IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2018, accepted January 7, 2019, date of publication January 15, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891933

Adapted K-Nearest Neighbors for Detecting Anomalies on Spatio–Temporal Traffic Flow

YOUCEF DJENOURI[1], ASMA BELHADI[2], JERRY CHUN-WEI LIN[3], AND ALBERTO CANO[4]

[1]Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway
[2]RIMA, University of Science and Technology Houari Boumediene, Algiers, Algeria
[3]Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Bergen, Norway
[4]Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA

Corresponding author: Alberto Cano (acano@vcu.edu)

ABSTRACT Outlier detection is an extensive research area, which has been intensively studied in several domains such as biological sciences, medical diagnosis, surveillance, and traffic anomaly detection. This paper explores advances in the outlier detection area by finding anomalies in spatio–temporal urban traffic flow. It proposes a new approach by considering the distribution of the flows in a given time interval. The flow distribution probability (FDP) databases are first constructed from the traffic flows by considering both spatial and temporal information. The outlier detection mechanism is then applied to the coming flow distribution probabilities, the inliers are stored to enrich the FDP databases, while the outliers are excluded from the FDP databases. Moreover, a k-nearest neighbor for distance-based outlier detection is investigated and adopted for FDP outlier detection. To validate the proposed framework, real data from Odense traffic flow case are evaluated at ten locations. The results reveal that the proposed framework is able to detect the real distribution of flow outliers. Another experiment has been carried out on Beijing data, the results show that our approach outperforms the baseline algorithms for high-urban traffic flow.

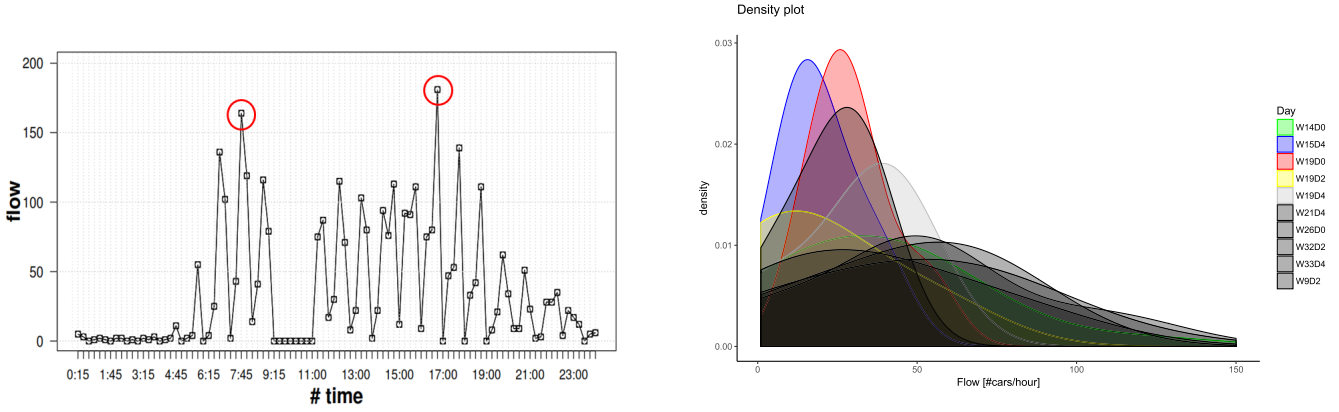
INDEX TERMS Anomaly detection, kNN, flow distribution probability.

I. INTRODUCTION

Spatio-temporal data embodies information related to space and time dimensions [1]. Spatio-temporal data mining is largely used in many number of domains including ecology [2], climatology [3], earth sciences [4], epidemiology [5], and urban traffic [6], [7]. The aim is to adapt classical data mining techniques and propose new methods for discovering useful knowledge from spatio-temporal data [8]. Recent surveys overviewing spatio-temporal data mining techniques including spatio-temporal clustering, spatio-temporal pattern mining can be found in [9]–[13]. One application of spatio-temporal data mining is spatio-temporal outlier detection. The goal is to identify anomalies from both spatial and temporal information from the input data [14].

With the popularization of GPS and IT devices, urban traffic flow analysis has attracted growing attention in the last decades. Zheng [15] and Feng *et al.* [7] reviewed spatio-temporal data mining techniques. The surveys included segmentation and clustering, detecting outliers and anomaly flows, classification sub-trajectories, and finding frequent and periodical sequential patterns from clusters of trajectories. The traffic flow is computed by counting the number of objects (cars, passengers, taxis, buses, etc) across a given location during a time interval. This generates a high number of Flow Distribution Probabilities (FDP).

One of the main applications in urban traffic analysis is detecting anomalies from the traffic flow data. The aim is to identify flow values significantly different to other flow values by considering both spatial and temporal information of urban traffic data. A useful way for anomaly detection on traffic flow is to apply outlier detection techniques. An outlier is defined as an observation (or a set of observations) which appears to be inconsistent with the remainder of that set of data [16]. Outlier detection has been intensively studied in the last two decades. It can be categorized as statistical-based methods, distance-based methods, deviation-based methods, density-based methods, and clustering-based methods [16]–[21]. Interesting recent surveys which reviews on existing outlier detection are found in [22] and [23]. In addition, many researchers have investigated outlier urban flow detection [24]–[27]. In the anomalous urban traffic flow data we aim to learn from different traffic actors (bikes, cars, buses, and trucks), the unusual behaviors represented by anomalous flow values due to some



(a) (b)

FIGURE 1. Motivated Example: outliers in flows vs. outliers in flow distributions. (a) Flows over time (potential outliers marked). (b) Flow probability distributions: outliers (in color) vs. inliers (in grey).

circumstances such as oversaturated conditions [28], traffic congestion [29], and bottlenecks [30].

A. MOTIVATION

Let us illustrate the conceptual difference with an example in Figure 1. In Figure 1(a) we illustrate the flows over time (a time series of the flow values measured over the day) for the Anderupvej location at Odense city in Denmark for two weeks excluding the weekend days. Each flow value is determined for an interval of 15 minutes. In Figure 1(b) we show the distributions of flow values per day between 07:00 to 10:00, for the working days over two weeks for the same location. Here, each flow value is determined for a 1-hour interval. The existing algorithms in the literature can only detect single flows. For example, the flows marked by red circles in Figure 1(a) might be unusual. Such methods could have the use case of real-time detection of, e.g., sudden peaks. However, unusual flow distributions (i.e., longer periods of flow measurements) as represented in Figure 1(b) cannot be detected by the state-of-the-art algorithms for outlier detection in flow sequences. In this example, the distributions colored red and blue, respectively, are rather different from the other flow distributions and hint at unusual conditions with impact on the overall traffic behavior on those days. Indeed, the distributions colored red and blue have large flow values between 0 to 50. Contrary to the other FDPs, which regular density of flows among the interval $[0, 100]$ is observed. The aim of this paper is to identify such outliers by proposing an outlier detection framework for flow distribution probabilities.

B. CONTRIBUTION

In this paper, we propose a technique based on the kNN algorithm for identifying anomalies on distributions of flows. The main contributions of this work are as follows.

1) We propose a new framework that updates the historical data for dealing with outlier FDP of the traffic flow data.
2) We propose a strategy for constructing the historical FDP database by taking into account both spatial and temporal information of the traffic flow data.
3) We extend the kNN algorithm for FDP data and we adapt the KL-divergence [31] for computing distances between the FDPs.
4) We present a case study on real data from both Odense and Beijing traffic flow to demonstrate the usefulness of the proposed framework. The results reveal that the proposed framework is able to detect the real distribution of flow outliers. In addition, it is very competitive compared to the state-of-the-art algorithms for solving big urban traffic networks.

C. OUTLINE

The paper is organized as follows. Section II reviews outlier techniques for spatio-temporal traffic data. Section III presents the proposed framework for outlier flow distribution probability detection. Section IV presents the experimental analysis on real world data. Section V summarizes the conclusions and outlines future work.

II. RELATED WORK

In this paper, we are interested in urban traffic. In the following, we relate recent studies on spatio-temporal urban traffic data mining techniques, and in particular, the outlier and anomaly detection from flow data.

A. URBAN TRAFFIC FLOW DATA MINING

In the last decade, several data mining approaches have been proposed for urban traffic analysis. Landesberger *et al.* [32] present a visual analysis study of people flows among places in the London city area. The people flows are aggregated into regions to reduce the mass mobility patterns using the k-means algorithm [33]. However, only aggregated regions are shown to the user for better understanding the flow distribution between places. Zheng *et al.* [34] address the

problem of mining sequential patterns in semantic trajectories, leveraging a new method named SPLITTER to discover fined-grained sequential patterns. Yuan *et al.* [35] address the problem of discovering regions of different activities in a city. The knowledge can help citizens to make decisions, e.g., whether to invest in real estate. Asif *et al.* [36] develop unsupervised learning methods to speed up the prediction in the context of multiple and heterogeneous road traffic. The experimental study reveals that the suggested approach produces better prediction accuracy compared to other forecasting algorithms. A dynamic segmentation model for hotspot detection in multiple levels of the spatial network is proposed in [37]. A shortest path tree pruning algorithm is introduced to filter the irrelevant hotspot detection. This algorithm suffers from the quality of returned paths that are considered hotspot which takes only the shortest paths and not meaningful paths.

B. OUTLIER AND ANOMALY DETECTION FOR URBAN TRAFFIC FLOW

Several surveys on outlier detection algorithms for traffic flow data have been published [7], [12], [38]. Here, we give a short survey, dividing these methods into three categories: statistical approaches, similarity-based approaches, and frequent pattern-based approaches.

1) STATISTICAL APPROACHES

Statistical approaches use statistical models and techniques such as the Gaussian aggregation model [39], principle component analysis [40], stochastic gradient descent [41], or Dirichlet Process Mixture [42]. In general, inlier flows are assumed to follow some common statistical process while the flows that deviate from this statistical mechanism are treated as outliers. Ngan *et al.* [43] used a DPMM (Dirichlet Process Mixture Model) for deriving outliers in urban traffic flow data. First, the set of all flow values $F = \{f_1, f_2, \ldots, f_{|F|}\}$ is projected into an n-dimensional space, where the i^{th} object is defined by the flow values $\{f_i, \ldots, f_{i+n-1}\}$. The obtained dimensions are then reduced by PCA (Principal Component Analysis) to a two-dimensional space. Then, the Chinese restaurant process [44] is performed to cluster the flow values with an infinite number of clusters. Each flow value is assigned to a new cluster with a probability proportional to a concentration parameter α, otherwise, it is assigned to the previously created cluster. Afterwards, all flow values belonging to the cluster having a maximum number of elements are considered inliers, the remaining flow values are outliers. Lin *et al.* [45] introduce an algorithm that uses Gaussian aggregation for road traffic speed prediction. Speed sensing data are first integrated with tweet and trajectory data to enrich the training data. A combination of a disaggregation model and a Gaussian process is then used in the overall framework. This combination allows to improve traffic speed prediction on the expense of computation time, especially when dealing with a high number of vehicles. Lakhina *et al.* [46] study the use of PCA with an algorithm for discovering anomalous flows to explore network-wide traffic data. The approach aims to separate network traffic into a normal component that is dominated by predictable traffic and an anomalous component which is noisier and contains the significant traffic spikes. Ye *et al.* [47] present an anomaly-tolerant traffic matrix estimation approach called SETMADA (Simultaneously Estimate Traffic Matrix and Detect Anomaly). It estimates the traffic matrix and uses it for anomaly detection. Based on the prior low-rank property and temporal characteristic of the traffic flow, the outlier detection is formulated as a prior information-guided matrix completion problem. Nevat *et al.* [48] address the problem of correlation between the anomalous traffic flows. The authors develop a statistical decision theoretic framework based on a Markov chain model [49] for temporally correlated traffic in networks. The anomaly detection problem is reformulated via the generalized likelihood ratio test [50]. A two-step approach is suggested: in the first step the cross entropy [51] is applied to quickly detect anomalous flows, in the second step a transformation of aggregated flows is performed using an efficient low-dimensional representation of the traffic flow.

2) SIMILARITY-BASED APPROACHES

Similarity-based approaches use distance metrics and neighborhood computation methods or classic outlier detection methods [16], [18] to find outliers. In general, the normal flows (inliers) are assumed to build dense regions while outlying flows are assumed to build regions of lower density. Dang *et al.* [52] proposed a combination between kNN [16] and PCA for outlier flow detection. A dimensionality reduction is performed by PCA. In the derived subspaces the kNN outlier detection [16] is applied. Tan *et al.* [53] proposed BLOF algorithm (a density-based bounded LOF) for large scale traffic flow data in Hong Kong. A three dimensional space is derived by PCA, then the LOF algorithm [18] is applied on this reduced space to find local outliers in the flow data. Huang *et al.* [54] proposed a dimensionality reduction algorithm for anomaly detection in traffic data by developing a distance-based subspace measure called DR-SS (Dimensionality Reduction based on Sub-Space measure). This measure aims to find an appropriate reduced subset of dimensions in a multi-dimensional space in different time intervals. Munoz-Organero *et al.* [55] proposed a distance-based algorithm to detect abnormal driving locations caused by particular traffic conditions such as traffic lights, street crossings, or roundabouts. The aim is to filter outlying driving points related to random traffic conditions such as traffic jams from infrastructural road elements. The Mahalanobis distance is used to compute the similarity between the single flows captured each second during 20 seconds. Dense flows with high similarity values are considered as inliers, the others are treated as outliers. Shi *et al.* [56] proposed a dynamic neighborhood-based approach to detect local anomalies in spatio-temporal flow data. The dynamic flow is first represented by the real-time velocity values of vehicles. The dynamic neighborhood structure is then designed by computing the similarity between spatio-temporal flows.

Lee *et al.* [57] proposed the *Athena* framework, a distributed application that allows to detect anomalies from the network flows. A wide range of anomaly detection services based on similarity approaches and network monitoring routines are integrated into this framework. The results reveal that *Athena* outperforms the Spark implementation [58] in terms of the runtime performance using real network flow data.

3) PATTERN MINING-BASED APPROACHES

Pattern mining-based approaches use techniques such as Apriori [59] or FP-growth [60] to discover connections between outliers. Liu *et al.* [61] introduced the problem of causal interaction in traffic data streams, i.e., the discovery of relationships among the detected outliers. The traffic data are first preprocessed by building a region graph. Temporal outliers are then identified based on a distortion function that computes the similarity between segment flows. Association rule mining is performed to extract relationships between the discovered outliers. Pang *et al.* [24] developed a pattern mining-based strategy for spatio-temporal outlier detection. Two kinds of outliers could be detected, persistent outliers and emerging outliers. An upper bounding strategy for both outliers is applied. Chawla *et al.* [25] focus on analyzing the traffic between regions rather than the entire flows. This strategy reduces considerably the computational cost of the proposed model. Nguyen *et al.* [62] predict frequently congested sites in spatio-temporal data and detect causal relationships among them from traffic data streams. A tree of segment flow outliers is constructed for snapshots over time, frequent subtrees are extracted from all trees, where the subtree is selected if its support rate exceeds the given support threshold. This algorithm allows not only to detect flow outliers in a single arc of the network but also to discover congestion patterns. In addition, a dynamic Bayesian network approach is applied to represent the congestion propagation between segment flows.

4) DISCUSSION

The statistical approaches are very sensitive to the outliers, i.e., outliers interfere with the model fitting. Furthermore, they rely on a specific statistical model and it is often not clear whether or not that model reflects well the actual distribution of the given traffic flow data. Similarity-based approaches solve this latter issue by adopting a non-parametric approach. However, these approaches do not deal with correlations between flow data and only try to detect single outlier flow data. Pattern mining-based approaches take into account also correlations between single outliers. On the other hand, these approaches are very time consuming as they are based on the frequent pattern mining process that needs multiple scans of the flow database.

To the best of our knowledge, there is only one approach called FPD-LOF [63], that explores the local outlier factor for finding out the set of distribution of flow outliers. This approach deals only with a single location, where a temporal dimension is used, to construct the distribution of flows, and considers the Bhattacharyya metric, to compute the local reachability distance for each distribution of flows. In order to deal with spatio-temporal data, and based on the success of kNN [64] and KL-divergence distance in a distribution data [65], this paper proposes a novel approach called kNN-FDP to derive the distribution of flow outliers.

To the best of our knowledge, there is only one approach called FDP-LOF [63], that explores the local outlier factor for finding out the set of distribution of flow outliers. This approach deals only with a single location, where a temporal dimension is used, to construct the distribution of flows, and considers the Bhattacharyya metric, to compute the local reachability distance for each distribution of flows. In order to deal with spatio-temporal data, and based on the success of kNN [64] and KL-divergence distance in a distribution data [65], this paper proposes a novel approach called kNN-FDP to derive the distribution of flow outliers.

III. PROPOSED FRAMEWORK

A. PROBLEM STATEMENT

In this paper, we focus on detecting anomalies from the flow distribution probabilities. A flow distribution probability can be regarded as a sequence of pairs of flow and its probability. A flow is defined by the number of objects at each determined time interval. In this investigation, the traffic flows are first determined during each time interval. The frequency of each flow is computed and then the flow distribution probability is derived. Let $L = \{L_1, L_2, \ldots, L_k\}$ be the set of k locations. Each location L_i is featured by the set of $FDP_i = \{FDP_i^1, FDP_i^2, \ldots, FDP_i^r\}$. The outlier FDP detection problem aims to identify outliers from the set $\{FDP_i\}$. In the other terms, it aims to divide the set $\{FDP_i\}$ into two sets (O_i and I_i). The set O_i consists of FDP outliers of the location L_i whereas the set I_i consists of the FDP inliers of the location L_i.

Definition 1: Consider a *score* function, defined as follows:

$$Score: FDP_i \rightarrow R \tag{1}$$

$$FDP_i^j \mapsto Score(FDP_i^j) \tag{2}$$

The outliers and the inliers sets are defined as follows:

$$\begin{cases} O_i = \{FDP_i^j \mid \forall\, FDP_i^l \in I_i, \\ Score(FDP_i^j) \geq Score(FDP_i^l)\} \\ I_i = FDP_i / O_i. \end{cases} \tag{3}$$

B. PRINCIPLE

The proposed framework is overviewed in Figure 2 and it consists of two steps:

1) FDP Construction: This step aims to create the historical flow distribution probability database from the urban traffic data. First, it extracts the information of each location from the traffic flow. Multiple databases are extracted, each of which is assigned to one specified location. Second, it builds the flow's distribution for each location. From each database of the given location, the flow's distribution is computed for a

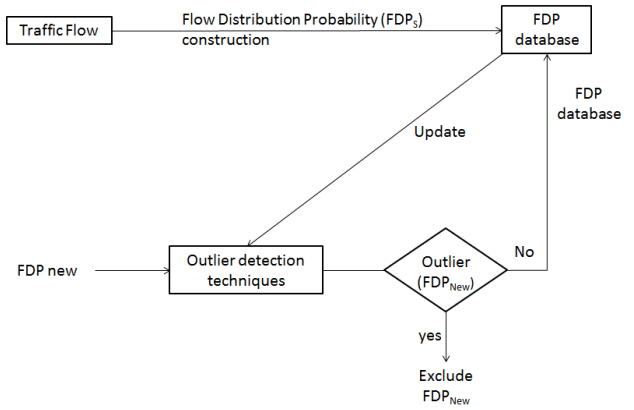


FIGURE 2. Proposed framework for outlier detection on flow distribution probabilities.

given time interval. Multiple historical flow's distribution databases are then created, each one represents the flow's distribution of the given location. Finally, the flow's distribution probability of each location is then determined and stored on an adequate structure.

2) Outlier Detection Technique: The historical flow distribution probabilities are used independently to detect outliers from new flow's distribution coming in a streaming way. Any outlier detection algorithm can be used. Several improvements may be investigated for this purpose. The most frequently used outliers detection techniques are adapted for flow's distribution data. If the new data is an inlier then it is added to the historical data, otherwise, it is excluded from the historical flow's distribution probability database for the next processing.

C. FDP CONSTRUCTION

The goal of this step is to build the flow distribution probability of each location.

Definition 2: Consider $L = \{L_1, \ldots, L_k\}$ is the set of k locations. We define the traffic flow $TF = \{TF_1, \ldots, TF_k\}$, where TF_i is the *traffic flow information* related to the location L_i.

Definition 3: Each location L_i has a set $FDP_i = \{FDP_i^1, \ldots, FDP_i^r\}$, where FDP_i^j represents the j^{th} *flow distribution probability* of the i^{th} location.

Definition 4: Note TFO_i as the *time flow observation* of the location L_i, $[T_{j-1}, T_j]$ the *time interval* of the FDP_i^j. We define NFO_i^j as the number of flow observations of the FDP_i^j by

$$NFO_i^j = \frac{T_j - T_{j-1}}{TFO_i}. \tag{4}$$

Definition 5: We define the l^{th} flow in FDP_i^j by the set of objects that across the location L_i in the time between $(T_j + (TFO_i \times (l-1)))$ and $(T_j + (TFO_i \times l))$ and we obtain $F_i^{j,l} = \{TF_i^r, |TF_i^r(time) \in [(T_j + (TFO_i \times (l-1))), (T_j + (TFO_i \times l))]\}$.

Definition 6: We define $F_i^{j,l}$ by the number of objects (e.g., pedestrians, bicycles, cars, trucks, buses) that cross a location L_i during some time interval $[i \times j, (i \times j) + 1]$ by means of various types of sensors in streets, in traffic light systems, or as mobile sensors. The maximum flow max_i^j of the j^{th} flow distribution in the i^{th} location is defined by

$$max_i^j = \{|F_i^{j,l}|, |\forall l' \in [1 - NFO_i^j], |F_i^{j,l}| \geq |F_i^{j,l'}|\} \tag{5}$$

Definition 7: We define the flow frequency at level m, i.e., the set of flows that occurs m times as

$$FF_i^j(m) = \{F_i^{j,l}|, |F_i^{j,l}| = m\} \tag{6}$$

Definition 8: We define the probability flow at level m, i.e., the probability flows that occurs m times as

$$FP_i^j(m) = \frac{|FF_i^j(m)|}{|NFO_i^j|} \tag{7}$$

Definition 9: We define FDP_j^i as

$$FDP_j^i = \{FP_i^j(m), |m \in [1, max_i^j]\} \tag{8}$$

The FDP database of each location is constructed using the definitions (Def. 2 to 9). Assume we have the traffic flow database $TF = \{TF_1, TF_2, \ldots, TF_k\}$ of k locations $L = \{L_1, L_2, \ldots, L_k\}$. Each location has time flow observation TFO_i, r FDPs, $\{FDP_i^1, FDP_i^2, \ldots, FDP_i^r\}$, the j^{th} flow distribution probability contains observations from the interval time $[T_{j-1}, T_j]$ (Def. 2, 3, 4). The number of flow observation is first determined of each flow distribution at each location using Def. 4. A single flow observation is then computed using Def. 5. The maximum flow of each observation is calculated using Def. 6. The flow frequency of each single observation is given using Def. 7. The probability flows of each location are finally extracted using Def. 8 and 9.

D. OUTLIER DETECTION TECHNIQUE

In this section, a kNN-FDP algorithm that adapts the k Nearest Neighbor algorithm *kNN* [16] is developed for detecting flow distribution probability outliers. Before introducing the algorithm, we show how to compute the distance similarity between two FDPs, i.e., the similarity function in *kNN*.

Definition 10: Consider FDP_i^j the j^{th} flow of the i^{th} location defined by Def.9. We define the vector A_i^j representing the $d_j - dimensional$ space of FDP_i^j by

$$\begin{cases} d_j = max_i^j \\ A_i^j[m] = FP_i^j(m) \quad \forall m \in [1, max_i^j] \end{cases} \tag{9}$$

Proposition 1: Consider two vectors defined by Def. 10, $(A_i^{j_1} and A_i^{j_2})$ with $d_{j_1} \geq d_{j_2}$. Then, d_{j_2} is transformed to d_{j_1} by setting all the missing values of $A_i^{j_2}$ to 0 to obtain:

$$\begin{cases} A_i^{j_2}(m) = A_i^{j_2} & m \in [1, max_i^{j_2}] \\ \quad\quad\quad\quad 0 & \forall m \in [max_i^{j_2} + 1, max_i^{j_1}] \end{cases} \tag{10}$$

Algorithm 1 kNN-FDP Algorithm

1: **Input**: $FDP_i = \{FDP_i^1, \ldots, FDP_i^r\}$: The FDP of the i^{th} location.
FDP_i^{new}: The novel FDP observed in the i^{th} location.
ϵ: ratio threshold.
k: The kNN threshold.
2: **Output**: Outlier: Boolean that indicates FDP_i^{new} is outlier or not.
3: **for** j=1 to r **do**
4: dist[j] $\leftarrow$ KL(FDP_i^{new}, FDP_i^j)
5: **end for**
6: d $\leftarrow$ kNN(dist)
7: **if** d $\leq \epsilon$ **then**
8: Outlier $\leftarrow$ false
9: **else**
10: Outlier $\leftarrow$ true
11: **end if**
12: **return** Outlier

In this paper, the KL-divergence (Kullback-Leibler divergence) [31] will be explored. The KL-divergence distance is chosen because it is the most used for probability similarity computation. In the following, we adapt the KL-divergence (KL) for FDP similarity computation as

$$KL(\mathcal{A}_i^{j_1}, \mathcal{A}_i^{j_2}) = \sum_{m=1}^{d_{j_1}} \mathcal{A}_i^{j_1}(m) \ln \frac{2\mathcal{A}_i^{j_1}(m)}{\mathcal{A}_i^{j_1}(m) + \mathcal{A}_i^{j_2}(m)} \quad (11)$$

Algorithm 1 describes the kNN-FDP algorithm that adapts the kNN outlier algorithm presented in [16]. kNN-FDP has as input the flow distribution probability of the i^{th} location, FDP_i, the novel FDP FDP_i^{new}, k and ϵ thresholds. It also uses an internal data structure represented by a vector *dist* to store the distance values. The algorithm returns a boolean variable that indicates whether FDP_i^{new} is an outlier or not. First, the distance between FDP_i^{new} and each FDP in FDP_i^j is determined (line 3 to 5). The distance value between FDP_i^{new} and its k^{th} nearest neighbor is selected using KL-divergence (line 6). If this value exceeds the ϵ threshold, then FDP_i^{new} will be considered as an outlier, otherwise, it will be considered as an inlier (line 7 to 11).

E. COMPLEXITY

The theoretical complexity cost of the proposed framework is divided into the following costs:

1) FDP construction cost: The flow is first built from the traffic flow on each location. This operation requires $|TF_i|$ scans of each location i, so, this operation needs $\sum_{i=1}^{k} |TF_i|$ scans. The FDP database is then designed from the flows, which needs to scan the entire flows, and requires $|F_i|$ scans for each location L_i. This operation requires $\sum_{i=1}^{k} |F_i|$ scans. Thus, the cost of FDP construction is $\sum_{i=1}^{k} (|TF_i| + |F_i|)$.
2) Outlier detection cost: The two outlier detection techniques presented in this paper mainly depend on the similarity metric used. The complexity cost of the adopted metrics in this paper is $O(d_i)$ where d_i is the number of dimensions on the FDP space, i.e., the maximum number of flows of all FDP at each location L_i. The kNN-FDP algorithm then requires $O(|FDP_i|)$ where $|FDP_i|$ is the size of the historical database of the location L_i. The complexity cost of this operation is $\sum_{i=1}^{k} (|FDP_i| \times d_i)$.

The complexity cost of the proposed framework is $\sum_{i=1}^{k} ((|TF_i| + |F_i|) + (|FDP_i| \times d_i))$, where k is the number of locations, $|FDP_i|$ is the size of the historical FDP database of the location L_i, and d_i is the maximum number of flows in FDP of each location L_i.

F. ILLUSTRATION

Consider the following traffic flow TF_1 related to the location L_1

Object	O_1	O_2	O_3	O_4	O_5
Time	0.2	0.4	0.8	1.2	1.4
Object	O_6	O_7	O_8	O_9	O_{10}
Time	1.9	2.3	2.5	3.9	4.5
Object	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}
Time	5.6	5.8	6.1	7.5	7.9
Object	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}
Time	8.1	8.3	8.5	8.9	9.5

Let us consider the TFO_1 set to 1, $FDP_1 = \{FDP_1^1, FDP_1^2\}$, which their time interval as

$[T_0 - T_1] = [0-5]$ representing the time interval of FDP_1^1,
$[T_1 - T_2] = [5-10]$ representing the time interval of FDP_1^2.
NFO_1^j is calculated using Def. 4 $\forall j \in [1-2]$ as
$NFO_1^1 = \frac{T_1 - T_0}{TFO_1} = 5$.
$NFO_1^2 = \frac{T_2 - T_1}{TFO_1} = 5$.
$F_i^{j,l}$ is defined (see Def. 5) $\forall j \in [1-2]$ and $\forall l \in [1-5]$ as

$$F_1^{1,1} = \{O_1, O_2, O_3\}, \quad F_1^{1,2} = \{O_4, O_5, O_6\}$$
$$F_1^{1,3} = \{O_7, O_8\}, \quad F_1^{1,4} = \{O_9\}$$
$$F_1^{1,5} = \{O_{10}, O_{11}\}.$$
$$F_1^{2,1} = \{O_{11}, O_{12}\}, \quad F_1^{2,2} = \{O_{13}, O_{14}\}$$
$$F_1^{2,3} = \{O_{15}, O_{16}\}, \quad F_1^{2,4} = \{O_{17}, O_{18}, O_{19}\}$$
$$F_1^{2,5} = \{O_{20}\}.$$

Using Def. 6, the maximum flow max_1^1 and max_1^2 are 3 and 3, respectively.

The flow frequency is given using Def. 7 as
$FF_1^1(1) = \{F_1^{1,4}\}$, $FF_1^1(2) = \{F_1^{1,3}, F_1^{1,5}\}$, and $FF_1^1(3) = \{F_1^{1,1}, F_1^{1,2}\}$.
$FF_1^2(1) = \{F_1^{2,5}\}$, $FF_1^2(2) = \{F_1^{2,1}, F_1^{2,2}, F_1^{2,3}\}$, and $FF_1^2(3) = \{F_1^{2,4}\}$.

The probability flow is thus determined using Def. 8 and 9 as $FDP_1^1 = \{FP_1^1(1), FP_1^1(2), FP_1^1(3)\}$ with

$FP_1^1(1) = \frac{|FF_1^1(1)|}{NFO_1^1} = 0.2,$

$FP_1^1(2) = \frac{|FF_1^1(2)|}{NFO_1^1} = 0.4,$ and

$FP_1^1(3) = \frac{|FF_1^1(3)|}{NFO_1^1} = 0.4.$

$FDP_1^2 = \{FP_1^2(1), FP_1^2(2), FP_1^2(3)\}$ with

$FP_1^2(1) = \frac{|FF_1^2(1)|}{NFO_1^2} = 0.2,$

$FP_1^2(2) = \frac{|FF_1^2(2)|}{NFO_1^2} = 0.6,$ and

$FP_1^2(3) = \frac{|FF_1^2(3)|}{NFO_1^2} = 0.2.$

Consider now the novel FDP observed in the first location as $FDP_1^{new} = \{FP_1^{new}(1), FP_1^{new}(2)\}$ with

$FP_1^{new}(1) = 0.9$, and $FP_1^{new}(2) = 0.1$.

To determine if FDP_1^{new} is outlier or not, we first create the vectors A_1^1, A_1^2, and A_1^{new} using Def. 10 and Prop. 1 as

$A_1^1 = \{0.2, 0.4, 0.4\}$, and $A_2^2 = \{0.2, 0.6, 0.2\}$, and

$A_1^{new} = \{0.9, 0.1, 0.0\}$

The distance between A_1^1, A_1^2 and A_1^{new} is then computed. For instance, using the Bhattaharyyan metric, we get:

$$KL(A_1^1, A_1^2) = 0.51.$$
$$KL(A_1^1, A_1^{new}) = 0.75.$$
$$KL(A_1^2, A_1^{new}) = 0.71.$$

If k is set to 1, then $kNN(A_1^{new}) = \{A_1^2\}$, and $kNN(A_1^2) = \{A_1^1\}$. The distance between A_1^{new} and A_1^2 is 0.71. If ϵ is set to 0.5, then FDP_1^{new} is an outlier.

IV. PERFORMANCE EVALUATION

A number of experiments have been carried out to demonstrate the performance of the proposed framework using real traffic flow data from Odense, Denmark, and Beijing, China. We first present the Odense traffic flow data and describe the configuration of the framework on each location. The output of the best configuration is then shown to detect separately outliers on each location. All codes are scripted on Java and JavaScript, run on an Intel Core i7, whereas all plots are done using RStudio. The source codes can be downloaded from[1] to facilitate the reproducibility of the experiments. In the following experiments, the interval time of determining each FDP is fixed to one hour, which is the most standard interval used by the traffic flow community.

A. DATA

In this experiment, two real urban traffic are used:

1) The first one is retrieved from Odense Kommune (Denmark)[2] is shown. The data is a set of rows, where each row contains information related to the cars detected at specific locations such as gap, length, location, datetime, speed, and class. The location is represented by latitude and longitude. The speed is calculated as km/h, and the datetime represents the year, the month, the day, the hour, the minute and the second that the car is passed by the given location. The most important information in each car is given as follows:
   - datetime: represents the time that the car passed on the location. Datetime is: YYYY-MM-DD hh:mm:ss.
   - latitude: defines the first dimension (horizontal position) of the location.
   - longitude: defines the second dimension (vertical position) of the location.
   - speed: defines the actual speed of the car where it across the location.
   - class: defines the type of vehicle, e.g. class is set to 2 represents a passenger car.

   IT infrastructure is installed to detect the cars passed on each location. In this study, we focus on ten locations described in Table 1. The traffic data input is obtained from Odense flow that is observed between 1^{st} January 2017 and 30^{th} September 2017. The global view of the ten locations is given by the map presented in Figure 3.
2) The second one is a real urban traffic data obtained from Beijing traffic flow, and retrieved from.[3]

TABLE 1. Odense data description.

Category	ID	Location Name	(Latitude, Longitude)	Flow File Size (KB)
Non-Dense	L_1	Anderupvej	(55.4383, 10.3896)	3.489
	L_2	Falen	(55.3868, 10.3569)	4.129
	L_3	AlÄŷkke Alle	(55.4036, 10.3682)	9.853
	L_4	Thomas B.	(55.4011, 10.3908)	12.503
	L_5	Niels Bohrs Alle	(55.3753, 10.4570)	16.982
	L_6	Rodegadesvej	(55.3854, 10.4168)	24.146
Dense	L_7	Rugardsvej	(55.4631, 10.0879)	96.153
	L_8	Nyborgvej	(55.0628, 10.6138)	97.307
	L_9	GrÄŷnlandsgade	(55.4009, 10.4020)	122.173
	L_{10}	Odins Bro	(55.4222, 10.3803)	161.482

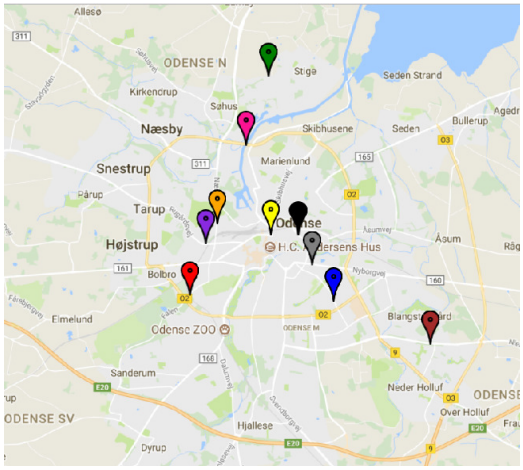


FIGURE 3. Location's Map: L_1 –green–, L_2 –red–, L_3 –orange–, L_4 –yellow–, L_5 –brown–, L_6 –blue–, L_7 –Violet–, L_8 –grey–, L_9 –black–, L_{10} –pink–.

[1] https://sites.google.com/site/youcefdjenouri/software

[2] http://dss.sdu.dk/projects/its.html

[3] https://www.beijingcitylab.com/

It consists of more than 900 million traffic flow values during a two-months time period on one location. The most important information of each car is given as follows: Datetime: represents the time that the car passed on the location, the datetime format is: YYYY-MM-DD hh:mm:ss. Class: It defines the type of vehicle or bus.

B. EVALUATION

The common problem in the evaluation of outlier detection techniques using new data is how to derive the outlier set from the inlier set. To solve this problem, virtual outlier flows are generated by simulating usual flows. In this experiment, four unusual flows are defined and generated as:

1) Null FDP: In the null FDP, the flow distribution is equal to 0, whatever, the number of the flow. In other words, we detect any flow during the observation, The definition of this outlier flow is presented as

$$FDP_i^j(m) = \{0 \mid \forall\, m \in [1, \ldots, max_i^j]\} \tag{12}$$

2) Stable FDP: In the stable FDP, the flow distribution is equal to 1 for flow equal to x, 0 otherwise. In other words, the flow is stable at x, its definition is presented as

$$FDP_i^j(m) = \begin{cases} 1 & m = x \\ 0 & Otherwise \end{cases} \tag{13}$$

3) Regular FDP: The flow here is equally distributed, it is defined as

$$FDP_i^j(m) = \{\frac{1}{|FDP_i^j|} \mid \forall\, m \in [1, \ldots, max_i^j]\} \tag{14}$$

4) Unexpected FDP: It is observed when an unexpected event occurs, such as city events or road accidents: It is performed into three main stages, stable flow from 1 to x, a cumulative flow from x to y, After, a Null flow from y to max_i^j, it is defined as

$$FDP_i^j(m) = \begin{cases} \epsilon & m \in [1, \ldots, x] \\ \Psi(m) & m \in [x, \ldots, y] \\ 0 & m \in [y, \ldots, max_i^j] \end{cases} \tag{15}$$

where $\Psi(m)$ is a function defined from $[x, \ldots, y]$ to $[\epsilon, \ldots, (1 - x\epsilon)]$, and described as

$$\begin{cases} \forall (m_1, m_2), m_1 \geq m_2 \iff \Psi(m_1) \geq \Psi(m_2) \\ \sum m\Psi(m) = (1 - x\epsilon) \end{cases} \tag{16}$$

Moreover, a set of noise FDPs are generated using Gaussian noise as in [63]. Figure 4 presents an example of generated noise FDP and original FDP for each location.

The ground truth is all FDPs generated including null, stable, regular, unexpected and noise FDPs. The evaluation is performed using the F-measure, which is defined as:

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{17}$$

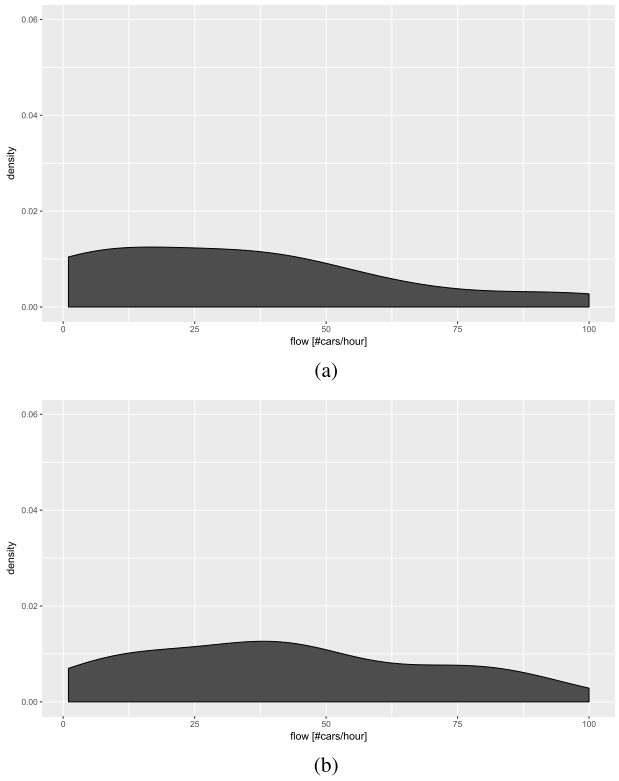


FIGURE 4. Noise FDPs (example). (a) Original FDP. (b) Noise FDP.

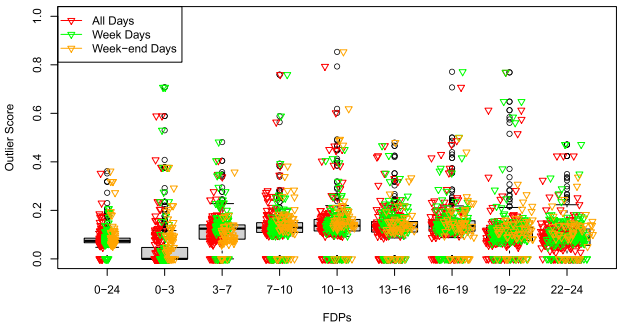


FIGURE 5. Boxplot of Outlier Scores of all FDPs in the Anderupjev Location.

$$Recall(A) = \frac{|O_A \bigcap O|}{|O|} \tag{18}$$

$$Precision(A) = \frac{|O_A \bigcap O|}{|O_A|} \tag{19}$$

O: The set of all outliers.
O_A: The set of outliers returned by the scenario A.
I_A: The set of inliers returned by the scenario A.

C. RESULTS ON ODENSE DATA

Figure 6 presents the boxplot of Outlier Scores of all FDPs in the Anderupjev Location. By varying the interval time (from 0 to 24), and the type of days (all days, weekdays, and weekend days), we can observe that it exists many outliers for the weekdays compared to the weekend days. Moreover, there are many outliers between $[7-10]$, $[10-13]$, $[16-19]$, and $[19-22]$. We can explain this result by the fact that many

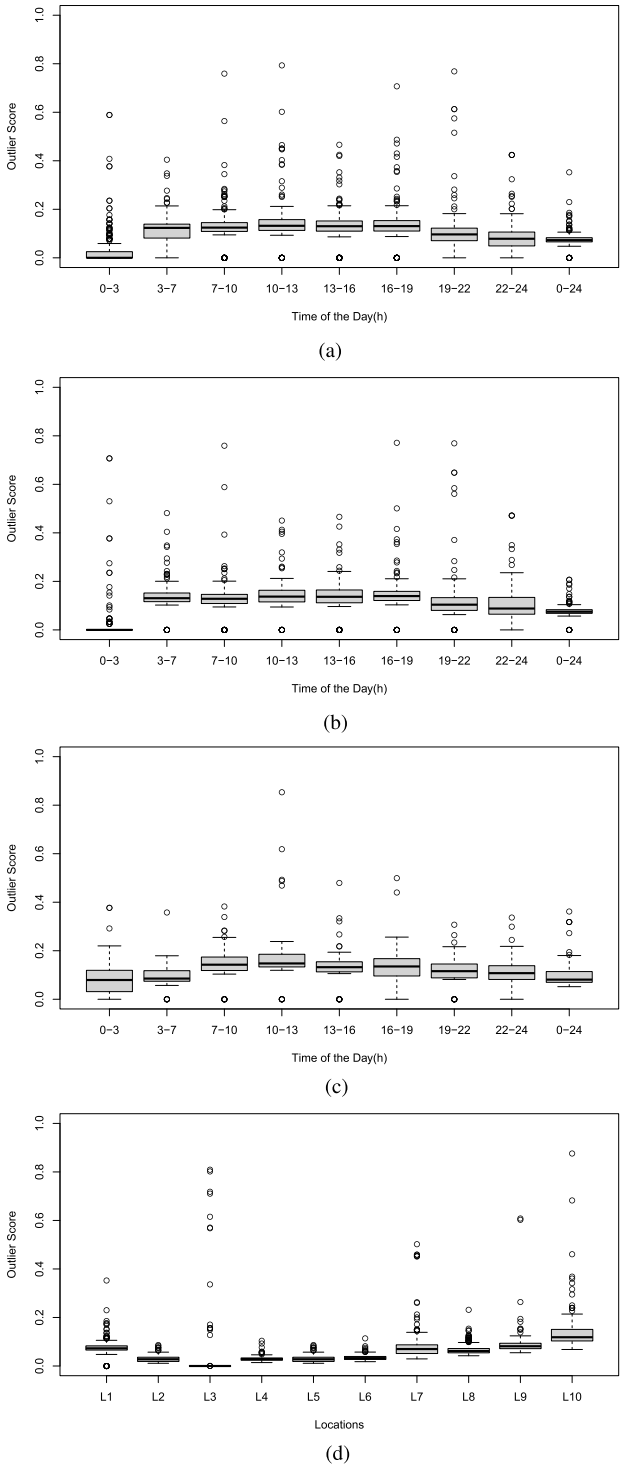


FIGURE 6. Boxplot of Outlier Scores. (a) FDPs of All Days of Anderpvej. (b) FDPs of Weekdays of Anderpvej. (c) FDPs of Weekend Days of Anderpvej. (d) Boxplot of Outlier Scores of all FDPs in all locations.

people go to work in the weekdays between $[7-10]$ (usually at 9:00 in Denmark) go to lunch between $[10-13]$ (usually at 11:30 in Denmark) and return home between $[16-19]$ (usually at 17:00 in Denmark). Figure 5 shows the boxplot of Outlier Scores of all FDPs in all locations. We remark that the number of outliers differs from location to another depending on the type of the location (dense or non-dense). For non-dense locations, there are few outliers except the location L_3. However, for dense locations, we observe many outliers, except at the location L_8.

TABLE 2. Description of the most frequently returned outliers of all locations.

Outliers DD-MMM-YYYY	Description
01-01-2017	The first day of the year.
08-02-2017	Biweekly Farmer's Market in Odense
09-02-2017	Biweekly Farmer's Market in Odense
14-02-2017	Saint valentine's day
23-02-2017	World Cup 2018 TOLT event (Sport, national event)
08-03-2017	Women's day
04-04-2017	Hans Christian Andersen's Birthday (For children, Sport)

In Figure 7 we depict the top three FDP outliers. Figures 7(a), 7(b), and 7(c) returned by FDP-LOF and the remaining FDPs, Figure 7(d), on location L_5. It is apparent that the top three outliers are very different from the majority of FDPs. The density of these outlier FDPs does not exceed 0.02. Location L_5 is a low-traffic location, where the flow values are rather small (flow values between 0 and 25 are the most abundant). The outliers in this location show a more even distribution with a larger amount of high flow values than the majority of the FDPs.

Table 2 show the interpretation of outliers for each location. According to this table, we can conclude that there are seven most frequent outliers repeated in the ten locations, for instance, the first day of the year (01-01-2017) repeated in three locations, the Women's day (08-03-2017) also repeated in three locations. We can justify the first case by the fact that people stay at home and take some reset after a celebration at the last night of 2016. However, we can justify the second case by the fact that women celebrate their day in public places (restaurant, cinemas, theaters, etc). There are also outliers caused by particular events in Odense like the World Cup 2018 TOLT hold on 23-02-2017. Detailed information of events in Odense city can be accessed at.[4]

Figure 8 presents the F-measure of kNN-FDP on Odins Bro location with different number of neighbors. By varying the number of neighbors from 1 to 10, the F-measure augments to 77% until the number of neighbors equals to 8, and then reduces to 72% for 10 neighbors. Similarly, Figure 9 presents the F-measure of kNN-FDP on Odins Bro location with different mining threshold values. By varying the mining threshold from 0.1 to 1, the F-measure increases to 80% until the mining threshold equal to 7, and then reduces to 74% for a mining threshold set to 1.0. From these experiments, we can conclude that kNN-FDP is sensitive to the number of neighbors, and the mining threshold values. Thus, it is crucial to choose the suitable values of these two parameters for each dataset.

The last experiment of this part aims to compare kNN-FDP with LOF-FDP [63]. Figure 10 shows in terms of

[4]www.visitodense.com

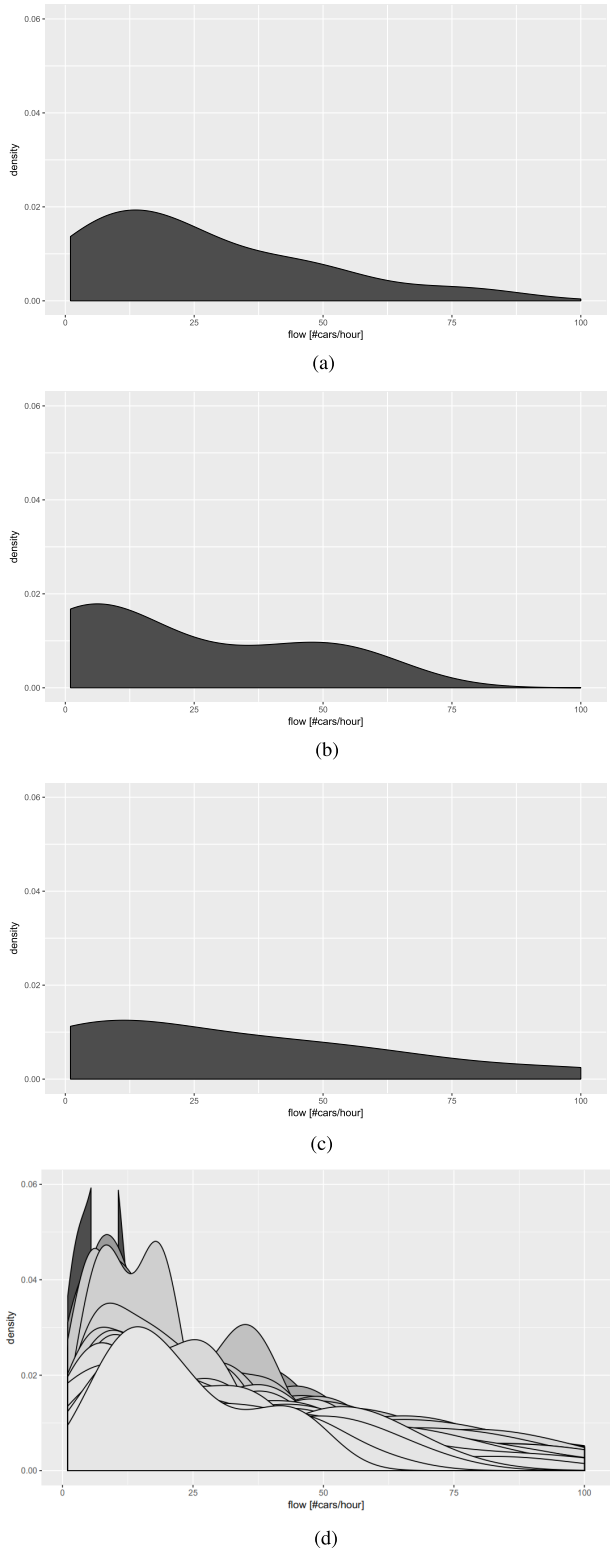


FIGURE 7. Comparison between the top three FDP outliers and the remaining FDPs (location L_5). (a) Top outlier. (b) Second outlier. (c) Third outlier. (d) Remaining FDPs.

the percentage of detected outliers of both kNN-FDP with LOF-FDP using ten Odense locations. The results reveal that kNN-FDP outperforms LOF-FDP in almost all locations.

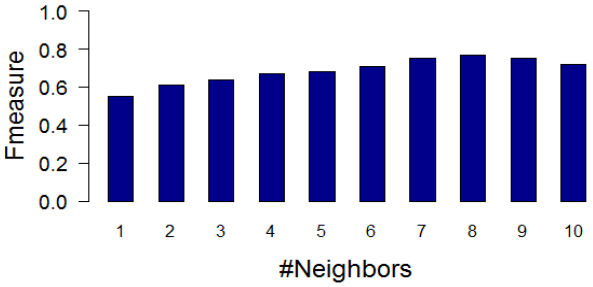


FIGURE 8. F-measure of the kNN-FDP on Odins Bro location with different number of neighbors.

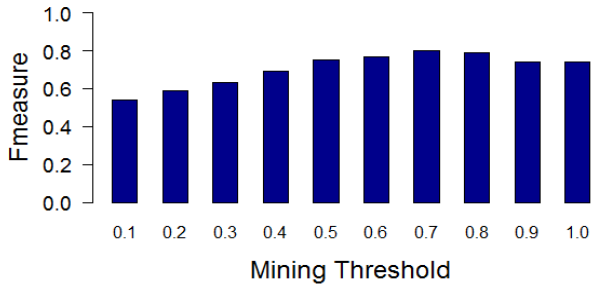


FIGURE 9. F-measure of the kNN-FDP on Odins Bro location with different mining threshold.

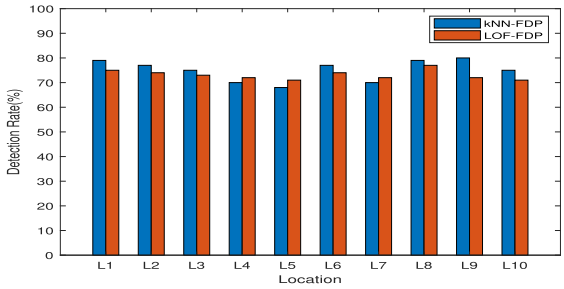


FIGURE 10. kNN-FDP Vs LOF-FDP on Odense locations.

The reasons of these results are i) The KL-divergence distance is more adequate than the Bhattacharyya distance for computing similarity between FDPs, and ii) Thanks to our framework which updates the historical FDP by new inliers FDPs. This enrichment influences positively to the quality of outliers obtained.

D. RESULTS ON BEIJING DATA

The aim of the experiments in this section is to show the performance of our approach using big datasets such as Beijing data. We compare our approach with the baseline methods (DPMM [43], PCA [46], and SETMADA [47]). Figure 11 shows the runtime in seconds of the kNN-FDP, and the baseline algorithms (DPMM, PCA, and SETMADA) using Beijing data. By varying the number of flows in million from 100 to 900, kNN-FDP outperforms the baseline algorithms. This result is obtained thanks to the kNN computation much faster than the other algorithms. Figure 12 shows the F-measure of the kNN-FDP, and the baseline algorithms

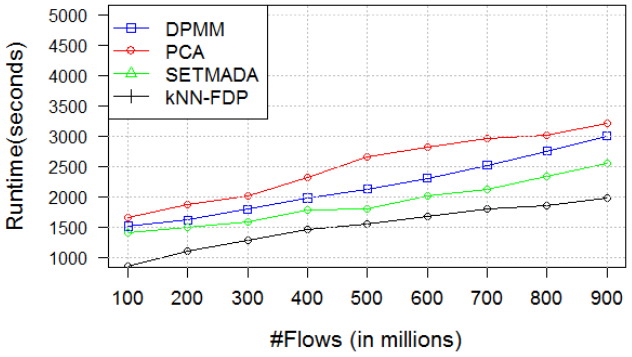


FIGURE 11. Runtime (seconds) of kNN-FDP, and the state-of-the-art urban traffic flow algorithms on Beijing data.

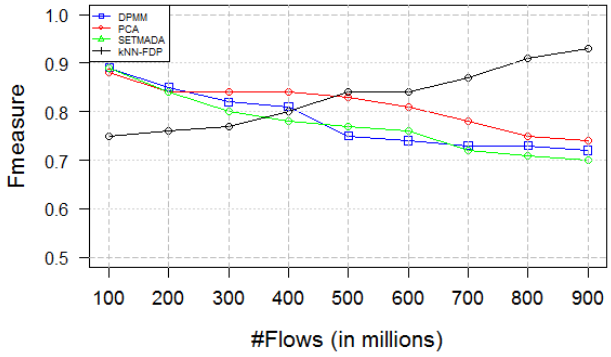


FIGURE 12. F-measure of kNN-FDP and the state-of-the-art urban traffic flow algorithms on Beijing data.

(DPMM, PCA, and SETMADA) using Beijing data. By varying the number of flows in million from 100 to 900, the F-measure of the kNN-FDP increases, while the F-measure of the other approaches decreases. Furthermore, up to 500 million of flows, kNN-FDP outperforms the baseline algorithms in terms of F-measure. These results are obtained thanks to the enrichment phase that adds the new inliers in the training databases.

E. DISCUSSION

For the sake of conciseness, in the remainder of this section, we discuss the main research findings from the application of our approach to Odense and Beijing real traffic data.

- The first finding of our study is that the number and the quality of the outliers differ from location to another. They also differ from the time of day to another, for instance, using the data of the *Anderupjev* location, we found many outliers in two interval times $[10-13]$ and $[16-19]$. Moreover, the number of outliers in *Odins-Bro* is more interesting compared to the outliers in *Anderupjev*.
- Based on a data level, our approach is the first approach in the literature, that considers the distribution of the flows in the outlier detection of spatio-temporal data. Moreover, our approach is able to outperform the baseline urban traffic flow algorithms for dealing with the challenging Beijing large-scale data.
- Being based on an architecture level, our approach is typically able to deal with streaming data by employing the enrichment phase that adds the new inliers in the training databases. In the context of spatio-temporal outlier detection techniques, we argue that the current tools do not deal with this primordial issue.
- Being based on a conceptual level, kNN-FDP is sensitive to the number of neighbors and the mining threshold. Selecting suitable values for these two parameters is crucial for each location.
- From a data mining research standpoint, our paper is an example of the application of a generic data mining technique to a specific context. The literature calls for this type of research, particularly in the times of massive spatio-temporal data, where increasingly large amounts of data are available in different locations and at different times. As in many other cases, porting a pure data mining technique into a specific application domain requires methodological refinement and adaptation. In this context, we argue that our approach benefits from the knowledge extracted in the refinement step that shifts the intelligence required for identifying the outlier flow distribution probabilities from traffic flow data on each location.

V. CONCLUSION

This paper introduced a new framework for outlier flow distribution probability detection. It performs on two steps: i) The FDP databases are first built using both spatial and temporal traffic flow information. ii) The outlier detection process is established to the coming FDP, the inliers are kept to enrich the FDP databases while the outliers are deleted. The kNN algorithm has been adapted for FDP outlier detection, this arises a new algorithm kNN-FDP. The KL-divergence distance is also investigated to compute the similarities between two FDPs. To demonstrate the performance of the suggested framework, several experiments have been carried out using nine months Odense traffic flow shared at ten different locations, and the challenging Beijing large-scale data. The results reveal that the proposed framework is able to detect real distribution of flow outliers. Moreover, it outperforms the baseline urban traffic flow algorithms on high urban traffic flow. As a perspective, we plan to investigate other outlier detection techniques to deal with flow distribution probability anomalies. We are also planning to apply other data mining techniques for flow distribution probability. Finally, proposing a parallel version that explores high-performance computing to launch the proposed framework on many locations in real time context is also in our agenda.

REFERENCES

[1] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, no. 4, p. 83, 2018.

[2] A. Lausch, A. Schmidt, and L. Tischendorf, "Data mining and linked open data—New perspectives for data analysis in environmental research," *Ecolog. Model.*, vol. 295, pp. 5–17, Jan. 2015.

[3] V. Kumar, "Big data in climate: Opportunities and challenges for machine learning," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, p. 3.
[4] R. Elkadiri *et al.*, "Development of a coupled spatiotemporal algal bloom model for coastal areas: A remote sensing and data mining-based approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 11, pp. 5159–5171, Nov. 2016.
[5] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Data mining techniques on satellite images for discovery of risk areas," *Expert Syst. Appl.*, vol. 72, pp. 443–456, Apr. 2017.
[6] M. B. Jensen, M. P. Philipsen, M. Trivedi, T. Møgelmose, and T. Moeslund, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1800–1815, Jul. 2016.
[7] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.
[8] C. Zhang, Q. Yuan, and J. Han, "Bringing semantics to spatiotemporal data mining: Challenges, methods, and applications," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1455–1458.
[9] S. Shekhar, M. R. Evans, J. M. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 193–214, 2011.
[10] X. Zhou, S. Shekhar, and R. Y. Ali, "Spatiotemporal change footprint pattern discovery: An inter-disciplinary survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 4, no. 1, pp. 1–23, 2014.
[11] K. Koperski, J. Adhikary, and J. Han, "Spatial data mining: Progress and challenges survey paper," in *Proc. ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discover*, 1996, pp. 1–10.
[12] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.
[13] S. Shekhar *et al.*, "Spatiotemporal data mining: A computational perspective," *ISPRS Int. J. Geo-Inf.*, vol. 4, no. 4, pp. 2306–2338, 2015.
[14] B. Barz, E. Rodner, Y. G. Garcia, and J. Denzler, "Detecting regions of maximal divergence for spatio-temporal anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[15] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, 2015.
[16] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.
[17] F. Angiulli and C. Pizzuti, "Outlier mining in large high-dimensional data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 203–215, Feb. 2005.
[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2000, vol. 29, no. 2, pp. 93–104.
[19] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, 2014.
[20] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 444–452.
[21] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 1, p. 5, 2015.
[22] G. O. Campos *et al.*, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, 2016.
[23] Y. Djenouri and A. Zimek, "Outlier detection in urban traffic data," in *Proc. Int. Conf. Web Intell., Mining Semantics*, 2018, p. 3.
[24] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng, "On mining anomalous patterns in road traffic streams," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2011, pp. 237–251.
[25] S. Chawla, Y. Zheng, and J. Hu, "Inferring the root cause in road traffic anomalies," in *Proc. 12th Int. Conf. Data Mining*, Dec. 2012, pp. 141–150.
[26] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Inf. Sci.*, vol. 348, pp. 243–271, Jun. 2016.
[27] D. Sun, H. Zhao, H. Yue, M. Zhao, S. Cheng, and W. Han, "ST TD outlier detection," *IET Intell. Transport Syst.*, vol. 11, no. 4, pp. 203–211, May 2017.
[28] D. Ma, X. Luo, S. Jin, D. Wang, W. Guo, and F. Wang, "Lane-based saturation degree estimation for signalized intersections using travel time data," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 136–148, 2017.
[29] Z. He, L. Zheng, P. Chen, and W. Guan, "Mapping to cells: A simple method to extract traffic dynamics from probe vehicle data," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 3, pp. 252–267, 2017.
[30] D. Ma, D. Wang, Y. Bie, S. Jin, and Z. Mei, "Identification of spillovers in urban street networks based on upstream fixed traffic data," *KSCE J. Civil Eng.*, vol. 18, no. 5, pp. 1539–1547, 2014.
[31] D. Johnson and S. Sinanovic, "Symmetrizing the Kullback-Leibler distance," *IEEE Trans. Inf. Theory*, 2001.
[32] T. von Landesberger, F. Brodkorb, P. Roskosch, N. Andrienko, G. Andrienko, and A. Kerren, "MobilityGraphs: Visual analysis of mass mobility dynamics via Spatio-temporal graphs and clustering," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 11–20, Jan. 2016.
[33] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1967, vol. 1, no. 14, pp. 281–297.
[34] K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou, "Online discovery of gathering patterns over trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1974–1988, Aug. 2014.
[35] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 186–194.
[36] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transp. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, Feb. 2014.
[37] X. Tang, E. Eftelioglu, D. Oliver, and S. Shekhar, "Significant linear hotspot discovery," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 140–153, Jun. 2017.
[38] X. Li, Z. Li, J. Han, and J.-G. Lee, "Temporal outlier detection in vehicle traffic data," in *Proc. IEEE 25th Int. Conf. Data Eng.*, Mar./Apr. 2009, pp. 1319–1322.
[39] F. Bunea, A. B. Tsybakov, and M. H. Wegkamp, "Aggregation for Gaussian regression," *Ann. Statist.*, vol. 35, no. 4, pp. 1674–1697, 2007.
[40] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.
[41] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 315–323.
[42] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational Dirichlet process mixture models," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 7, 2007, pp. 2796–2801.
[43] H. Y. Ngan, N. H. Yung, and A. G. Yeh, "Outlier detection in traffic data based on the Dirichlet process mixture model," *IET Intell. Transport Syst.*, vol. 9, no. 7, pp. 773–781, Sep. 2015.
[44] A. Ahmed and E. Xing, "Dynamic non-parametric mixture models and the recurrent chinese restaurant process: With applications to evolutionary clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 219–230.
[45] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2018.
[46] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2004, pp. 219–230.
[47] W. Ye, L. Chen, G. Yang, H. Dai, and F. Xiao, "Anomaly-tolerant traffic matrix estimation via prior information guided matrix completion," *IEEE Access*, vol. 5, pp. 3172–3182, 2017.
[48] I. Nevat *et al.*, "Anomaly detection and attribution in networks with temporally correlated traffic," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 131–144, Feb. 2017.
[49] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
[50] L. Xu and J. Li, "Iterative generalized-likelihood ratio test for MIMO radar," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2375–2385, Jun. 2007.
[51] Z. Wu, M. Kolonko, and R. H. Möhring, "Stochastic runtime analysis of the cross-entropy algorithm," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 616–628, Aug. 2017.
[52] T. T. Dang, H. Y. T. Ngan, and W. Liu, "Distance-based k-nearest neighbors outlier detection method in large-scale traffic data," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, Jul. 2015, pp. 507–510.

[53] J. Tang and H. Y. Ngan, "Traffic outlier detection by density-based bounded local outlier factors," *Inf. Technol. Ind.*, vol. 4, no. 1, pp. 6–18, 2016.
[54] T. Huang, H. Sethu, and N. Kandasamy, "A new approach to dimensionality reduction for anomaly detection in data traffic," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 651–665, Sep. 2016.
[55] M. Munoz-Organero, R. Ruiz-Blaquez, and L. Sánchez-Fernández, "Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving," *Comput., Environ. Urban Syst.*, vol. 98, pp. 1–8, Mar. 2017.
[56] Y. Shi, M. Deng, X. Yang, and J. Gong, "Detecting anomalies in spatio-temporal flow data by constructing dynamic neighbourhoods," *Comput., Environ. Urban Syst.*, vol. 67, pp. 80–96, Jan. 2018.
[57] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran, "Athena: A framework for scalable anomaly detection in software-defined networks," in *Proc. IEEE/IFIP 47th Annu. Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2017, pp. 249–260.
[58] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. USENIX Workshop Hot Topics Cloud Comput.*, 2010, p. 95.
[59] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *Acm Sigmod Rec.*, vol. 22, no. 2, pp. 207–216, 1993.
[60] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, 2000.
[61] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1010–1018.
[62] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 169–180, Jun. 2017.
[63] Y. Djenouri, A. Zimek, and M. Chiarandini, "Outlier detection in urban traffic flow distributions," in *Proc. Int. Conf. Data Mining*, Nov. 2018, pp. 935–940.
[64] W. Gao, S. Oh, and P. Viswanath, "Demystifying fixed k-nearest neighbor information estimators," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5629–5661, Aug. 2018.
[65] Y. Bu, S. Zou, Y. Liang, and V. V. Veeravalli, "Estimation of KL divergence: Optimal minimax rate," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2648–2674, Apr. 2018.

YOUCEF DJENOURI received the Ph.D. degree in computer engineering from the University of Science and Technology Houari Boumediene, Algeria, in 2014. From 2014 to 2015, he was a permanent Teacher-Researcher with the University of Blida, Algeria, where he is currently a member of the LRDSI Laboratory. He has received a Postdoctoral Fellowship from UNIST, South Korea, and he was involved in the BPM Project supported by UNIST, in 2016. In 2017, he was a Postdoctoral Researcher with Southern Denmark University, where he was involved in urban traffic data analysis. He was with the Norwegian University of Science and Technology, Trondheim, Norway. He has published over 25 refereed conference papers, 20 international journal articles, 2 book chapters, and 1 tutorial paper in the areas of data mining, parallel computing, and artificial intelligence. He is currently involved in topics related to artificial intelligence and data mining, with a focus on association rules mining, frequent itemsets mining, parallel computing, swarm and evolutionary algorithms, and pruning association rules. He has received a Postdoctoral Fellowship from the European Research Consortium on Informatics and Mathematics. He has participated in many international conferences worldwide, and he has received short-term research visitor internships to many renowned universities including ENSMEA, Poitiers, the University of Poitiers, and the University of Lorraine.

ASMA BELHADI received the Ph.D. degree in computer engineering from the University of Science and Technology Houari Boumediene, Algeria, in 2016. She has published over 10 refereed research articles in the areas of artificial intelligence. She has received short-term research visitor internships to many renowned universities including IRIT, Toulouse. She is currently involved in topics related to artificial intelligence and data mining, with a focus on logic programming. She has participated in many international conferences worldwide.

JERRY CHUN-WEI LIN received the Ph.D. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2010. He is currently an Associate Professor with the Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences (HVL), Bergen, Norway. He has published over 200 research papers in peer-reviewed international conferences and journals, which have received over 2200 citations. His research interests include data mining, privacy-preserving and security, Big Data analytics, and social networks. He is also the Co-Leader of the popular SPMF open-source data-mining library. He is currently the Editor-in-Chief of *Data Mining* and *Pattern Recognition*, an Associate Editor of the *Journal of Internet Technology* and the IEEE Access, and an Editorial Board Member of intelligent data analysis.

ALBERTO CANO received the B.Sc. degree in computer engineering and the B.Sc. degree in computer science from the University of Córdoba, Spain, in 2008 and 2010, respectively, and the M.Sc. degree in intelligent systems and the Ph.D. degree in computer science from the University of Granada, Spain, in 2011 and 2014, respectively. He is currently an Assistant Professor with the Department of Computer Science, Virginia Commonwealth University, USA, where he is also the Head of High-Performance Data Mining Lab. He has published over 34 articles in high-impact factor journals, 39 contributions to international conferences, two book chapters, and one book in the areas of machine learning, data mining, and parallel, distributed, and GPU computing. His research was supported by the VCU Presidential Research Quest Fund in 2018. His research interests include machine learning, data mining, general-purpose computing on graphics processing units, Apache Spark, and evolutionary computation. His research has received the Amazon AWS Machine Learning Award, in 2018. He is an Associate Editor of the IEEE Access.