International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

An Examination of Machine Learning-Based Outlier Identification from Mobile Phone Tracks

Mr. P Isaac Paul[1], Mr. A.V. Ramana[2], Mr. N. Vara Prasad[3]

[1]Associate Professor, Dept. of Computer Science and Engg., RISE Krishna Sai Gandhi Group of Institutions, Ongole, A.P.
[2,3]Assistant Professor, Dept. of Computer Science and Engg., RISE Krishna Sai Prakasam Group of Institutions, Ongole, A.P.

Abstract:
In this paper, two machine learning algorithms—local outlier factor (LOF) and density-based spatial clustering of applications with noise (DBSCAN)—that are used to identify outliers in the context of a continuous framework for point of interest (PoI) detection are analyzed. The mobile trajectories of users are continuously and almost instantaneously loaded into this system. These frameworks are still in their infancy, but they are already essential for large-scale sensing deployments, such as Smart City planning deployments, where the anonymous individual mobile user trajectories can be valuable to improve urban planning. There are two contributions made by this paper. First, the functional design of the entire PoI detection architecture is provided by the study. Second, the study evaluates the effectiveness.

Keywords: outliers; DBSCAN; LOF; GPS trajectories; machine learning.

1. Introduction:

Municipalities look for areas of interest for both inhabitants and visitors as part of their ongoing planning and development of Smart Cities in order to better design the services that are offered. Finding new sites of interest (PoIs) that actually match these criteria is therefore essential. The population's demands for amusement and recreation, whether cultural or otherwise. Pervasive computing and pervasive technology are crucial in this situation because they enable constant tracking and learning of PoIs depending on citizen preferences and without compromising the privacy of personal data. In the past ten years, people-centric hardware and systems have emerged in pervasive technology, which has also grown more sustainable and efficient. Citizens carry around personal mobile IoT devices [1], including smart phones, which can help with inferring trajectories in a way that is anonymous and does not compromise privacy. A series of waypoints is represented by anonymous trajectories that are continuously gathered. Then, using various machine learning (ML) techniques, individual trajectories can be pooled and analyzed to estimate PoIs related to a population's social patterns. The accuracy and viability of using ubiquitous technology as the foundation for creating extensive sensing frameworks that can support Smart City planning have been the subject of related research [2, 3]. These frameworks frequently use planned PoIs by depending on offline data analysis, but not a persistent identification of points of interest (PoIs) throughout time, such as those connected to seasonal events or evolving group interests. Therefore, some PoIs in Smart Cities are not explicitly regarded as such, and those are typically the ones that are more difficult to detect. However, they are highly relevant to achieve efficient city planning, as they are derived from citizens' social behavior and mobility preferences. In the traditional definition, a PoI is a point of interest (a location) that may be of interest to someone. Our definition of PoI attempts to capture such locations, currently unknown to a municipality or to a community, and that is revealed to us by the social trajectories of a large number of people, which cross a specific location. The aim of our work is to consider non-intrusive technology to detect such PoIs. To discover them, we use cell phone traces data that reflect the trajectories followed by people in their daily lives. These trajectories reveal their individual stopping points (stop points). A cluster in time and in space of stop points can turn out to be a PoI for the local community or for an individual visitor. While trajectories collected from smart phones assist in detecting PoIs, they also include outliers. By definition, an outlier is a representation of a point that is beyond the limits of a well-defined population [4]. The traces of stored cell phones, which represent the trajectories followed by users, experience several problems, from communication failures to problems with data recording on the cloud. Hence, an outlier is a cluster of points that derive from an error, and not from stop points. It is this discovery of outliers, in the pursuit of data cleansing, that is reflected in our current work. In this context, this work proposes a design for a novel non-intrusive framework for inference of PoIs based on mobility trajectories derived from anonymised smart phone data, and performs a performance analysis of DBSCAN and LOF for outlier detection, in the context of a first functional block of such framework, related to outlier detection. We consider that a relevant framework in this context would be a framework that could assist in inferring similarity in individual (and collective) mobility patterns. Data would be obtained via external devices (e.g., smart phones and other embedded devices carried by citizens), and the inference could be done (a) in real time or (b) close to real time. The initial framework described in this paper considers that trajectory data can be obtained from the city Internet of Things (IoT) infrastructure and also from citizens' personal devices, such as smart phones, upon consent. Such a framework would, therefore, infer some form of human behavior and map such behaviour in time and space to potential PoIs which can then be deployed on a Smart City dashboard, or on Smart City user applications. To reach such a level of inference, it is necessary to design a framework that can assist a continuous PoI detection, by comparing different ML approaches, and this aspect is currently a major gap in the literature. This work contributes to overcoming such a gap by providing the following contributions:

1. Provides the initial functional design of a novel framework for continuous detection of PoIs based on smart and anonymised trajectory data collected from personal devices and IoT Smart City infrastructure.
2. Addresses the issue of outlier detection and provides a validation of outlier detection based on two ML algorithms, density-based spatial clustering of applications with noise (DBSCAN) and local outlier factor (LOF).

The first reason for selecting these two specific algorithms, DBSCAN and LOF, lies in the fact that related literature states that these two algorithms are within the ones most relevant in the identification of outliers as shall be debated in Section 2. Some authors, such as Osmar et al., prefer LOF [5]. Other authors, such as Allhussein et al., prefer DBSCAN to analyze outliers [6]. While both algorithms exhibit interesting properties, there is no study comparing both of them in terms of capability to support outlier detection, assuming a framework that relies on trajectory data captured by IoT and personal devices. A second reason for considering these two specific algorithms lies in the simplicity of both algorithms, a key aspect to consider in a continuous PoI detection framework, which is further addressed in Section 3. The remainder of this paper is organized as follows. Section 2 provides a description of related literature and of our contributions in comparison to prior work. Section 3 defines our proposal for continuous PoI detection, and its functional blocks, debating PoI detection and inference aspects and introducing also challenges with outlier detection. Section 4.

2. Existing work:

Over the past decade, there has been an increase in the usage of mobile crowd sensing (MCS) for urban planning in Smart Cities [7]. Yang et al. analyzed mobile phone traces to identify particular points of interest (POIs), including home and work [2]. The authors demonstrate that mobile trajectories may accurately identify user interests at a fine-grained level. However, detecting outliers is not a central theme in this study.

Butron-Revilla et al. [8] use mobile phone data to identify movement patterns and points of interest. Viswanathan et al. use situational awareness to categorize points of interest (PoIs) as stopovers, special interest areas, or occasional stops (9).

Another approach to PoI detection relies on detecting outliers. Authors handle the topic of outliers differently, based on their techniques and use cases. Prioritize LOF when comparing global and local outlier detectors (source: [5]). Their paper focuses on data streams used in big data, analyzing both parametric and non-parametric methodologies and proposing a novel way for using LOF in data streams. This study supports the applicability of LOF to data streams and offers a reliable approach for its application. However, the research did not compare LOF to other algorithms like DBSCAN.

Several writers, including Markou and Singh [11], Goldstein et al. [12], Patcha et al. [13], and Alimohammadi et al. [14], have published surveys and reviews on outlier identification approaches, including analytical comparisons of their characteristics. Outlier detection solutions commonly used in this area include LOF and DBSCAN.

Zhipeng and Dechang offer DBTOD, a complicated approach that uses Hausdorff distance in metric spaces to identify outlying sub-trajectories (16). We focus on identifying outliers in sub-trajectories, rather than eradicating them.

Youcef and Djenouri used group trajectory outlier detection (GTOD) and closed DBSCAN k-nearest (CDkNN-GTOD) algorithms [17] to identify outlier groups in a dataset. Goodge et al. presented Lunar, a graph neural network technique for outlier identification. They identified LOF and DBSCAN as significant approaches because to their simplicity and efficacy [18]. This concept is intriguing and will be explored more in future studies.

3. A Framework for Continuous PoI Detection in Smart Cities.

3.1. Smart Cities and Urban Sensing: Background

Data collection from cyber-physical systems (CPS) embedded in IoT infrastructures and mobile personal devices, such as smart phones, is crucial for developing Smart Cities or Smart Communities. MCS Smart City apps leverage IoT infrastructures and personal CPS to enhance people-centric services. MCS is used in various Smart City services, including infrastructure monitoring (e.g., energy consumption), social behavior awareness [19,20], traffic pattern improvement [21], and detecting points of interest based on user behavior and preferences [22].

The focus is on identifying sites of interest (POIs) based on user movement behavior in a city, rather than making suggestions based on pre-established criteria. For this reason, the next parts begin by outlining the suggested design and then introduce the notion of PoI in the context of our work. Then, discuss how to find points of interest and outliers.

3.2. Proposed Framework Functional Blocks

The stated step sequence is not arbitrary; nevertheless, some functional blocks may be arranged in a different order. Outliers, for example, might be computed before or after the identification of stop points (SPs), which are probable locations of interest in a trajectory. We selected to execute a previous identification of SPs based on visit duration (the amount of time a user spends motionless in a spot), distance traveled, and speed between SPs.

The system analyzes trajectory data to extract metrics such as speed, distance between measurements, and visit duration to help detect PoIs more precisely. Additional information may be inferred, such as the method of transportation considered by the user, which may help discover similarities in different trajectories and hence aid in the detection of PoIs. The following functional block (B) detects outliers and uses the remaining data to estimate SPs. While outlier identification may be conducted after calculating SPs, we believe that it should be addressed first. To prevent erroneous conclusions, undesired values (outliers) must be deleted before to the SP inference.

Still, in the context of SP inference, there is the need to validate next the obtained SPs, by removing SPs that may occur sporadically, for instance. For instance, it is feasible to consider the number of times that the detected SPs are visited, and then cross-reference each SP across different individual trajectories. PoIs can then be obtained after validating the number of different trajectories sharing a common SP, and comparing the average visit time against individual visit times of that SP. Then, a final list of PoIs will be created. However, this process is continuous, and when repeated for new data sources and respective trajectories, the PoIs already found are automatically removed during the SP Detection in order to discover new PoIs. The next section provides further detail concerning PoI detection.

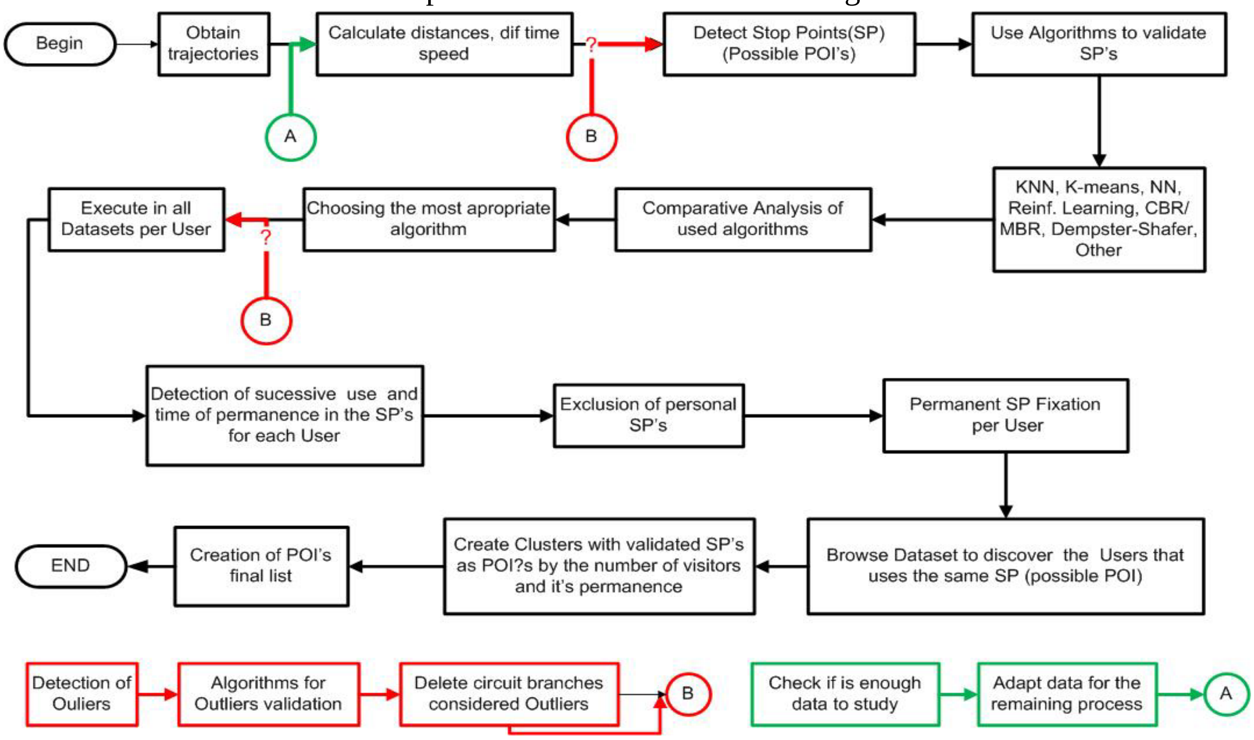

Figure 1. Functional blocks of a framework for continuous detection of PoIs in a Smart City, assuming data obtained in close to real time from MCS applications.

3.3. Interdisciplinary PoI Definition and Detection Aspects

The definition of PoI [4] in the context of our work relates to MCS and integrates user social behavior. Our PoI concept goes beyond the usual spatial data (e.g., GPS coordinates), and follows the line of work that considers PoIs to be a product of space, time, and some measure of influence/attraction [24]. For instance, Chan et al. defined a framework for personalized tour recommendations based on user interests and network visit duration [25]. Their work assumes that there are already pre-established PoIs (municipality data), and the recommendation engine provides a recommendation based on such a PoI set only. However, the overlapping of different individual trajectories can also assist in detecting PoIs which are based on user preferences. From an individual (one user perspective) a PoI is related to the social attractiveness of a user to a specific event or activity, which is defined by different attributes, geo-location being one such attribute. The social attractiveness level varies with time and space and increases with a larger visit to a specific location. For instance, a person can, in his/her daily routine, stop at a specific location due to having met an acquaintance, or even for curiosity and not necessarily due to an activity or event. This would be a transient SP and should not result in a PoI.

The detection of new PoIs is, therefore, derived from cluster similarity obtained when considering multiple individual trajectories. For this purpose, our work currently considers two different definitions to detect PoIs.

Definition 1: A PoI is defined by its edge between's, which refers to the number of trajectories crossing a certain point or radius. Similarity analysis can identify a point of interest within a specified time and space range.

Definition 2: A PoI is the grouping of individual trajectories based on their spatial overlap. In this case, a PoI is identified when clusters are densely packed. Other criteria like as speed, visit time, and temporal granularity have an influence on both definitions.

3.4. Outlier Detection Features

As mentioned in Section 1, an outlier is a point that exceeds the boundaries of a well-defined population. Detecting outliers is the first step in our suggested system to prevent erroneous data aggregation. Outliers might be caused by sensor failures, faulty equipment or software, or data movement from user devices to the cloud. Individual changes in residents' mobility behavior may potentially contribute to these incidents.

3.5. Privacy and Security Considerations

MCS demands, first and foremost, the user's permission. Then, MCS relies on acquired data (for example, accesses to wireless networks), which may be piggybacked, compromising privacy. Obtaining answers regarding people' behavior may require identifying personal data from users or smartphones, thus compromising privacy. While MCS do not always gather personal data, the General Data Protection Regulation (GDPR) enacted in the European Union requires users to give their consent before using MCS apps.

A continuous PoI detection mechanism, however, does not require any personal information. In reality, trajectories detected by a Smart City IoT infrastructure or mobile phones.

4. ML for Outlier Detection

Outlier detection in the context of the framework explained in Section 3.2 is the first functional block to realise the overall framework. If outliers are not removed, then the resulting PoIs will be inconsistent. Such inconsistency may relate to poor data collection, for instance. Another use of outlier detection is to discover abnormal patterns. We intend to exclude all points and/or trajectory segments that impair trajectory detection. ML is, therefore, relevant to be considered in this context. The choice of specific ML

algorithms requires an approach that best serves the requirements of the specific solution. ML algorithms relevant to outlier detection can be categorised as follows [30]:

1. **Nearest neighbour-based**. Based on the comparison of the distances between various points and their nearest neighbours [31].
2. **Density-based** [32]. Based on the measurement of the higher or lower density of points in a given region, which has resulted in a new definition of local outliers, with the same principle. Here we consider both LOF and DBSCAN as representative algorithms of this category.
3. **Distance-based** [33]. Defined by measuring the distances from a given point O to other points, resulting in points in its neighborhood and outliers. Examples are k-nearest neighbours (kNN), k-means (k-MEANS) clustering, and learning vector quantisation (LVQ).

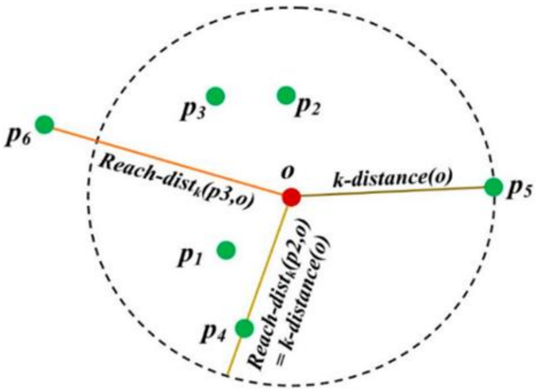

Figure 2. LOF principles: k-distance of an object p from a point 0 and k-distance neighbourhood of p, Nk, and range distance (rdist) (p).

5. Performance Analysis and Evaluation

This section describes the performance evaluation developed to detect outliers. We have carried out experiments with DBSCAN and LOF for the same datasets.

5.1. Methodology

With regard to the methodology used, it can be described as follows:

1. **Selection and analysis of different datasets**. Specific PoI databases are still sparse, in particular considering the parameters proposed for the PoI definition, e.g., speed, visit time, time granularity and geo-location
2. **Dataset cleanup and validation**. Selected datasets have been transformed into new datasets also integrating time travelled across sequential waypoints of a trajectory; distance travelled; speed; day of the week; user id (obfuscated); means of transport.

3. **Selection of ML algorithms.** The algorithms proposed in this work, DBSCAN and LOF, were applied to detect outliers with the help of Python programming language, proceeding where necessary to any adjustments of parameters or possible tuning of the algorithms.
4. **Performance evaluation**. The accuracy of the algorithms used was measured, verifying their validity and applicability throughout the study, and verifying which factors were preponderant for the analysis.

5.2. Performance Evaluation Parameters

We have evaluated the capability of DBSCAN and LOF in detecting outliers based on the classification parameters precision and recall. We have also analysed the overall model accuracy. **Accuracy** provides a measure of the overall performance of the algorithm. It corresponds to all true classifications (true positives, true negatives) over all classified values (positive and negative). By definition, **precision** corresponds to the number of items correctly labelled as true positives, divided by the total number of elements belonging to the positive class. Precision, defined in Equation (11), provides a measure of how well an algorithm can detect only outliers. Precision is provided as a percentage of the number of outliers in the dataset. The precision is as close to one as the false positives are close to zero.

5.3. Outlier Detection with DBSCAN

In the context of this work, DBSCAN is applied to different individual trajectories. This is not a trivial task, as the two parameters eps and MinPts require calibration for each individual trajectory. On our proposed framework for continuous detection of PoIs, this implies that the inference engine will have to use a large range of MinPts and eps, as exemplified in Table 1, that is, with the DBSCAN algorithm, a range of eps parameter values has been tested, and for each, a range of MinPts values has also been tested. For instance, assuming an EPS of 7, then a MinPts value of 2 or of 3 has the same result, it assists in detecting 35 outliers. This approach involves human assistance in the sense that the modification necessitates a study of the visual outcome. This suggests that an excessive amount of work will be performed on each trajectory, which is incompatible with a continuous detection engine. Our technique is based on detecting the knee point, which is roughly defined as the point of greatest curvature in a system as well as the ideal value in terms of eps. This strategy, as seen in Kaggle (https://www.kaggle.com/kevinarvai/knee-elbow-point-detection, seen on August 12, 2022), is better suited for a continuous engine handling various trajectories (dataflows). After optimizing the eps value for a certain MinPts, we performed the computation using DBSCAN.

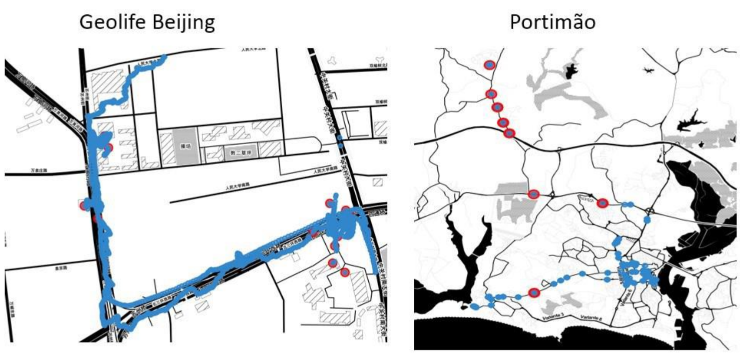

Figure 3. Illustration of detected outliers in the GEO and PTM datasets with DBSCAN. Outliers are highlighted in red.

5.4. LOF and DBSCAN Comparison

5.4.1. GEO Dataset Results

A global perspective on the performance (precision, recall, accuracy) of LOF, when applied to the GEO dataset, is provided in Figure 7, while the same performance perspective for DBSCAN is provided in Figure 8. For each chart, the X-axis represents the number of individual trajectories available. LOF (refer to Figure 7) reaches a stable precision across all trajectories, while the accuracy varies still within a good level. The recall (how well outliers are detected) results show, however, that LOF exhibits some difficulty in detecting outliers.

DBSCAN (refer to Figure 8) results in more variability in terms of the three evaluation dimensions (precision, accuracy, and recall). Therefore, LOF is the algorithm that performs best in terms of outlier detection for the GEO dataset.

5.4.2. PTM Dataset Results

For the smaller dataset, PTM, LOF results are shown in Figure 9 and DBSCAN results are provided in Figure 10. LOF exhibits a good level of accuracy again, but precision and recall are lower. DBSCAN (refer to Figure 10) has a significantly lower precision, recall, and accuracy.

5.5. Discussion of Results

Table 3 provides the achieved accuracy for LOF and GEO. Overall, LOF provided the best results in terms of outlier detection. The reason for this is related to the fact that LOF gives more importance to local outlier detection than other methods such as DBSCAN.

Table 3. Average accuracy of DBSCAN and LOF for the datasets GEO and PTM.

ML Approach	GEO	PTM
DBSCAN	0.78	0.20
PTM	0.88	0.78

Moreover, LOF is easier to parameterize because it varies by only one parameter (MinPts) and the variability of this factor can be tested more easily. LOF also shows better accuracy on dense datasets (GEO). DBSCAN is more difficult to parameterize due to the fact that one has to articulate two parameters, eps and MinPts, even if a refinement approach, such as kNN (as we have considered) is applied. DBSCAN exhibited better performance on the sparser dataset (PTM), but overall, lower performance in terms of outlier detection. In the same way, we can compare the performance of DBSCAN in the different datasets, highlighting the fact that the PTM dataset is very small and, in addition, each trajectory is much smaller and covers a much smaller distance than that revealed by the GEO Dataset. However, as mentioned, LOF behaves very well in both datasets, showing an accuracy above 80%. While DBSCAN exhibits a more variable behaviour, as accuracy significantly lowers when DBSCAN is applied to a sparse dataset such as PTM.

6. Summary

This paper presents an innovative framework for the continuous detection of PoIs based on mobile phone trajectories, and analyses ML-based algorithms, specifically, DBSCAN and LOF, to be applied for the continuous detection of outliers. The detection of outliers corresponds to one of the relevant

functional blocks in the proposed PoI detection framework. To the best of our knowledge and as corroborated in Section 2, where we have analysed related work, the framework for continuous detection of PoIs is novel and presents the basis for a much-desired aspect in urban planning in Smart Cities. This is the possibility to improve services via existing data, via a consented, non-intrusive, and pervasive data-collecting approach. In addition to the architectural design of such a framework, the paper focuses on the detection of outliers. After checking different algorithms as explained in Section 4, DBSCAN and LOF have been selected as they are representative algorithms for both density-based outlier detection and distance-based outlier detection.
The aim was to understand which algorithm could best suit a continuous PoI detection framework in the context of outlier detection (one of the proposed blocks).

References

1. Samara, M.A.; Bennis, I.; Abouaissa, A.; Lorenz, P. A survey of outlier detection techniques in IoT: Review and classification. J. Sens. Actuator Netw. **2022**, 11, 4.
2. Yang, Y.; Xiong, C.; Zhuo, J.; Cai, M. Detecting home and work locations from mobile phone cellular signaling data. Mob. Inf. Syst. **2021**, 2021, 5546329.
3. Karnatak, H.; Pandey, K.; Raghavaswamy, V. Online Geodata Repositories, Geoweb Services and Emerging Geospatial Technologies for Smart City Planning. In Smart Cities for Sustainable Development; Springer: Berlin/Heidelberg, Germany, 2022; pp. 211–229.
4. Almeida, A.; Alves, A.; Gomes, R. Automatic POI Matching Using an Outlier Detection Based Approach. In Advances in Intelligent Data Analysis XVII, Proceedings of the 17th International Symposium, IDA 2018, Hertogenbosch, The Netherlands, 24–26 October 2018;
5. Duivesteijn, W., Siebes, A., Ukkonen, A., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 40–51.
6. Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A review of local outlier factor algorithms for outlier detection in big data streams. Big Data Cogn. Comput. **2021**, 5, 1.
7. Alhussein, I.; Ali, A.H. Application of DBSCAN to Anomaly Detection in Airport Terminals. In Proceedings of the 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), Najaf, Iraq, 6–7 September 2020; pp. 112–116.
8. Capponi, A.; Fiandrino, C.; Kantarci, B.; Foschini, L.; Kliazovich, D.; Bouvry, P. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. IEEE Commun. Surv. Tutor. **2019**, 21, 2419–2465.
9. Butron-Revilla, C.; Suarez-Lopez, E.; Laura-Ochoa, L. Discovering urban mobility patterns and demand for uses of urban spaces from mobile phone data. In Proceedings of the 2021 2nd Sustainable Cities Latin America Conference (SCLA), Medellin, Colombia, 25–27 August 2021; pp. 1–6.
10. Viswanathan, S.; Boulard, C.; Bruyat, A.; Maria Grasso, A. Situational Recommender: Are You On the Spot, Refining Plans, or Just Bored? In Proceedings of the CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April–5 May 2022; pp.
11. Ma, M.X.; Ngan, H.Y.; Liu,W. Density-based outlier detection by local outlier factor on largescale traffic data. Electron. Imaging **2016**, 2016, 1–4.
12. Markou, M.; Singh, S. Novelty detection: A review—Part 1: Statistical approaches. Signal Process. **2003**, 83, 2481–2497.

13. Goldstein, M.; Uchida, S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS ONE **2016**, 11, e0152173.
14. Patcha, A.; Park, J.M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw. **2007**, 51, 3448–3470.
15. Alimohammadi, H.; Chen, S.N. Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. Expert Syst. Appl. **2022**, 191, 116371.
16. Sabarish, B.; Karthi, R.; Kumar, T.G. Spatial Outlier Detection Algorithm for Trajectory-Data. Int. J. Pure Appl. Math. **2018**, 118, 325–331.
17. Liu, Z.; Pi, D.; Jiang, J. Density-based trajectory outlier detection algorithm. J. Syst. Eng. Electron. **2013**, 24, 335–340.
18. Djenouri, Y.; Nørvåg, K.; Ramampiaro, H.; Li, J.C.W. Fast and accurate group outlier detection for trajectory data. In New Trends in Databases and Information Systems, Proceedings of the European Conference on Advances in Databases and Information Systems, Lyon, France, 25–27 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 60–70.
19. Goodge, A.; Hooi, B.; Ng, S.K.; Ng, W.S. Lunar: Unifying local outlier detection methods via graph neural networks. Proc. AAAI Conf. Artif. Intell. **2022**, 36, 6737–6745.
20. Marakkalage, S.H.; Sarica, S.; Lau, B.P.L.; Viswanath, S.K.; Balasubramaniam, T.; Yuen, C.; Yuen, B.; Luo, J.; Nayak, R. Understanding the lifestyle of older population: Mobile crowdsensing approach. IEEE Trans. Comput. Soc. Syst. **2018**, 6, 82–95
21. Carvalho, L.I.; Sofia, R.C. A review on scaling mobile sensing platforms for human activity recognition: Challenges and recommendations for future research. IoT **2020**, 1, 451–473.
22. Ali, A.; Ayub, N.; Shiraz, M.; Ullah, N.; Gani, A.; Qureshi, M. Traffic Efficiency Models for Urban Traffic Management Using Mobile Crowd Sensing: A Survey. Sustainability **2021**, 13, 13068.
23. Marakkalage, S.H.; Lau, B.P.L.; Zhou, Y.; Liu, R.; Yuen, C.; Yow,W.Q.; Chong, K.H. WiFi Fingerprint Clustering for Urban Mobility Analysis. IEEE Access **2021**, 9, 69527–69538.
24. Abbas, R.; Hassan, G.M.; Al-Razgan, M.; Zhang, M.; Amran, G.A.; Al Bakhrani, A.A.; Alfakih, T.; Al-Sanabani, H.; Rahman, S.M.M. A Serendipity-Oriented Personalized Trip Recommendation Model. Electronics **2022**, 11, 1660.
25. Sofia, R.C.; Carvalho, L.; Pereira, F.M. The Role of Smart Data in Inference of Human Behavior and Interaction. In Smart Data; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019; pp. 191–214.
26. Lim, K.H.; Chan, J.; Leckie, C.; Karunasekera, S. Personalized tour recommendation based on user interests and points of interest visit durations. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
27. Liu, L. From Data Privacy to Location Privacy: Models and Algorithms. In Proceedings of the 33rd International Conference on Very Large Data Bases, Vienna, Austria, 23–27 September 2007; Volume 7, pp. 1429–1430.
28. Ali, T.; Asghar, S.; Sajid, N.A. Critical analysis of DBSCAN variations. In Proceedings of the 2010 International Conference on Information and Emerging Technologies, Karachi, Pakistan, 14–16 June 2010; pp. 1–6.
29. Kifer, D.; Machanavajjhala, A. No free lunch in data privacy. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, Athens, Greece, 12–16 June 2011; pp. 193–204.

30. Li, S.; Shen, H.; Sang, Y.; Tian, H. An efficient method for privacy-preserving trajectory data publishing based on data partitioning. J. Supercomput. **2020**, 76, 5276–5300.
31. Knox, E.M.; Ng, R.T. Algorithms for mining distance based outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases, New York, NY, USA, 24–27 August 1998; pp. 392–403.
32. Adam, N.R.; Janeja, V.P.; Atluri, V. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; pp. 576–583.
33. Parimala, M.; Lopez, D.; Senthilkumar, N. A survey on density based clustering algorithms for mining large spatial databases. Int. J. Adv. Sci. Technol. **2011**, 31, 59–66.
34. Hubballi, N.; Patra, B.K.; Nandi, S. NDoT: Nearest neighbor distance based outlier detection technique. In International Conference on Pattern Recognition and Machine Intelligence, Proceedings of the 4th International Conference, PReMI 2011, Moscow, Russia, 27 June–1 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 36–42.
35. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X.; et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
36. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; pp. 93–104.
37. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the Fifth International Conference on the Applications of Digital Information andWeb Technologies (ICADIWT 2014), Bangalore, India, 17–19 February 2014; pp. 232–238.
38. Liu, J.; Cao, Y.; Li, Y.; Guo, Y.; Deng,W. A big data cleaning method based on improved CLOF and Random Forest for distribution network. CSEE J. Power Energy Syst. 2020, early access.
39. Ester, M.; Kriegel, H.P.; Xu, X. A Database Interface for Clustering in Large Spatial Databases; Institut für Informatik: Oldenburg, Germany, 1995; Volume 2.