter and Facebook to show the locations of cohesive social communities and culture hearths (Bailey et al., 2018; Koylu, 2018).

Our approach differs from community detection methods in two fundamental ways. First, the detection approach presented here depends on node *location*, while community detection methods often divide the (aspatial) network and then map the nodes based on their resultant communities. Second, community detection methods often partition the entire network into subgraphs, where each node belongs to a partition (i.e., group, cluster), while in the methods presented here, not all nodes participate in a cluster.

3 | METHODS

3.1 | Network detection moving window (scan) methods

We first describe the basic concepts and structures that underlie EdgeScan and NDScan (i.e., Network Density Scan), based on moving window scan techniques. The goal of the two methods is to detect places where there are incidences of connected nodes. Both methods input a set of nodes and edges, a moving window neighborhood

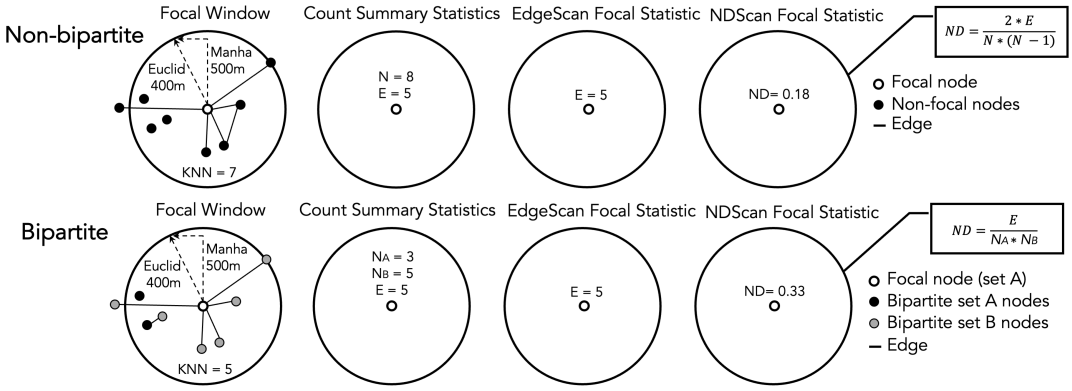

FIGURE 1 EdgeScan and NDScan statistics from the focal node in a bipartite and non-bipartite network. The example contains nine nodes and six edges where only eight and five of each are within the focal window. The focal window simultaneously represents a neighborhood definition of 400 m in Euclidean distance, 500 m in Manhattan distance, and 7-Nearest neighbors (5-Nearest neighbors for the bipartite network). A typical scan method would produce a EdgeScan or NDScan statistic, assigned to the central node. *N* is the number of nodes, *E* is the number of edges, *ND* is network density, N_A is the number of nodes in set *A* of a bipartite network, and N_B is the number of nodes in set *B*.

size, and include optional parameters, such as distinctions as weighted and bipartite, a user-defined distance matrix, or a threshold value. Like in scan statistics for general hotspot detection within point patterns, these algorithms select each individual node and scan its neighborhood (or focal area) systematically.

We define a *neighborhood* as a circle with *K* neighbors (e.g., *K*=15) including the focal node and its *K* (e.g., 15) closest neighbors, a circle with a radius *R* (e.g., a focal node and all other nodes within 1-km buffer from the focal node) (Kulldorff & Nagarwalla, 1995), or a user-defined distance threshold. *R* can be measured using Euclidean distance, Manhattan distance, or travel distance based on a road network, or any allometric distance included as a user-defined matrix. As in raster computing practices, a neighborhood can also be defined as a specific number of cells (Mu & Holloway, 2019), but since networks are not typically represented as rasters, we do not consider this definition here.

Both methods output two tables: a node table containing the scan statistics of the neighborhoods for each node and an edge table indicating whether the edge is within the window size for each edge. Regarding special considerations, social networks and networks of social flows can have properties such as edge weights and direction, and nodes can bipartite (i.e., two sets of nodes that only connect with those in the alternate set). Each method accounts for these differently (see Figure 1).

Both EdgeScan and NDScan resultant statistics are tabulated and visualized at the node level. It is also possible to use the above methods to produce a statistical surface (e.g., a raster grid) of values, as many scan statistics produce such continuous raster grids from points such as elevation or rainfall. Regarding visualization, this outcome prevents a haystack visualization problem—a common challenge for nonplanar network visualization in GISystems (Andris et al., 2018).

3.1.1 | EdgeScan

EdgeScan marks each focal node with the number of edges that have both endpoints within a focal area, regardless of whether the edges are connected to the focal node or not. As such, the algorithm produces nodes with values that detect concentrated sets of nearby edges as opposed to just concentrated sets of points. The focal area is determined by the search window neighborhood definition used.

EdgeScan can be applied to weighted, bipartite, and/or directed networks. For weighted networks, the EdgeScan algorithm will report the sum of edge weights for each node instead of the number of edges. Edge weights are defined in the *Weight* column of the input edge data table. For bipartite networks, EdgeScan only reports values for nodes that are in set *A*, indicated by a binary column *Bipartite* in the input node data table *N*. The constraint of the minimal cluster size (defined by *Min* parameter) will also only apply to nodes in set *A*. For directed networks, the user should process each directed edge as a single edge (e.g., a mutual friendship between person X and person Y will should two edges: X to Y and Y to X). As such, the outcome can simply be interpreted as the number of directed edges within the scan neighborhood without further adjustments.

ALGORITHM 1: EdgeScan

```
Input: N for a node table (a list of lists), E for an edge table (a list of lists), W for window size (One of: R
  radius of search window, M Manhattan distance of search window, waking distance matrix, or K number of
  nearest neighbors in search window), Min (Optional: minimum number of points in the window),
  Weighted (Optional: whether edges are weighted), Bipartite (Optional: whether the network is bipartite)
output = EdgeScan(N, E, W, Min, Weighted, Bipartite)
if Bipartite then
 | Reduce N to node set A that will receive heat value
end
forall Node n in N do
    if Bipartite then
     | Obtain subset N′ of nodes from set B within n's window size
    else
     | Obtain subset N′ of nodes within n's window size
    end
end
if N′ < Min then
 | heat(n) = NA
else
 | heat(n) = {# of edges or sum of edge weights with both endpoints in the window.}
end
forall Edge e in E do
    if the euclidean length of e is within n's window size then
     | edgeWithin(e) = 1
    else
     | edgeWithin(e) = 0
    end
end
return list(heat, edgeWithin)
```

3.1.2 | NDScan

NDScan searches for groups of well-connected nodes within a neighborhood. It computes the *network density ND*, which is calculated as the number of intra-node edges within the given neighborhood to the number of possible edges, given the same number of nodes (i.e., the number of edges required to connect all nodes to each other, forming a clique) (Equation 1). If the focal area has a higher network density *D* than the entire network, we consider that focal region to be specially connected. Similar to EdgeScan, three definitions of neighborhoods are available and a parameter to control cluster size is optional for distance metrics in the scan methods.

NDScan can be used on bipartite and directed networks but not weighted networks, due to the definition of network density: In a directed network, network density *ND′* accounts for twice of the potential connections than those in an undirected network (Equation 1).

In a (undirected) bipartite network, the potential connections in network density *NDb* are all possible connections between nodes in set *A* and set *B*. If the bipartite network is also directed, the potential connections in network density *NDb′* will double (Equation 2).

$$ND = \frac{2E_i}{N_i(N_i - 1)} \quad ND' = \frac{E_i}{N_i(N_i - 1)} \tag{1}$$

$$NDb = \frac{E_i}{N_{A_i} \times N_{B_i}} \quad NDb' = \frac{E_i}{N_{A_i} \times N_{B_i} \times 2} \tag{2}$$

where ND is network density in an undirected network, ND' is network density in a directed network, NDb is network density in an undirected bipartite network, NDb' is network density in a directed bipartite network, E_i and N_i are the numbers of edges and nodes, and N_{A_i} and N_{B_i} are the number of edges and nodes in set A and B, respectively, in region i.

ALGORITHM 2: NDScan

```
Input: N for a node table, E for an edge table, W for window size (One of: R radius of the search window, M
 Manhattan distance of search window, or K number of nearest neighbors in the search window), Min
 (Optional: minimum number of points in the window), Directed (Optional: whether edges are directed),
 Bipartite (Optional: whether the network is bipartite)
output = NDScan(N, E, W, Min, Directed, Bipartite)
if Bipartite then
    Reduce N to node set A that will receive heat value
end
forall Node n in N do
    if Bipartite then
        Obtain subset N′ of nodes from set B within n's window size.
        Obtain subset N″ of nodes from set A within n's window size.
        if Directed then
            potential = N′ * N″ * 2
        else
            potential = N′ * N″
        end
    else
        Obtain subset N′ of nodes within n's window size.
        if Directed then
            potential = N′ * (N′-1)
        else
            potential =N′ * (N′-1) / 2
        end
    end
    numEdges = {# of edges between with both endpoints in the window.}
    if N′ < Min then
        heat(n) = NA
    else
        heat(n) = numEdges / potential
    end
end
forall Edge e in E do
    if the euclidean length of e is within any n's window size then
        edgeWithin(e) = 1
    else
        edgeWithin(e) = 0
    end
end
return list(heat, edgeWithin)
```

3.2 | Case study applications

We test the EdgeScan and NDScan methods on two case studies: a 1960s Mafia network in New York City and a 2019 POI visit network in City of Atlanta.

The nodes in the Mafia network represent home addresses of the mafia members and the edges represent criminal associations, as documented by the US Federal Bureau of Narcotics (DellaPosta, 2017). Mafia families are criminal organizations whose members engage in a variety of illegal (e.g., extortion, illegal gambling, drug trafficking) and sometimes legal (e.g., some members owned bars or restaurants) economic activities or "rackets." While there are many varieties of organized crime, US Mafia families specifically include groups that descended from similar groups in southern Italy and restricted full membership to ethnic Italians (though non-Italians could still be "associates" of the family and we include such individuals). As described in qualitative and historical accounts of the US Mafia (Abadinsky, 1983; Gambetta, 1993), a member's network of connections was perhaps their most important resource for carrying out criminal activities. For this reason, it is important to not just document clusters of co-located nodes but also the face-to-face connections between them that facilitated collaboration in risky and dangerous criminal activities.

The Mafia network is an unweighted, non-bipartite social network. We limit our case study to New York City's five boroughs, as the network is concentrated in this location. The data include 298 nodes and 946 edges, yielding a density of 0.021 and an average degree of 6.35. The average distance of an edge is 12.97 km, and the standard deviation is 10.55 (max = 55.3 km). There are two edges with 0 distance, as these individuals live at the same address (a home or apartment building). Triads and isolates both emerge, as well as a few examples of "cliques."

The Atlanta POI visit network is a weighted, bipartite spatial network sourced from SafeGraph Inc. SafeGraph's origin–destination data are aggregated from GPS pings of mobile phone applications and represent 10% devices in the United States. While the data capture mobility at an aggregated scale (Kang et al., 2020), they underrepresent non-White and older populations (Coston et al., 2021). We sum the visitor counts from a census block group (origin) to any POI (destination) from March to June 2019. Each row of the data contains a POI, POI coordinates, the visitors' home census block groups (henceforth, "neighborhood"), and the number of visitor counts from these neighborhoods. We only use POIs in the *Restaurant and Other Eating Places* category to capture Atlanta's foodscape as a case study, as restaurants are key for generating foot traffic and economic vitality in cities (Credit & Mack, 2019). We keep POIs with more than three visits, and those that connect to more than one neighborhood. Then, we calculate the percentage of visits from a neighborhood to a restaurant out of the total visits to this restaurant as the edge weights and keep edges with edge weights greater than 4%.

The nodes in our final POI visit networks are restaurants and centroids of neighborhoods, and the edges are the percentage of visits from neighborhoods to restaurants. The data include 1356 nodes (1145 restaurants and 311 neighborhoods) and 7926 edges, an average degree weight of 8.6% with a standard deviation of 6.5%. On average, a restaurant connects to eight neighborhoods with a standard deviation of five. Identifying areas with restaurants that serve residents from nearby neighborhoods may reveal places that cultivate local culture and place-making, or in the opposite, reveal destinations whose clients are from distant parts of the city.

To test whether our networks have more local ties than expected, we applied a two-sample Kolmogorov–Smirnov test to capture the difference between the distribution of existing edge distances (i.e., Euclidean distance between two connected nodes) and the distribution of all possible edge distances (i.e., Euclidean distance between fully connected nodes). Both networks return statistically significant p-values (<0.0001), indicating that they are likely to have locally concentrated ties.

3.3 | Comparison with Getis–Ord Gi* hotspot detection

We compare EdgeScan and NDScan values to results from a classic node density detection method, Getis–Ord Gi*, at the node level, to examine whether our methods can reveal different insights. We divide the New York City

study area into a 1000×1000 m grid and the Atlanta study area into a 500×500 m grid. Since the Getis–Ord Gi* values may be sensitive to the grid size, we experiment on a few grid sizes and still reach the same conclusions. The final grid sizes are chosen relative to the scale of the study areas and are big enough to not be covered by the nodes in the map. We perform the Getis–Ord Gi* analysis for each grid in both study areas and visualize nodes with high EdgeScan or NDScan values (red and dark red nodes) and Gi* hotspots that are above 90% confidence (red grids). We quantify the number of detected SSN hotspots (applying the DBSCAN algorithm using EdgeScan and NDScan values) and the number of Getis–Ord Gi* hotspots.

3.4 | Sensitivity test

We perform two sensitivity tests. First, We compare mean and standard deviation values of EdgeScan and NDScan for mafia and POI networks using window sizes of 0.5, 1, and 2 km Euclidean, Manhattan, and road-based (henceforth, "walking") distance and 5, 10, and 20 K-nearest neighbors. Walking distance is calculated using OpenStreetMap data for 2023 through the *osrm* package in the R statistical computing environment. We also report the number of nodes with at least two neighbors (i.e., min = 3) captured by various window sizes. Second, we plot how average EdgeScan and NDScan values change for each node cluster to explore an optimal window size. We identify node clusters using the DBSCAN algorithm. For the New York case study, we assign the DBSCAN algorithm with the epsilon values (i.e., the maximum distance between two points while still belonging to the same cluster) with the corresponding window size and a minimum of three points. For the Atlanta data, we assign the DBSCAN algorithm's epsilon value with a fixed distance of 1000 m and also a minimum of three points. We use KNN in Atlanta because the absolute distance between restaurants is much larger for peri-urban restaurants than urban restaurants, and KNN allows for this relative distance.

3.5 | Implementation

Code development, computation, and visualization are all conducted in the R statistical computing environment, using the following packages: *tidyverse*, *ggplot2*, *tmap*, *sf*, *NbClust*, *cluster*, *basemaps*, *spdep*, *raster*, *grid*, and *ggdist*. New York City data are projected to the NAD 83 New York Long Island Coordinate System and the City of Atlanta data are projected to NAD 83 State Plane Georgia West coordinate system.

4 | RESULTS

We apply EdgeScan and NDScan methods on data showing social connections between members of the Mafia in New York and data showing visits from visitors' neighborhoods to restaurants. First, we explore the results of EdgeScan and NDScan under a set of chosen parameters (i.e., Euclidean distance = 1 km for the Mafia network and KNN = 5 for the POI network). Next, we compare network hotspots derived from the EdgeScan and NDScan results with spatial hotspots identified by the Getis–Ord Gi* statistics. Finally, we examine the sensitivity of EdgeScan and NDScan results under varying window sizes and neighborhood definitions and suggest potentially optimal window sizes.

4.1 | EdgeScan and NDScan results

We visualize scan outcomes for the New York Mafia network with a neighborhood of 1-km Euclidean distance (with at least three nodes), as this is a reasonable walking distance for meeting (Yang & Diez-Roux, 2012). Connections

at this range suggest the possibility of face-to-face meetings. On the contrary, distance in the Atlanta POI visit network is not always a meaningful neighborhood definition, because restaurant density can decrease with distance from the city center. Therefore, we use 5-nearest neighbors as the scan neighborhood definition.

Insets of the EdgeScan and NDScan maps show the tight concentrations of edges and/or high network density (nodes with dark red color). In New York, these areas include East Bronx (Inset a), West Bronx (Inset b), East Harlem (Inset c), Little Italy (Inset d), Dyker Heights (Inset e), Bushwick (Inset f), and Flushing (Inset g) (Figure 2). These areas, such as Little Italy and East Harlem (also referred to as Upper East Side), are known sites of historic Mafia activities (Maas, 1969). Dyker Heights is home to the Profaci family, a Mafia family that has the most densely clustered member distribution (Andris et al., 2021). Most local connections in this area are ties between members of this family, including a few high-degree members. Similarly, West Bronx and East Bronx's local social connections belong to the Gambino family; Flushing's local ties are between members of the Lucchese family; Bushwick's triad members are from the Genovese family.

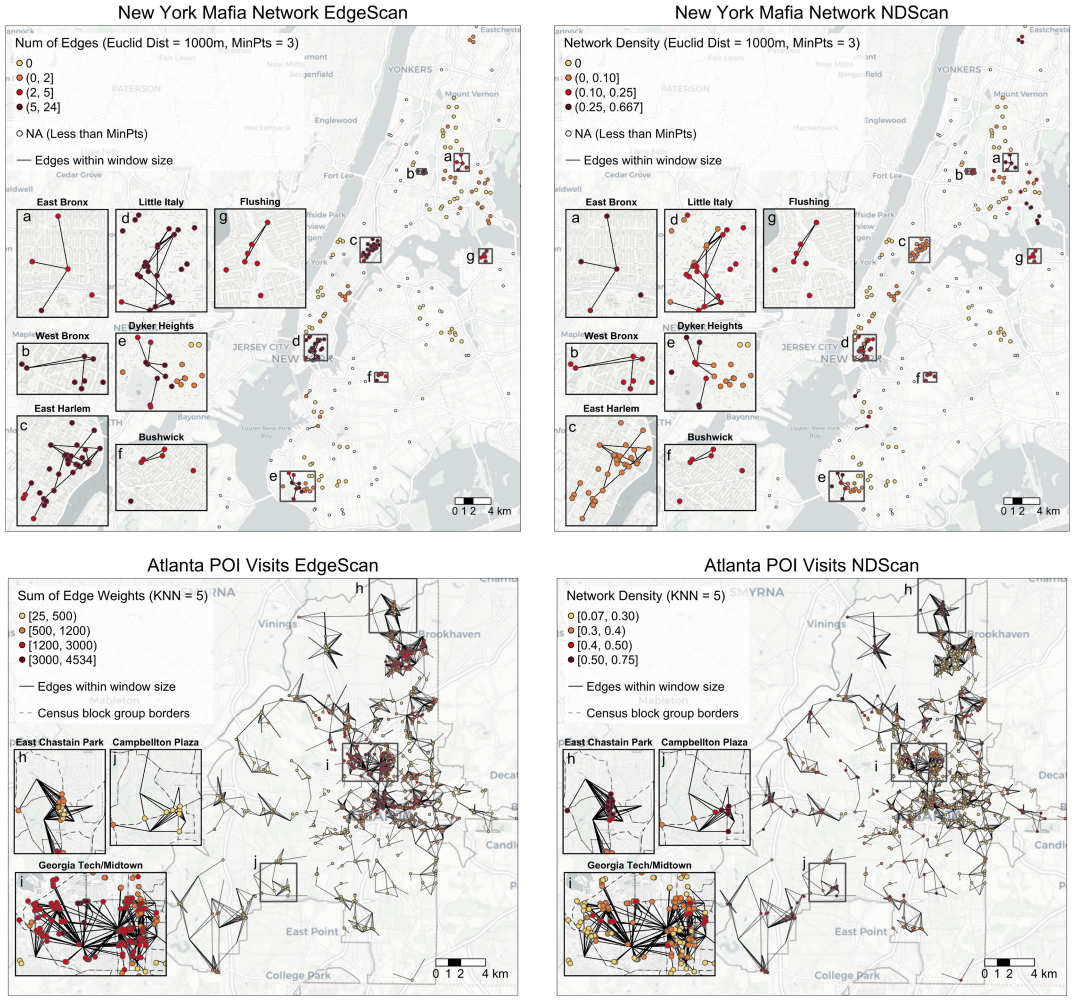

FIGURE 2 Maps of EdgeScan and NDScan results for New York Mafia social networks (unweighted non-bipartite) and Atlanta point of interest (POI) visits (weighted and bipartite). Both networks are undirected. The New York case study assigns values to each Mafia member and uses Euclidean distance, while the Atlanta case study only assigns values to POIs and uses K-nearest neighbors as the window size. Only edges within the window size are visualized. Insets highlight areas of interest.

In Atlanta, some example areas with high concentration of customers coming from nearby neighborhoods include East Chastain Park (inset h), Georgia Tech and Midtown (inset i), and Campbellton Plaza (inset j) (Figure 2). East Chastain Park and Campbellton Plaza are parts of the North and Southwest Atlanta suburbs, respectively. Georgia Tech is a public university with a large population of resident students and Midtown is a high-density, walkable, mixed-used area for working professionals. Both areas' populations rely on not only nearby restaurants, but also host tourists and visitors, which can explain the concentration of local clientele.

While our results highlight areas with tight local spatial or social connections, the most "connected" areas differ between EdgeScan and NDScan methods (Figure 2). For example, in New York, East Harlem has many edges (EdgeScan) within a 1-km neighborhood, while the network density (NDScan) value is medium-low. This difference is expected because the EdgeScan definition ignores the number of nodes in the neighborhood and can increase in areas with more nodes, while NDScan is sensitive to the number of nodes in the neighborhood. The low network density values in East Harlem indicate that despite the spatial clustering of nodes and higher-than-average connections in the neighborhood, Mafia members live near each other, but are disconnected. This may be due to three Mafia families (Genovese, Gambino, and Lucchese) co-locating in the neighborhood (Andris et al., 2021), who connect with distant family members instead of nearby nodes.

Similarly, in Atlanta, EdgeScan values are also biased toward areas with many restaurants, while NDScan values can be high for small clusters. Georgia Tech and Midtown (Figure 2) have relatively high regional EdgeScan values, which is likely due to a high concentration of restaurants in the area. Yet, some spots in the area have relatively low NDScan results, indicating that some restaurants may serve distant and local customers alike. On the contrary, suburban areas such as East Chastain Park and Campbellton Plaza have low EdgeScan values due to small cluster size, but most restaurants serve visitors from each adjacent neighborhood.

Besides revealing the hotspots of local spatial or social ties, we find cold spots of ties where nodes have zero values in EdgeScan or NDScan results despite co-location of nodes. For example, in New York City, some dense clusters and dispersed nodes have no connections (denoted by yellow nodes) with their neighbors, perhaps because the window size is too small to capture connections, or simply because they do not interact with each other. Even with an expanded window size (KNN = 10), two nodes still have zero local edges and network density. One such node is in the center of the point pattern, perhaps indicating that it may be between two separate groups, although this node does not have connections to either side (i.e., it is not playing a broker role). The second node with zero values for both EdgeScan and NDScan lies north of a dense network of the Profaci Family members who cluster in the Dyker Heights area (inset e). The zero value for this node indicates that while the node is near other Mafia members, these members are not connected.

4.2 | Comparison with traditional hotspot detection Getis–Ord Gi*

The SSN scan methods detect both similar and different hotspots than the traditional statistics, and this difference is more pronounced with NDScan (which can detect small and well-connected clusters) than EdgeScan. Here, we consider SSN hotspots as geographically clustered nodes with high EdgeScan or NDScan values. Areas where nodes form hotspots based on Gi* statistic overlap with some but not all the areas detected by the EdgeScan or NDScan method (Figure 3). For example, in New York City, DBSCAN algorithm detected seven SSN hotspots with at least two edges in EdgeScan and nine SSN hotspots with more than 0.1 network density in NDScan, while Getis–Ord Gi* reported only two 90% confidence hotspots and both hotspots overlap with SSN hotspots in EdgeScan and NDScan. Yet, this numeric comparison is sensitive to the cutoff values of a SSN hotspot, and in our maps, such value has been chosen carefully with sensitivity tests. Thus, we focus on a qualitative comparison of hotspots' location differences instead.

In New York City, East Harlem (only in EdgeScan) and Little Italy contain clusters of nodes that not only co-locate in geographic proximity but also connect intensively via social interactions. However, Dyker Heights, East

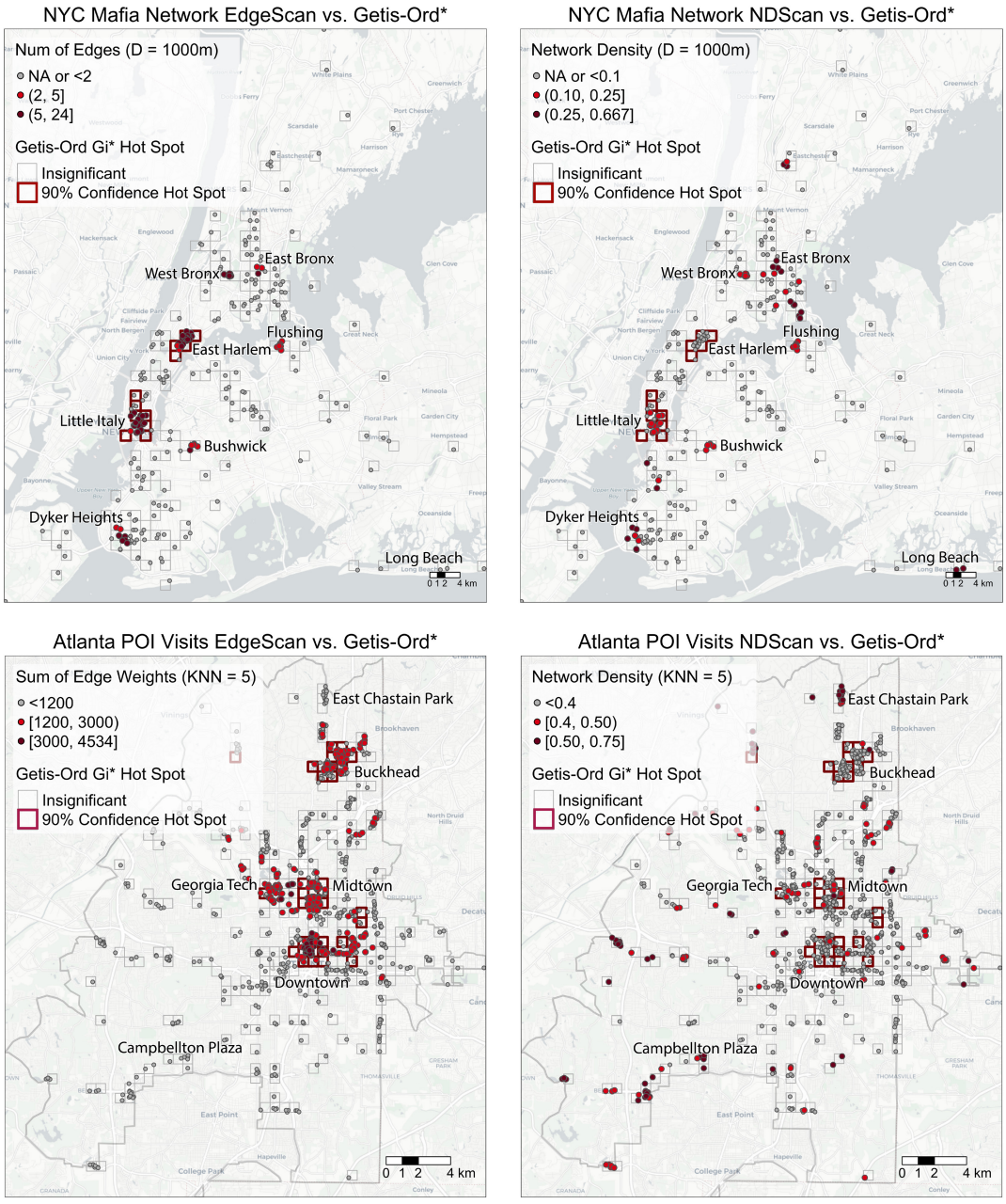

FIGURE 3 Maps of EdgeScan and NDScan results versus Getis–Ord Gi* statistics for New York and Atlanta. The EdgeScan and NDScan results are generated with the same set of parameters in the previous figure on the node level, while the Getis–Ord Gi* statistics are generated using grid cells.

Bronx, West Bronx, Flushing, East Harlem (only in NDScan), and Bushwick are not statistically significant in the traditional hotspot analysis, despite high value nodes in EdgeScan and NDScan results. This difference implies that our methods can find small groups of nodes that are well-connected but do not qualify as spatial clusters. These small clusters can be important if contextualized with other node attributes. For instance, Long Beach has two isolated yet interconnected Mafia members; one is John Ormento, the capo (captain of the organization) of the Lucchese family and the highest degree member in the Mafia network. In Atlanta, the EdgeScan hotspots closely

resemble the Gi* hotspots, overlapping at Buckhead, Georgia Tech, Midtown, and Downtown Atlanta. These locations are popular live-work-play destinations. However, the NDScan hotspots are more spatially dispersed, located in both the city center and the city periphery. Many NDScan hotspots are composed of only a few restaurants, including our examples of East Chastain Park and Campbellton Plaza, and are not significant spatial clusters, but have noteworthy local connections.

4.3 | Result sensitivity by neighborhood definition and window size

We examine how EdgeScan and NDScan values for the entire network and for each node cluster change by neighborhood definition and window size. We first report the mean, standard deviation, and the percentage of nodes with values (i.e., with more than three nodes in the scanning window) under various window sizes of Euclidean distance, Manhattan distance, walking distance, and KNN neighborhoods.

The average number of detected nonplanar edges per node for New York City ranges from 2.49 to 33.14 and scales with neighborhood size (Table 1). Because EdgeScan simply detects number of edges, it is more likely to increase with increased area. The smallest values are found when using Manhattan distance with a 0.5-km radius, wherein nearly 72% of nodes have no neighbors within this radius. The largest values are found when using KNN, which, unlike the scalar distance methods, increases rapidly with more neighbors. Whether the scalar distance metrics or KNN will result in higher scan statistics depends on the spatial dispersion of the nodes: a more dispersed distribution yields higher values for KNN. Similar to New York, the average EdgeScan value per node in Atlanta also grows with neighborhood size and adding neighbors to KNN can quickly lead to higher values.

The average NDScan results show that for both cities and for each neighborhood definition, network density decreases or stays the same when window size expands. The network density values and the extent of change (over window sizes) are somewhat consistent between Euclidean and Manhattan neighborhood and yet different with KNN. In New York City, network density values in KNN are slightly higher than those using distance-based neighborhoods. In Atlanta, network density values in KNN are lower and decrease more consistently than those in the distance metrics.

TABLE 1 Mean EdgeScan and NDScan results and the number of nodes that have values with MinPts = 3 by neighborhood definition.

Neighborhood	Mean (SD)	Mean (SD)	*N* (node ≥ 0)	Mean (SD)	Mean (SD)	*N* (node ≥ 0)
Definition	NYC EdgeScan	NYC NDScan	NYC nodes	ATL EdgeScan	ATL NDScan	ATL nodes
Euclidean (km = 0.5)	2.99 (3.62)	0.12 (0.16)	118 (40%)	373 (297)	0.47 (0.12)	75 (7%)
Euclidean (km = 1)	5.16 (6.88)	0.1 (0.12)	204 (68%)	955 (803)	0.37 (0.13)	816 (78%)
Euclidean (km = 2)	8.7 (9.23)	0.08 (0.09)	263 (88%)	3117 (2156)	0.21 (0.12)	1033 (99%)
Manhattan (km = 0.5)	2.49 (2.59)	0.12 (0.15)	83 (28%)	266 (180)	0.49 (0.09)	27 (12%)
Manhattan (km = 1)	3.94 (5.05)	0.11 (0.13)	172 (58%)	664 (593)	0.41 (0.13)	554 (32%)
Manhattan (km = 2)	6.96 (8.25)	0.09 (0.11)	246 (83%)	1943 (1436)	0.26 (0.12)	1018 (68%)
Walking (km = 0.5)	3.03 (2.52)	0.16 (0.21)	62 (21%)	274 (173)	0.43 (0.06)	12 (1%)
Walking (km = 1)	4.93 (5.40)	0.12 (0.15)	134 (45%)	650 (581)	0.36 (0.12)	344 (33%)
Walking (km = 2)	7.93 (8.75)	0.10 (0.10)	223 (75%)	1699 (1454)	0.24 (0.11)	975 (93%)
KNN (*K* = 5)	1.99 (1.85)	0.13 (0.12)	298 (100%)	1258 (955)	0.32 (0.10)	1045 (100%)
KNN (*K* = 10)	8.68 (4.95)	0.16 (0.09)	298 (100%)	3414 (1860)	0.16 (0.05)	1045 (100%)
KNN (*K* = 20)	33.14 (13.32)	0.16 (0.06)	298 (100%)	9523 (3903)	0.09 (0.02)	1045 (100%)

Different neighborhoods also bring different numbers of nodes into consideration. For example, the percentage of nodes with at least two neighbors (i.e., three points in the moving window, including the focal node for a non-bipartite network and excluding the focal node for a bipartite network) increases quickly with window size under Euclidean, Manhattan distance, or walking distance neighborhood definitions. Within the distance definitions, window size is the largest with Euclidean distance and smallest with walking distance. Yet for the KNN neighborhood definition, each node has a value even for the smallest window size because the moving window for a node will expand until it finds its closest neighbor(s).

Next, we ask whether there is an optimal neighborhood definition or window size for the data. Neighborhood definition should match theoretical questions, such as: Is distance between nodes meaningful; is travel time a factor in the study; or do events form natural clusters of varying radii? As such, there may not be an optimal window size, but visualizing outputs can help with decisions. We plot the sensitivity of EdgeScan and NDScan results by window sizes at the DBSCAN cluster level (see Section 3) (Figure 4), wherein consistent *y* values across the *x* axis would signal a robust method that is not sensitive to change in neighborhood size.

We look for a window size that best represents patterns of change in the sensitivity plot 4, especially node clusters with high *y* axis values. The EdgeScan box plot for New York shows two clusters with a growing number of edges with distance (labeled with arrows a and b). It suggests that a window size of 1.5 or 2 km may be optimal to capture the connections for these two clusters. Similarly, the EdgeScan box plot for Atlanta also has two clusters whose edge weight sum increases significantly faster than other clusters (labeled with arrow f). This suggests that restaurants in these two clusters serve both local and distant customers, while most other areas' EdgeScan values increase slowly with increased window size, signaling that they tend to serve locals. In this case, a window size of KNN = 8 may be an optimal window size to capture the two leading clusters.

The NDScan box plot for New York highlights three interesting distance ranges for optimal window sizes. At a window size of 100–200 m, there is a fully connected cluster of Mafia members (labeled with arrow c in Figure 4). From 400 to 2000 m, a cluster is consistent with the network density of 0.67 despite the increasing window size,

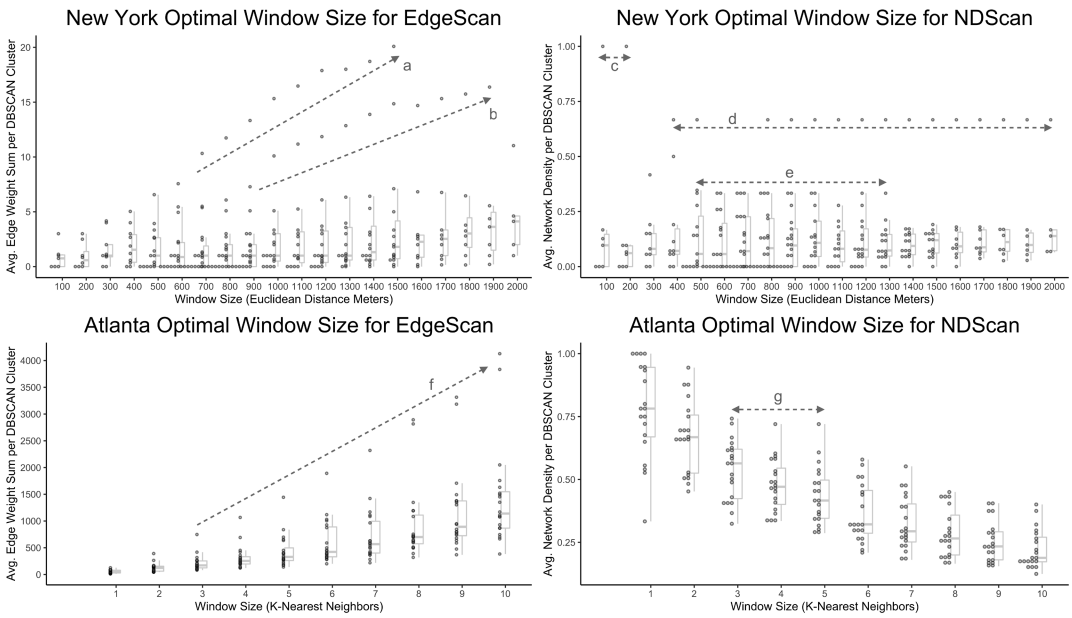

FIGURE 4 New York and Atlanta's EdgeScan and NDScan results by increasing window size (*x*-axis). Each node is a DBSCAN cluster and the *y*-axis represents the cluster's average EdgeScan or NDScan values. The box plot shows the median and the upper and lower quantiles of clusters' values at a specific window size. The labeled arrows highlight some potentially "optimal" window sizes.

suggesting that the cluster is either spatially isolated from other Mafia members or any additional Mafia member within the increased scanning window is fully connected to existing cluster members (labeled with arrow d). From 500 m to 1.3 km, there are four to five medium-high-density clusters that are stable across window sizes (labeled with arrow e). Therefore, these ranges may be optimal to reveal interesting or robust outcomes. The NDScan values for Atlanta decrease for all clusters when more proximal neighborhoods are considered. The variation is the largest across clusters when KNN is small: four clusters have restaurants that all serve the closest neighborhood (network density = 1) and one cluster has one in three restaurants that serve the closest neighborhood (network density = 0.33). This discrepancy narrows with increasing window sizes, with one or two high value clusters stabilizing briefly from KNN = 3 to KNN = 5 (label with arrow g). Thus, a KNN value between three and five may be the optimal window size for NDScan.

5 | DISCUSSION AND FUTURE WORK

In this work, we attempted to discover localized areas where a SSN has a high density of ties using two moving window methods: EdgeScan and NDScan. Using a social network of Mafia ties in New York City and a spatial network of restaurant visits in Atlanta, we detected areas with a high concentration of connections and a high local network density, and compared our results to hotspots detected by traditional spatial hotspot statistics. We further tested the sensitivity of EdgeScan and NDScan results under varying neighborhood definitions and window sizes to identify window size ranges that yield stable and robust results.

We found that the locations of hotspots using EdgeScan and NDScan were collectively different than hotspots detected by the traditional Getis–Ord Gi* method. This finding indicates that traditional detection methods are well-suited for detecting *independent* incidences of non-networked events, but when looking to distinguish *connected* (not just proximal) entities as "special," the scan methods presented here can detect hotspots of networked events. EdgeScan and NDScan are well suited for *conceptual* connections such as social network ties, as other methods exist for logistics and transportation network connectivity (e.g., Shiode, 2011). In our case study, resulting hotspots revealed where face-to-face meetings between Mafia members may have been facile, and where restaurants had a local clientele. They revealed places where coordination and local information transfer could have occurred. We also found that the results were sensitive to neighborhood definitions and window sizes. Compared with distance metrics, KNN methods tended to yield higher values, resulting in more distributed SSN clusters. The sensitivity is more pronounced with different window sizes. Some highly connected clusters may only be captured in a distance range and the network structures can change significantly when different distance segments are used.

Our study has some drawbacks. First, the EdgeScan and NDScan value of a node does not indicate whether the node itself is integrated in the local SSN. For example, a focal node bordering a dense local social network cluster can have high EdgeScan and NDScan values despite having no connections to its neighbors. Thus, the EdgeScan and NDScan values at the node level can only be said to reflect the conditions in the node's focal area. If needed, users can query the output table to compute how many edges (or the sum of edge weights) are associated with individual nodes, and re-compute network density for nodes with edges within the window.

Second, our SSN hotspots cannot generate statistical significance like the *Z* scores in traditional spatial hotspot method Getis–Ord Gi* statistics. A future solution may be to spatially permute the network to create different values for expected number of geographic ties in each location, and compare expected to actual number of ties. We can stochastically randomly reassign *n* edges to two individuals, where *n* is drawn from the network's degree distribution. The rewiring method can be improved by including wiring probabilities that decay with distance (Anderson & Dragićević, 2020) or by maintaining the distance distribution of the edges. This approach will allow us to measure the probability of our actual network configuration, given the point pattern.

Next, these methods focus on local ties and do not capture distant ties. Future work can detect nodes that belong to distant groups; that is, nodes that are "out of place." For instance, a newcomer to an area may have connections "back home" that are clustered but distant, and as such, could be marked as notable. These methods also only focus on connections and graph structures and do not capture the nature of interactions. In the future, users can also detect network triads, broker configurations, edge types, and internal versus external group interactions that are of interest to the social network community.

As implemented, these scan methods do not account for geographic features that may alter the results. For instance, a study of relationship connections in New York City may consider whether some natural or built environment features, such as rivers, highways, parks, or other walkable facilities, may hinder or facilitate the concentration of local ties. In addition, road network distances were calculated using road network data from 2023 and thus do not reflect the actual road network that would have been used by members in the dataset during the 1960s. Finally, with social network connections, a connection does not always imply that the shared local space of this relationship is used in the relationship; Two connected individuals may only relate to one another in the virtual world, on paper, or through telecommunications. As such, propinquity may not manifest itself in actual face-to-face meetings or in common identification with and involvement in local activities. However, it implies that nearness could facilitate these interactions.

In conclusion, the methods described here identify SSN hotspots and provide different insights than analysis methods that solely focus on the clustering of (disconnected) points. These methods can help researchers capture interactions between entities in geographic space.

ACKNOWLEDGMENTS

The authors would like to thank SafeGraph for the datasets provided for this research. This research was supported by NSF-BCS-2045271.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Replication Data & Code for "Is Your Neighbor Your Friend?" at http://doi.org/10.6084/m9.figshare.22179862.

ORCID


Clio Andris https://orcid.org/0000-0002-8559-5079

REFERENCES

Abadinsky, H. (1983). *The criminal elite*. Greenwood Press.

Anderson, T., & Dragićević, S. (2020). Representing complex evolving spatial networks: Geographic network automata. *ISPRS International Journal of Geo-Information*, *9*, 270. https://doi.org/10.3390/ijgi9040270

Andris, C., DellaPosta, D., Freelin, B. N., Zhu, X., Hinger, B., & Chen, H. (2021). To racketeer among neighbors: Spatial features of criminal collaboration in the American Mafia. *International Journal of Geographical Information Science*, *35*, 2463–2488. https://doi.org/10.1080/13658816.2021.1884869

Andris, C., Liu, X., & Ferreira, J. (2018). Challenges for social flows. *Computers, Environment and Urban Systems*, *70*, 197–207. https://doi.org/10.1016/j.compenvurbsys.2018.03.008

Anselin, L. (1995). Local indicators of spatial association—LISA. *Geographical Analysis*, *27*, 93–115. https://doi.org/10.1111/j.1538-4632.1995.tb00338.x

Baddeley, A., Rubak, E., & Turner, R. (2015). *Spatial point patterns: Methodology and applications with R*. CRC Press. https://doi.org/10.1201/b19708

Bailey, M., Cao, R., Kuchler, T., Stroebel, J., & Wong, A. (2018). Social connectedness: Measurement, determinants, and effects. *Journal of Economic Perspectives*, *32*, 259–280. https://doi.org/10.1257/jep.32.3.259

Besag, J., & Newell, J. (1991). The detection of clusters in rare diseases. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *154*, 143–155. https://doi.org/10.2307/2982708

Borruso, G. (2008). Network density estimation: A GIS approach for analysing point patterns in a network space. *Transactions in GIS*, *12*, 377–402. https://doi.org/10.1111/j.1467-9671.2008.01107.x

Bossard, J. H. (1932). Residential propinquity as a factor in marriage selection. *American Journal of Sociology*, *38*, 219–224. https://doi.org/10.1086/216031

Brelsford, C., Thakur, G., Arthur, R., & Williams, H. (2019). Using digital trace data to identify regions and cities. *2nd ACM SIGSPATIAL International Workshop on Advances on Resilient and Intelligent Cities*, Chicago, IL (pp. 5–8). ACM.

Chainey, S., Tompson, L., & Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, *21*, 4–28. https://doi.org/10.1057/palgrave.sj.8350066

Comber, A. J., Brunsdon, C. F., & Farmer, C. J. (2012). Community detection in spatial networks: Inferring land use from a planar graph of land cover objects. *International Journal of Applied Earth Observation and Geoinformation*, *18*, 274–282. https://doi.org/10.1016/j.jag.2012.01.020

Coston, A., Guha, N., Ouyang, D., Lu, L., Chouldechova, A., & Ho, D. E. (2021). Leveraging administrative data for bias audits: Assessing disparate coverage with mobility data for COVID-19 policy. *ACM Conference on Fairness, Accountability, and Transparency* (pp. 173–184). ACM.

Credit, K., & Mack, E. (2019). Place-making and performance: The impact of walkable built environments on business performance in phoenix and Boston. *Environment and Planning B: Urban Analytics and City Science*, *46*, 264–285. https://doi.org/10.1177/2399808317710466

Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

Das, A., Ghosh, S., Das, K., Basu, T., Dutta, I., & Das, M. (2021). Living environment matters: Unravelling the spatial clustering of COVID-19 hotspots in Kolkata megacity, India. *Sustainable Cities and Society*, *65*, 102577. https://doi.org/10.1016/j.scs.2020.102577

DellaPosta, D. (2017). Network closure and integration in the mid-20th century American Mafia. *Social Networks*, *51*, 148–157. https://doi.org/10.1016/j.socnet.2016.11.005

Edwards, J. E. (2020). *Over the river and through the woods: Examining the relationship between network structure, collaboration and geography* [Ph.D. thesis]. Virginia Tech.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR (pp. 226–231). ACM.

Fischer, C. S. (1982). *To dwell among friends: Personal networks in town and city*. University of Chicago Press.

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, *659*, 1–44. https://doi.org/10.1016/j.physrep.2016.09.002

Gambetta, D. (1993). *The Sicilian Mafia: The business of private protection*. Harvard University Press.

Gao, Y., Li, T., Wang, S., Jeong, M.-H., & Soltani, K. (2018). A multidimensional spatial scan statistics approach to movement pattern comparison. *International Journal of Geographical Information Science*, *32*, 1304–1325. https://doi.org/10.1080/13658816.2018.1426859

Getis, A., & Franklin, J. (1987). Second-order neighborhood analysis of mapped point patterns. *Ecology*, *68*, 473–477. https://doi.org/10.2307/1938452

Getis, A., & Ord, J. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis*, *24*, 189–206. https://doi.org/10.1111/j.1538-4632.1992.tb00261.x

Giordano, A., Cole, T., & Le Noc, M. (2022). Spatial social networks for the humanities: A visualization and analytical model. *Transactions in GIS*, *26*, 1683–1702. https://doi.org/10.1111/tgis.12938

Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, *22*, 801–823. https://doi.org/10.1080/13658810701674970

Hu, Y., Wang, F., & Xierali, I. M. (2018). Automated delineation of hospital service areas and hospital referral regions by modularity optimization. *Health Services Research*, *53*, 236–255. https://doi.org/10.1111/1475-6773.12616

Hurlbert, A. H., & Jetz, W. (2007). Species richness, hotspots, and the scale dependence of range maps in ecology and conservation. *Proceedings of the National Academy of Sciences of the United States of America*, *104*, 13384–13389. https://doi.org/10.1073/pnas.0704469104

Ishioka, F., Kawahara, J., Mizuta, M., Minato, S., & Kurihara, K. (2019). Evaluation of hotspot cluster detection using spatial scan statistic based on exact counting. *Japanese Journal of Statistics and Data Science*, *2*, 241–262. https://doi.org/10.1007/s42081-018-0030-6

Kang, Y., Gao, S., Liang, Y., Li, M., Rao, J., & Kruse, J. (2020). Multiscale dynamic human mobility flow dataset in the US during the COVID-19 epidemic. *Scientific Data*, *7*, 1–13. https://doi.org/10.1038/s41597-020-00734-5

Koylu, C. (2018). Discovering multi-scale community structures from the interpersonal communication network on Twitter. In L. Perez, E. K. Kim, & R. Sengupta (Eds.), *Agent-based models and complexity science in the age of geospatial big data: Selected papers from a workshop on agent-based models and complexity science (GIScience 2016)* (pp. 87–102). Springer.

Kulldorff, M., & Nagarwalla, N. (1995). Spatial disease clusters: Detection and inference. *Statistics in Medicine*, *14*, 799–810. https://doi.org/10.1002/sim.4780140809

Liu, Y., Tong, D., & Liu, X. (2015). Measuring spatial autocorrelation of vectors. *Geographical Analysis*, *47*, 300–319. https://doi.org/10.1111/gean.12069

Maas, P. (1969). *The Valachi papers*. Bantam Books.

Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., Cleveland, W. S., Grannis, S. J., & Ebert, D. S. (2009). A visual analytics approach to understanding spatiotemporal hotspots. *IEEE Transactions on Visualization and Computer Graphics*, *16*, 205–220. https://doi.org/10.1109/TVCG.2009.100

Masser, I., & Scheurwater, J. (1980). Functional regionalisation of spatial interaction data: An evaluation of some suggested strategies. *Environment and Planning A*, *12*, 1357–1382. https://doi.org/10.1068/a121357

Mu, L., & Holloway, S. (2019). Neighborhoods. In J. P. Wilson (Ed.), *The geographic information science & technology body of knowledge* (1st quarter 2019 edition). University Consortium for Geographic Information Science.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 8577–8582. https://doi.org/10.1073/pnas.0601602103

Okabe, A., Okunuki, K., & Shiode, S. (2006). SANET: A toolbox for spatial analysis on a network. *Geographical Analysis*, *38*, 57–66. https://doi.org/10.1111/j.0016-7363.2005.00674.x

Okabe, A., & Yamada, I. (2001). The k-function method on a network and its computational implementation. *Geographical Analysis*, *33*, 271–290. https://doi.org/10.1111/j.1538-4632.2001.tb00448.x

Oliver, D. (2016). *Spatial network data: Concepts and techniques for summarization*. Springer. https://doi.org/10.1007/978-3-319-39621-7

Ord, J., & Getis, A. (1995). Local spatial autocorrelation statistics: Distributional issues and an application. *Geographical Analysis*, *27*, 286–306. https://doi.org/10.1111/j.1538-4632.1995.tb00912.x

Prestby, T., App, J., Kang, Y., & Gao, S. (2020). Understanding neighborhood isolation through spatial interaction network analysis using location big data. *Environment and Planning A: Economy and Space*, *52*, 1027–1031. https://doi.org/10.1177/0308518X19891911

Rae, A. (2009). From spatial interaction data to spatial interaction information? Geovisualisation and spatial structures of migration from the 2001 UK census. *Computers, Environment and Urban Systems*, *33*, 161–178. https://doi.org/10.1016/j.compenvurbsys.2009.01.007

Rae, A. (2011). Flow-data analysis with geographical information systems: A visual approach. *Environment and Planning B: Planning and Design*, *38*, 776–794. https://doi.org/10.1068/b36126

Ripley, B. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, *13*, 255–266. https://doi.org/10.2307/3212829

Sarkar, D., Sieber, R., & Sengupta, R. (2016). GIScience considerations in spatial social networks. In J. A. Miller, D. O'Sullivan, & N. Wiegand (Eds.), *9th International Conference on Geographic Information Science* (pp. 85–98). Springer. https://doi.org/10.1007/978-3-319-45738-3_6

Sarkar, D., & Yadav, P. (2021). Donut visualizations for network-level and regional-level overview of spatial social networks. *arXiv:2101.00929*.

Shiode, S. (2011). Street-level spatial scan statistic and STAC for analysing street crime concentrations. *Transactions in GIS*, *15*, 365–383. https://doi.org/10.1111/j.1467-9671.2011.01255.x

Shiode, S., & Shiode, N. (2013). Network-based space-time search-window technique for hotspot detection of street-level crime incidents. *International Journal of Geographical Information Science*, *27*, 866–882. https://doi.org/10.1080/13658816.2012.724175

Shivanasab, P., Abbaspour, R. A., & Chehreghan, A. (2021). An assessment on the performance of the shape functions in clustering based on representative trajectories of dense areas. *GIScience & Remote Sensing*, *58*, 1219–1249. https://doi.org/10.1080/15481603.2021.1973217

Silverman, B. W. (1998). *Density estimation for statistics and data analysis*. Chapman & Hall/CRC Press.

Tao, R., & Thill, J.-C. (2016). A density-based spatial flow cluster detection method. *9th International Conference on GIScience*, Montreal, QC, Canada (Vol. *1*, pp. 288–291).

Wang, T.-C., & Phoa, F. K. H. (2016). A scanning method for detecting clustering pattern of both attribute and structure in social networks. *Physica A: Statistical Mechanics and its Applications*, *445*, 295–309. https://doi.org/10.1016/j.physa.2015.10.009

Wang, Y., Kang, C., Bettencourt, L. M., Liu, Y., & Andris, C. (2019). Linked activity spaces: Embedding social networks in urban space. In M. Helbich, J. J. Arsanjani, & M. Leitner (Eds.), *Computational approaches for urban environments* (pp. 313–336). Springer. https://doi.org/10.1007/978-3-319-11469-9_13

Yang, Y., & Diez-Roux, A. V. (2012). Walking distance by trip purpose and population subgroups. *American Journal of Preventive Medicine*, *43*, 11–19. https://doi.org/10.1016/j.amepre.2012.03.015

How to cite this article: Liang, X., Baker, J., DellaPosta, D., & Andris, C. (2023). Is your neighbor your friend? Scan methods for spatial social network hotspot detection. *Transactions in GIS*, *27*, 607–625. https://doi.org/10.1111/tgis.13050