



# Urban Sensing for Anomalous Event Detection: Distinguishing Between Legitimate Traffic Changes and Abnormal Traffic Variability

Masoomah Zamani<sup>1</sup>(✉), Mengyi He<sup>2</sup>, Masud Moshtaghi<sup>3</sup>, Zahra Ghafoori<sup>1</sup>,  
Christopher Leckie<sup>1</sup>, James C. Bezdek<sup>1</sup>, and Kotagiri Ramamohanarao<sup>1</sup>

<sup>1</sup> The University of Melbourne, Melbourne, Australia  
mzamemi@student.unimelb.edu.au,  
{zahra.ghafoori, caleckie, jbezdek, kotagiri}@unimelb.edu.au  
<sup>2</sup> Kepler Analytics, Melbourne, Australia  
miley.he@kepleranalytics.com.au  
<sup>3</sup> Amazon, Manhattan Beach, CA, USA  
mmasud@amazon.com

**Abstract.** Sensors deployed in different parts of a city continuously record traffic data, such as vehicle flows and pedestrian counts. We define an unexpected change in the traffic counts as an *anomalous local event*. Reliable discovery of such events is very important in real-world applications such as real-time crash detection or traffic congestion detection. One of the main challenges to detecting anomalous local events is to distinguish them from legitimate global traffic changes, which happen due to seasonal effects, weather and holidays. Existing anomaly detection techniques often raise many false alarms for these legitimate traffic changes, making such techniques less reliable. To address this issue, we introduce an unsupervised anomaly detection system that represents relationships between different locations in a city. Our method uses training data to estimate the traffic count at each sensor location given the traffic counts at the other locations. The estimation error is then used to calculate the anomaly score at any given time and location in the network. We test our method on two real traffic datasets collected in the city of Melbourne, Australia, for detecting anomalous local events. Empirical results show the greater robustness of our method to legitimate global changes in traffic count than four benchmark anomaly detection methods examined in this paper. Data related to this paper are available at: [https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/147696bb47544a209e0a5e79e165d1b0\\_0](https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/147696bb47544a209e0a5e79e165d1b0_0).

**Keywords:** Pedestrian event detection  
Vehicle traffic event detection · Anomaly detection  
Urban sensing · Smart cities

## 1 Introduction

With the advent of the *Internet of Things* (IoT), fine-grained urban information can be continuously recorded. Many cities are equipped with such sensor devices to measure traffic counts in different locations [10]. Analyzing this data can discover anomalous traffic changes that are caused by events such as accidents, protests, sports events, celebrations, disasters and road works. For example, real-time crash detection can increase survival rates by reducing emergency response time. As another example, automatic real-time traffic congestion alarms can reduce energy consumption and increase productivity by providing timely advice to drivers [15]. Anomaly detection also plays an important role in city management by reducing costs and identifying problems with critical infrastructure.

**Definition 1. *Anomalous local events:*** *Events that occur in a local area of a city and cause an unexpected change in the traffic measurements are called anomalous local events in this paper. Local events can occur in a single location or a small set of spatially close neighbor locations.*

City festivals, such as the White Night event or the Queen Victoria night market (QVM) in the *Central Business District* (CBD) of Melbourne, Australia, are examples of anomalous local events. The White Night event causes a significant decrease in the vehicle counts in some local areas in the CBD due to road closures. The QVM night market causes a significant increase in the pedestrian traffic in a market in the CBD.

**Definition 2. *Legitimate global traffic changes:*** *Global traffic changes that occur in almost all locations of the city are called legitimate global traffic changes in this paper.*

Global changes to traffic counts due to seasonal effects, weather and holidays are examples of legitimate changes, and these changes should not be considered as anomalies. Most existing anomaly detection techniques raise many false alarms for these legitimate global traffic changes, making such anomaly detection techniques unreliable for use in real-world applications. In this paper, we propose a *City Traffic Event Detection* (CTED) method that is able to detect *anomalous local events* while ignoring *legitimate global traffic changes* as anomalous.

Consider the case study of pedestrian and road vehicle traffic counts in the Melbourne CBD. Pedestrian and vehicle count data is continuously measured by sensors at different locations. For example, pedestrian counts are recorded at hourly intervals at 32 locations, while vehicle traffic counts are recorded at 15 min intervals at 105 locations. This data has been made publicly available [1,2]. Figure 1 shows the map of the locations of the pedestrian count sensors in the CBD of Melbourne. Figure 2 shows some examples of legitimate global vehicle traffic changes including two weekends and a weekday public holiday (Australia Day), and also an anomaly that occurred due to a road closure at a location in the vehicle traffic dataset in the Melbourne CBD on 16 January 2014. Our goal is to detect *when* and *where* an anomaly occurs in the pedestrian and vehicle traffic data when a relatively small amount of training data exists.

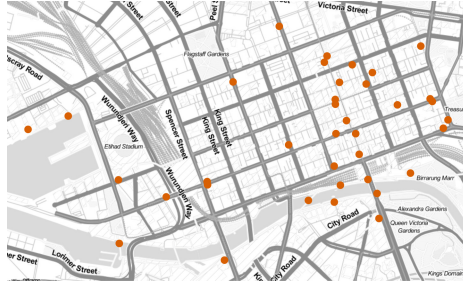


Fig. 1. Pedestrian counting locations in Melbourne, Australia.

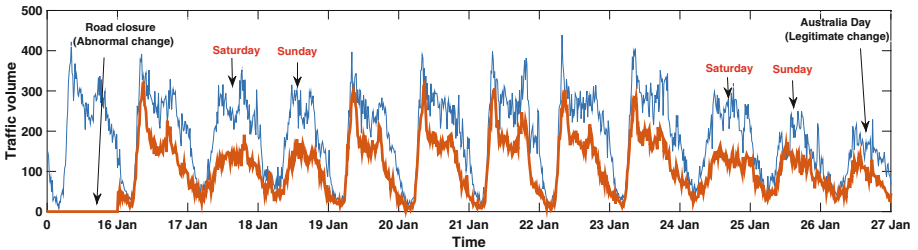


Fig. 2. Road vehicle traffic counts at two Melbourne locations (red and blue) over 12 days. (Color figure online)

Existing work [5, 6, 13, 18] has several limitations for use in real-world applications. Specifically:

- **Unsupervised:** A key challenge in detecting anomalous local events is the lack of labeled (ground truth) data. Our proposed method, CTED, is unsupervised, so it circumvents this problem. Moreover, insufficient training data limits the use of anomaly detection methods that require a large number of observations for training. For example, techniques based on *One Class Support Vector Machines* (OCSVMs) [19] and Deep Learning [7] methods are limited by this requirement. CTED is able to work with a relatively small amount of training data.
- **Detecting both spatial and temporal anomalies:** Most anomaly detection methods for data streams [7, 10, 14] can only identify the time but not the location of anomalous traffic events. In contrast, CTED can detect when and where an unexpected traffic change occurs.
- **Independence of prior data distributional knowledge:** Many existing anomaly detection methods rely on prior distributional knowledge about the data [6, 7, 18]. In contrast, CTED is based on a simple linear regression technique that avoids this requirement.

- **Robustness to legitimate global traffic changes:** Existing anomaly detection methods often misclassify legitimate global traffic changes as anomalies. CTED offers greater robustness to legitimate global traffic changes by using linear regression and modeling the relationships between traffic counts at different locations. The main contributions of our paper are as follows:
  - To the best of our knowledge, we develop CTED, which is the first unsupervised anomaly event detection method focused on legitimate global traffic changes that identifies not only the time but also the location of anomalous traffic changes in a city environment.
  - Our distribution-free approach builds relative normal models instead of absolute normal models for each location by investigating the typical relationships between traffic counts in different locations (note that we use the term “normal model” to mean “non-anomalous model” and not “Gaussian distribution”). This resolves problems caused by using absolute traffic counts such as declaring legitimate global traffic changes as anomalous.
  - We conduct our experiments on two real datasets collected in the city of Melbourne and evaluate our method on real events to verify the accuracy of the proposed method in real applications.
  - We show that our proposed method detects real anomalous local events more accurately than the comparison methods used in this paper, while being more robust to legitimate global traffic changes.

## 2 Related Work

*Anomaly detection* (AD) methods that can identify both the location and time of anomalies [4–6, 13, 17] (Temporal and Spatial AD) use two different approaches to compute an anomaly score at a specific location. The first approach, the *single-profile* approach, relies solely on the traffic counts at the location itself. The second approach, the *cluster-profile* approach, combines the traffic counts at other locations when determining the anomaly score at a particular location.

Methods in [5, 6, 16, 22–24] are single-profile anomaly detection approaches. For example, in [24], anomalies in urban traffic are detected using *Stable Principal Component Pursuit* (SPCP). The *Global Positioning System* (GPS) data from mobile phone users in Japan is used in [23] to detect anomalous events using 53 *Hidden Markov Models* (HMMs). In [5], anomalies in pedestrian flows are detected using a frequent item set mining approach, which was improved in [6] by using a window-based dynamic ensemble clustering approach.

Approaches reported in [3, 9, 13, 18] allow traffic counts at other locations to influence AD at a specified location. For example, in [9], anomalies that are detected at a specific location are partially influenced by the traffic at locations with similar behavior found using the k-means algorithm. Rajasegarar et al. [18] detect anomalies in resource-constrained Wireless Sensor Networks (WSNs) using multiple hyperellipsoidal clusters and calculating the relative remoteness between neighbors. The method in [3] analyzes WiFi access point utilization

patterns on a university campus to detect special events in physical spaces. This method has many false alarms at the beginning of each working day.

The technique that is most similar to CTED is [13], where a framework for temporal outlier detection in vehicle traffic networks, called *Temporal Outlier Discovery* (TOD), is proposed. TOD is based on updating historical similarity values using a reward/punishment rule.

None of the methods reviewed is robust to *legitimate global traffic changes*, since these methods either do not consider other sensors while calculating the anomaly score at a sensor, or only consider spatially close neighbor sensors to a specific sensor in calculating its anomaly score. This study introduces a method for detecting *anomalous local events* in traffic count data that is highly robust to *legitimate global traffic changes*. Table 1 compares properties of several methods that have been used for AD in sensor networks to those of CTED.

**Table 1.** Comparison of related methods for anomaly detection in sensor networks

| Category        | Reference | Technique                   | Property   |             |              |                      |                          |                   |                          |
|-----------------|-----------|-----------------------------|------------|-------------|--------------|----------------------|--------------------------|-------------------|--------------------------|
|                 |           |                             | Spatial AD | Temporal AD | Unsupervised | Relative count based | Robust to global changes | Distribution-free | #parameters <sup>7</sup> |
| Single-profile  | [7]       | Cluster-based               | ×          | ✓           | ✓            | ×                    | ×                        | ×                 | 6                        |
|                 | [10]      | Classification-based        | ×          | ✓           | ×            | ×                    | ×                        | ✓                 | 3                        |
|                 | [22]      | Statistic-based             | ✓          | ✓           | ✓            | ×                    | ×                        | ✓                 | 2                        |
|                 | [16]      | Statistic-based             | ✓          | ✓           | ✓            | ×                    | ×                        | ×                 | 5                        |
|                 | [24]      | Classification-based        | ✓          | ✓           | ✓            | ×                    | ×                        | ✓                 | 3                        |
|                 | [23]      | HMM-based                   | ×          | ✓           | ×            | ×                    | ×                        | ✓                 | 4                        |
|                 | [5]       | Frequent item set-based     | ✓          | ✓           | ✓            | ×                    | ×                        | ✓                 | 3                        |
|                 | [6]       | Window-based                | ✓          | ✓           | ✓            | ×                    | ×                        | ×                 | 5                        |
| Cluster-profile | [9]       | Cluster-based               | ✓          | ✓           | ×            | ×                    | ×                        | ✓                 | 2                        |
|                 | [18]      | Cluster-based               | ✓          | ✓           | ✓            | ×                    | ×                        | ×                 | 4                        |
|                 | [3]       | Window-based                | ✓          | ✓           | ✓            | ✓                    | ×                        | ✓                 | 2                        |
|                 | [13]      | Historical similarity-based | ✓          | ✓           | ✓            | ×                    | ×                        | ✓                 | 5                        |
|                 | CTED      | Linear regression-based     | ✓          | ✓           | ✓            | ✓                    | ✓                        | ✓                 | 1                        |

<sup>7</sup> Number of input parameters required to implement the anomaly detection method.

### 3 Problem Statement

Suppose there are  $m$  different locations in a city,  $L = \{1, \dots, m\}$ . We consider the traffic counts in an hourly basis,  $H = \{1, \dots, 24\}$ . Assume  $n_i^{(h)}(q)$  is the traffic count for location  $i$  at hour  $h \in H$  of day  $q$ , denoted by  $(i, h, q)$ , and

let  $TD = \{n_i^{(h)}(q) : 1 \leq i \leq m, h \in H, 1 \leq q \leq N\}$  be the training traffic data collected during  $N$  days. Our goal is to detect the location and time of *unexpected traffic counts* (Definition 3) for  $q > N$ , which we regard as *anomalous local events* (Definition 1). The proposed method should distinguish the anomalous local events from *legitimate global traffic changes* (Definition 2). We assume that the majority of the data in  $TD$  corresponds to normal traffic.

**Definition 3. Unexpected traffic count:** An Unexpected traffic count occurs at  $(i, h, q)$  if its observed traffic count,  $n_i^{(h)}(q)$ , is significantly different from its expected traffic count,  $\hat{n}_{ij}^{(h)}(q)$ .

### 4 Proposed Method - CTED

#### 4.1 Overview of the Method

The basic research questions that need to be addressed are: (a) How can we find the expected traffic count at location  $i$ ? (b) Do the traffic counts at other locations affect the traffic count at location  $i$ ? (c) How to reduce the false alarm rate of anomaly detection on legitimate global traffic changes?

To address these research problems, CTED consists of two phases: an *offline phase*, which builds a series of models to estimate the normal traffic for each location  $i$  at hour  $h$  of the day, and weights the normal models, and an *online phase*, which uses the accumulated weighted error of the normal models for each location  $i$  at hour  $h$  and the current traffic measurements at the other locations to compute the anomaly score at  $(i, h, q)$  for  $q > N$ .

Figure 3 presents the main steps of our system. Next, we explain each of these phases in detail.

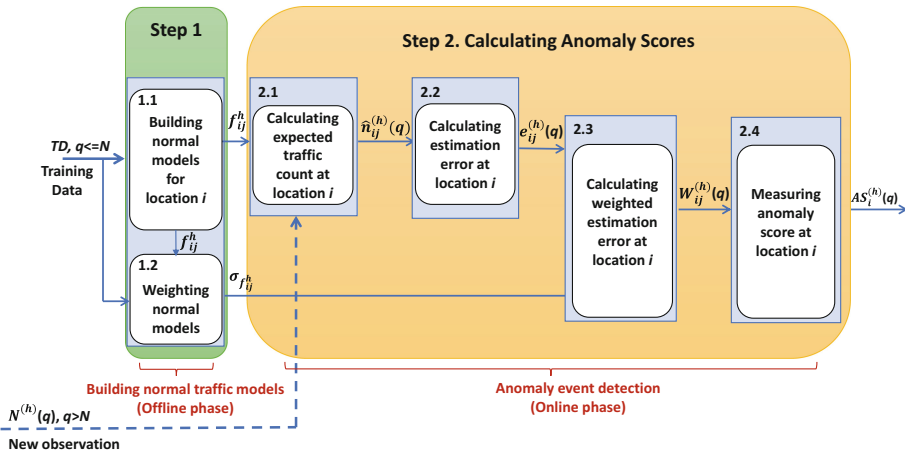
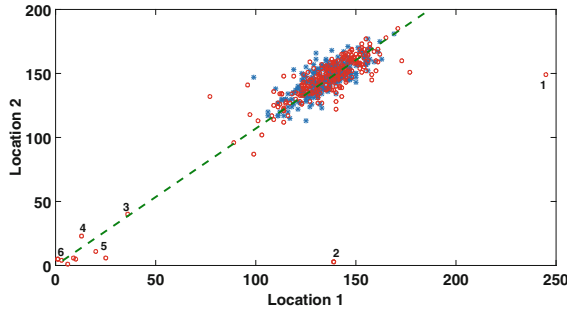


Fig. 3. Main steps of CTED for detecting anomalous events at location  $i$ .

## 4.2 Step 1: Build and Weight Normal Models

In a legitimate global traffic change, traffic counts change with almost the same ratio in all locations. This fact encouraged us to make our event detection model insensitive to this ratio change. To this end, we investigated the relative traffic counts between different locations. For example, we found that the traffic counts at Elizabeth Street in the Melbourne CBD are usually two times higher than the traffic at Spring Street. So, in a legitimate global traffic change, when the traffic at the Elizabeth Street increases/decreases by 1.5 times, we expect the traffic at the Spring Street to increase/decrease by almost 1.5 times. This suggested that we deploy a linear regression model.



**Fig. 4.** Linear regression and legitimate global traffic changes for the vehicle count dataset. Linear regression considers observations affected by legitimate global traffic changes (observation numbers 3 to 6) as normal observations for the vehicle count dataset. (Color figure online)

Figure 4 shows an example where the traffic counts change almost linearly in a legitimate global traffic change (weekday holidays in this example represented by numbers 3 to 6) in two linearly correlated locations. The green linear regression line models the normal behavior between two locations at 1 pm for the vehicle traffic dataset. The blue star points (\*) are training data observations and the red circle points (O) are upcoming test observations. This figure shows how using a simple linear regression for modeling the relative normal behaviours increases the robustness of CTED to legitimate global traffic changes (see observation numbers 3 to 6). This figure also shows that a linear regression models the normal behavior of data better than other methods such as clustering. Clustering techniques detect one normal cluster for the training observations and do not generalize well to the legitimate traffic changes (observation numbers 3 to 6 in the bottom left of the figure, which are affected by holidays in this example).

Let functions  $f_{ij}^{(h)}$ ,  $j \neq i$  represent our normal models that estimate the traffic counts at location  $i$  at hour  $h$  given the traffic counts at location  $j$  at the same hour  $h$ . We learn linear regression models of the form  $f_{ij}^{(h)}(x) = a_{ij}^{(h)}x + b_{ij}^{(h)}$  for the traffic data, where  $a_{ij}^{(h)}$  and  $b_{ij}^{(h)}$  are the coefficients of the model learnt using the training data  $\{n_i^{(h)}(q), n_j^{(h)}(q)\}$ ,  $q = 1 \dots N$ , where  $n_k^{(h)}(q)$  is the traffic count

at location  $k$  at hour  $h$  of day  $q$ . We train a model  $f_{ij}^{(h)}$  for each hour  $h$  of the day using the training data  $TD$ . Each trained model is then used to evaluate new observations for the same hour  $h$  of the day.

Although the observed traffic counts at other locations are used to estimate the traffic counts at location  $i$  at hour  $h$ , the locations that have the highest linear correlation with location  $i$  are more important. Therefore, we assign higher weights to the locations that have the largest linear correlation with location  $i$  (step 1.2). To this end, we weight  $f_{ij}^{(h)}$  models by their standard errors,  $\sigma_{f_{ij}^{(h)}}$  using the training data (Eq. 1).

$$\sigma_{f_{ij}^{(h)}} = \left[ \frac{\sum_{q=1}^N \left( n_i^{(h)}(q) - f_{ij}^{(h)}(n_j^{(h)}(q)) \right)^2}{N} \right]^{\frac{1}{2}} \tag{1}$$

where  $f_{ij}^{(h)}(n_j^{(h)}(q))$  is the estimated traffic count at  $(i, h, q)$  using the observed traffic count at location  $j$  at hour  $h$  of day  $q$ ,  $n_j^{(h)}(q)$ .

### 4.3 Step 2: Calculate Anomaly Scores

In this step, we use the trained models  $f_{ij}^{(h)}$  to evaluate the observations  $(i, h, q)$ ,  $q > N$ . The expected traffic count at  $(i, h, q > N)$ ,  $\hat{n}_{ij}^{(h)}(q)$ , based on the current traffic counts at location  $j$ ,  $n_j^{(h)}(q)$ , is calculated in Eq. 2 (see also step 2.1 in Fig. 3).

$$\hat{n}_{ij}^{(h)}(q) = f_{ij}^{(h)}(n_j^{(h)}(q)), \quad q > N \tag{2}$$

For each upcoming observation at location  $i$ , the absolute estimation error based on the traffic counts at  $j \in CK_i^{(h)}$ , for  $(h, q)$  is calculated in Eq. 3.

$$e_{ij}^{(h)}(q) = \left| n_i^{(h)}(q) - \hat{n}_{ij}^{(h)}(q) \right| \tag{3}$$

To give more importance to the locations that have high linear correlation with location  $i$ , we weight the estimation error of the traffic count at  $(i, h)$  as shown in Eq. 4 using  $\sigma_{f_{ij}^{(h)}}$ .

$$W_{ij}^{(h)}(q) = \frac{e_{ij}^{(h)}(q)}{\sigma_{f_{ij}^{(h)}}} \tag{4}$$

where  $\sigma_{f_{ij}^{(h)}}$  is the standard error of the trained normal model  $f_{ij}^{(h)}$  from Eq. 1.

**Measuring Anomaly Scores.** The anomaly score at  $(i, h, q)$  is calculated in Eq. 5.

$$AS_i^{(h)}(q) = \sum_{j \in CK_i^{(h)}, j \neq i} W_{ij}^{(h)}(q) \tag{5}$$



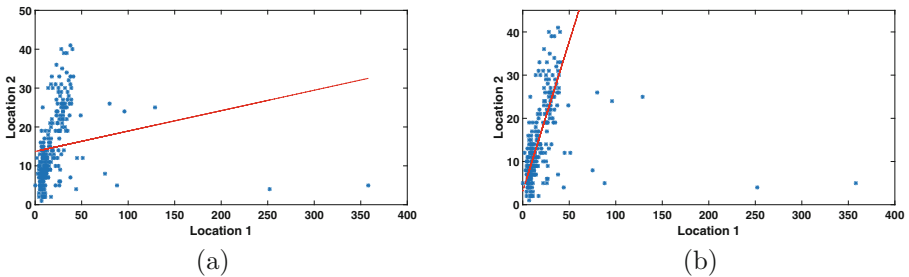
$AS_i^{(h)}(q)$  is the sum of the weighted estimation errors of the traffic counts at  $(i, h, q)$ . An anomalous traffic event is declared at  $(i, h, q)$  if its anomaly score,  $AS_i^{(h)}(q)$ , exceeds a pre-specified threshold,  $thr_{CTED}$ . We discuss the selection of this threshold in Sect. 5.

*Why do we ignore lag effects?* In building normal models and calculating anomaly scores, we do not consider lag effects. Usually, vehicles move in “waves”, i.e., a high traffic count at location  $i$  at time  $h$  is expected to correspond to a high traffic count at a spatially close neighbor correlated location  $j$  at the next time  $h + 1$ . However, we do not consider lag effects because of the properties of our datasets, i.e., the low sampling rate (15-min for the vehicle traffic data and one-hour for the pedestrian data) and the small distances between locations (many of the locations are just one block apart). We considered lag effects in our studies, and noticed a small reduction in the accuracy of our anomaly event detection method. When distances between sensors and/or sampling times increase, accounting for lag becomes more effective.

#### 4.4 DBSCAN for Removing Outliers from the Training Data

We examined the linear correlation between different locations using the *Pearson Correlation Coefficient* (PCC) and we found that in some pairs of locations, the linear regressions are affected by outliers in the training data, resulting in a low PCC (see Fig. 5(a)).

To prevent the effect of outliers, we use the *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) method [8] to remove outliers before building linear regression models. We chose DBSCAN as it is a high performance unsupervised outlier detection method used in many recent research papers [12, 20, 21]. Figure 5(b) shows the linear regression model after using DBSCAN for removing outliers from the training data, which confirms that the linear regression model after removing outliers is a more reliable fit.



**Fig. 5.** Linear regression at two locations. (a) Before removing outliers, (b) After removing outliers using DBSCAN.

For the pedestrian data, removing outliers reduced the proportion of locations with low PCC values (lower than 0.6) from 6% to 0.6%. For the vehicle traffic data, this value reduced from 15% to 4%. When we used DBSCAN, we assumed that 20% of the training data are outliers. We changed this assumed outlier percentage in the range from 5 to 30 and there was little variation in the results. The effect of outliers on the linear correlations in the training vehicle traffic data is less than their effect on the coefficients in pedestrian data because the time resolution of the training data is lower in the vehicle traffic data than for the pedestrian traffic data. The smaller the sample size, the greater the effect of outliers on the normal models.

## 5 Experiments

### 5.1 Datasets and Ground Truth

We use the real datasets of the pedestrian count data [2] and the vehicle traffic count data [1] as described in Sect. 1. We ignore missing values in the training data. In the vehicle traffic data, only 3% of the training data is missing.

The performance of our method is compared to the benchmark algorithms based on some known real anomalous local events and real legitimate global traffic changes (see Table 2). For anomalous local events, we choose winter fireworks in Waterfront City (Docklands) and night market festivals in the *Queen Victoria Market* (QVM) in 2015 for the pedestrian data where an uncharacteristic pedestrian flow is experienced. We also consider the White Night event that happened in some parts of the city in 2014 as an anomalous local event for the vehicle traffic data as only some road segments in the CBD are blocked or partially affected by this event. Normal traffic patterns in April are the normal ground truth for both datasets. We also consider weekday holidays in Melbourne such as New Year’s Eve, Good Friday and the Queen’s Birthday as legitimate global traffic changes for both datasets (see Table 2).

**Table 2.** Normal and anomaly ground truth

| Dataset    | Scenario | $GT_{Normal}$   | $GT_{Anomaly}$ (anomalous local events)      |
|------------|----------|---|--|
| Pedestrian | 1        | Normal traffic patterns in April                      | QVM night market, Docklands winter fireworks |
|            | 2        | Weekday holidays (Legitimate global traffic changes)  | QVM night market, Docklands winter fireworks |
|            | 3        | Normal traffic patterns in April and weekday holidays | QVM night market, Docklands winter fireworks |
| Vehicle    | 4        | Normal traffic patterns in April                      | White night                                  |
|            | 5        | Weekday holidays (Legitimate global traffic changes)  | White night                                  |
|            | 6        | Normal traffic patterns in April and weekday holidays | White night                                  |

## 5.2 Comparison Methods

We compare our method with four other methods: OCSVM [11], TOD [13], Boxplot [22], and  $k$ -sigma rule [16]. OCSVM models a training set comprising only the normal data as a single class. Dense subsets of the input space are labelled as normal whereas observations from other subsets of the input space are labelled as anomalies. We train an OCSVM model with the *Radial Basis Function* (RBF) in an unsupervised manner given the technique proposed in [11].

TOD proposed in [13] for detecting anomalies in vehicle traffic data, and is similar to CTED. The main differences are that TOD considers all locations to have the *same importance* (*weight*) when determining the anomaly score of each location and uses absolute traffic counts. TOD uses a reinforcement technique and expects two historically similar locations to remain similar and two historically dissimilar locations to stay dissimilar.

The last two comparison methods are the extended versions of the standard Boxplot [22] and the 3-sigma [16] anomaly detection methods. Boxplot constructs a box whose length is the *Inter Quartile Range* (IQR). Observations that lie outside  $1.5 * \text{IQR}$  are defined as anomalies. In this paper, we learn IQR for each hour  $h$  using the training data, and then we consider different ratios of IQR as the threshold for anomalies. We define observations that are outside  $k_{thr_B} * \text{IQR}$  as anomalies and investigate the overall performance of Boxplot for different threshold values,  $k_{thr_B}$ . The 3-sigma rule calculates the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the training data and then declares the current observation ( $n_i^{(h)}(q)$ ) anomalous if  $\left|n_i^{(h)}(q) - \mu\right| > 3\sigma$ . We learn  $\mu$  and  $\sigma$  for each hour  $h$  of the day using the training data. We then use a general version,  $k$ -sigma rule, where observations that are outside  $\left|n_i^{(h)}(q) - \mu\right| > k_{thr_S}\sigma$  are defined as the anomalies.

## 5.3 Experimental Setup

We evaluate the performance of CTED by computing the *Area Under the ROC Curve* (AUC) metric. Table 2 shows the ground truth (GT) that we use for both real datasets under different experimental scenarios discussed in Sect. 5.1. In calculating the AUC metric, true positives are anomalous local events, (*location, hour, day*) triples, belonging to  $GT_{Anomaly}$  that are declared anomalous, and false positives are normal traffic changes (including legitimate global ones), (*location, hour, day*) triples, that belong to  $GT_{Normal}$  but are misdetected as anomalous events.

**Setting Parameters.** The threshold that is compared to anomaly score values is the parameter that defines anomalies in all the comparison methods and the proposed method:  $thr_{AS}$  in CTED,  $thr_{SVM}$  in OCSVM,  $k_{thr_B}$  in Boxplot and  $k_{thr_S}$  in  $k$ -sigma. This parameter is the only parameter required by CTED.

In OCSVM, to extract features for training the models,  $n_c$  correlated sensors are identified for each sensor. The features of OCSVM are the ratio of the counts

in two correlated sensors. On our experiments, we change  $n_c \in \{5, 10, 15, 20\}$  for the number of correlated locations to each location and we report the best results. Note that by increasing  $n_c$  from 20, we observed reduction in the accuracy. We train a model for each hour of the day using the training data. These models are used for evaluating the current observations.

In TOD, in addition to the anomaly score threshold, a similarity threshold and three other parameters,  $\alpha_1 < 1$ ,  $\alpha_2 \geq 0$  and  $\beta > 1$ , must also be determined. Setting appropriate values for these parameters is difficult and is best done using prior knowledge about the dataset. In our experiments, we changed  $\alpha_1$  in the range  $[0-1)$ ,  $\alpha_2$  in the range  $[0-10]$  and  $\beta$  in the range  $(1-10]$  for the test data. We found that the values of 0.7, 2 and 1 respectively for  $\alpha_1$ ,  $\alpha_2$  and  $\beta$  lead to the highest AUC value for the test data in the pedestrian traffic data and the values of 0.99, 0 and 1.1 respectively for  $\alpha_1$ ,  $\alpha_2$  and  $\beta$  lead to the highest AUC value for the test data in the vehicle traffic data. We used these parameters for TOD in our experiments. In practice, the necessity to tune the TOD parameters using test data makes this approach difficult and time consuming.

In the Boxplot and  $k$ -sigma rule methods, we estimate IQR, the observation mean  $\mu$  and the observation standard error of the estimate  $\sigma$  for each hour  $h$  of the day using training data.

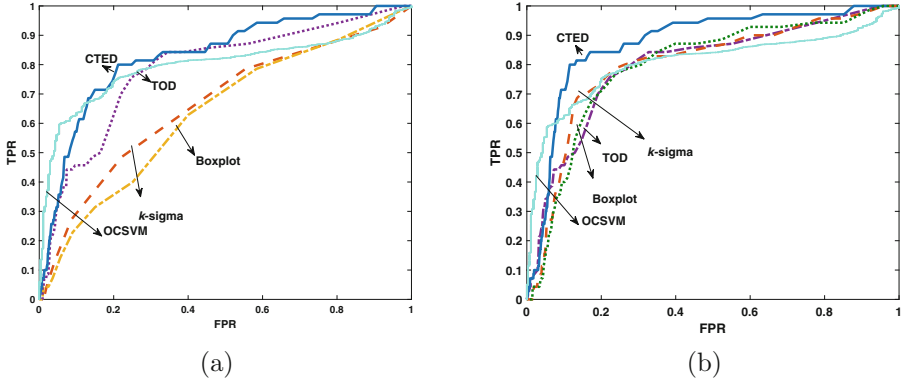
**Table 3.** AUC values for CTED and the benchmarks

| Dataset    | Scenario | OCSVM<br>[11] | TOD<br>[13] | Boxplot<br>[22] | $k$ -sigma<br>[16] | CTED Proposed<br>method |
|------------|----------|---------------|-------------|-----------------|--------------------|-------------------------|
| Pedestrian | 1        | 0.86          | 0.71        | 0.85            | 0.84               | <b>0.86</b>             |
|            | 2        | 0.78          | 0.71        | 0.78            | 0.71               | <b>0.84</b>             |
|            | 3        | 0.84          | 0.71        | 0.83            | 0.82               | <b>0.86</b>             |
| Vehicle    | 4        | 0.82          | 0.8         | 0.79            | 0.79               | <b>0.87</b>             |
|            | 5        | 0.77          | 0.79        | 0.64            | 0.65               | <b>0.88</b>             |
|            | 6        | 0.82          | 0.79        | 0.77            | 0.78               | <b>0.88</b>             |

### 5.4 Results

Figure 6 and Table 3 compare the resulting ROC curves and AUC values for the comparison methods against CTED produced by changing the threshold of anomaly scores for three different experimental scenarios in Table 2 as discussed in Sect. 5.1. In OCSVM, we set  $n_c \in \{5, 10, 15, 20\}$  for the number of correlated locations to each location and we found that OCSVM is sensitive to the choice of this parameter. Specifically, increasing  $n_c$  from 20 resulted in a large reduction in the accuracy of OCSVM. We changed the number of correlated locations to each location for CTED and we found that CTED has a low sensitivity to  $n_c$ . In OCSVM, best results for the pedestrian dataset was achieved at  $n_c = 15$ , while we got the best results for the vehicle dataset when we set  $n_c = 10$ .

The bolded results in Table 3 show that the AUC values of CTED are higher than all the benchmark approaches for all the above-mentioned scenarios in both the pedestrian counts and the vehicle traffic datasets. In Fig. 6, we plot the *Receiver Operating Characteristic* (ROC) curve for the vehicle traffic dataset against the three benchmarks. This figure confirms that CTED performs better than other benchmarks for all the scenarios.



**Fig. 6.** ROC curves for vehicle traffic data for the real anomalous events in Table 2. (a) Legitimate global traffic changes are the normal ground truth, (b) Normal traffic patterns in April are the normal ground truth.

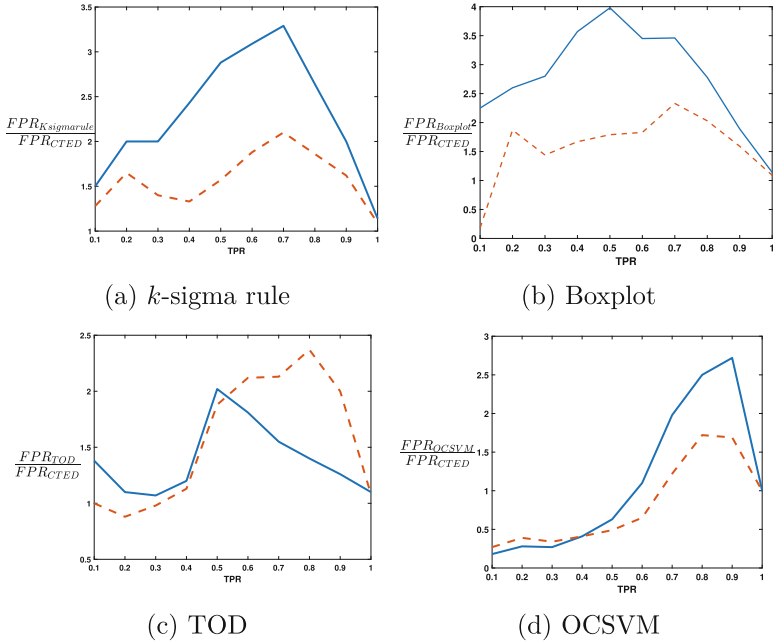
Comparing Fig. 6(a) and (b) reveals that the difference between the performance of CTED and the other benchmarks is mostly larger when legitimate global traffic changes are the normal ground truth. This larger difference stems from the lower false positive rate of CTED because it is more robust to legitimate global traffic changes compared to the benchmark techniques.

Reliable anomaly detection is difficult in practice. Reducing false positives is very important as this makes the anomaly detection system more reliable for city management purposes. A system that is not robust to legitimate global traffic changes generates many false alarms. This makes existing anomaly event detection methods unreliable for use in real applications, such as vehicle accident detection and traffic congestion detection systems.

### 5.5 Robustness to Legitimate Global Traffic Changes

Figure 7 compares the ratio of the false positive rate of CTED to the other three benchmarks for different values of *True Positive Rate* (TPR). Figure 7 shows that the *False Positive Ratio* (FPR) ratio between CTED and the benchmarks for the legitimate global traffic changes is lower than the local events, which confirms the greater robustness of CTED compared to the benchmarks for legitimate global traffic changes.

Figure 7(a) and (b) highlight that the  $k$ -sigma and Boxplot methods produce much higher false positives for legitimate global traffic changes than local events for vehicle traffic data. However, Fig. 7(c) and (d) show that TOD and OCSVM are more robust than the  $k$ -sigma and Boxplot methods to legitimate global traffic changes but still less robust than CTED. The greater robustness of TOD and OCSVM to legitimate global traffic changes is mainly due to considering traffic counts at other locations (relative traffic counts) when computing the anomaly score at each location.



**Fig. 7.** The ratio of FPR produced by comparison methods compared to CTED for the same values of TPR for the vehicle traffic dataset. The anomaly ground truth is the the White Night event in all the cases. The dashed red lines show the results when the normal traffic patterns in April are considered as the normal ground truth while the blue lines show the results when legitimate global traffic changes are considered as the normal ground truth. (Color figure online)

### 5.6 Time Complexity

CTED is composed of an offline and an online phase. The complexity of the offline phase in the worst case is  $O(m^2 n_{TD}^2)$ , where  $m$  is the number of locations, and  $n_{TD}$  is the number of observation vectors in the training data. The complexity of DBSCAN in the worst case is  $O(m^2 n_{TD}^2)$ , and the time complexity for building normal linear regression models and weighting them is  $O(m^2 n_{TD})$ . The offline phase only executed once. Note that  $n_{TD} \gg n$ , as we discussed in Sect. 4.4.

The online phase is executed whenever a new observation vector arrives. The time complexity for processing each new observation is  $O(m^2)$  as we find the estimation error for the current observation in each location based on the other locations.

## 6 Conclusions

In this paper, we proposed a new unsupervised method for detecting anomalous local traffic events, called CTED. This method that is highly *robust* to legitimate global traffic changes. This method builds normal models for each location by investigating the linear relationships between different locations in the city and uses the models to detect anomalous local events. Our experiments on two real traffic datasets collected in the Melbourne CBD, the pedestrian count and the vehicle traffic count datasets, verify that our simple linear regression-based method accurately detects anomalous real local events while reducing the false positive rate on legitimate global traffic changes compared to four other benchmark methods for anomaly detection in traffic data.

Changes in the city infrastructure can change the normal behaviour of the traffic in several locations of a city. As a future direction of our research, we aim to exploit time series change point detection methods to find the time of these behavioural changes in the traffic, and automatically update CTED when it is necessary.

## References

1. Viceroads open traffic volume data. [https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/147696bb47544a209e0a5e79e165d1b0\\_0](https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/147696bb47544a209e0a5e79e165d1b0_0) (2014)
2. City of Melbourne pedestrian counting system. <http://www.pedestrian.melbourne.vic.gov.au/> (2015)
3. Baras, K., Moreira, A.: Anomaly detection in university campus WiFi zones. In: 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM), pp. 202–207 (2010)
4. Dani, M.-C., Jollois, F.-X., Nadif, M., Freixo, C.: Adaptive threshold for anomaly detection using time series segmentation. In: International Conference on Neural Information Processing, pp. 82–89 (2015)
5. Doan, M.T., Rajasegarar, S., Leckie, C.: Profiling pedestrian activity patterns in a dynamic urban environment. In: 4th International Workshop on Urban Computing (UrbComp) (2015)
6. Doan, M.T., Rajasegarar, S., Salehi, M., Moshtaghi, M., Leckie, C.: Profiling pedestrian distribution and anomaly detection in a dynamic environment. In: CIKM, pp. 1827–1830 (2015)
7. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recogn.* **58**, 121–134 (2016)
8. 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: KDD vol. 34, pp. 226–231 (1996)

9. Frias-Martinez, V., Stolfo, S.J., Keromytis, A.D.: Behavior-profile clustering for false alert reduction in anomaly detection sensors. In: Annual Computer Security Applications Conference (ACSAC), pp. 367–376 (2008)
10. Garcia-Font, V., Garrigues, C., Rifà-Pous, H.: A comparative study of anomaly detection techniques for smart city wireless sensor networks. *Sensors* **16**(6), 868 (2016)
11. Ghafoori, Z., Erfani, S.M., Rajasegarar, S., Bezdek, J.C., Karunasekera, S., Leckie, C.: Efficient unsupervised parameter estimation for one-class support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* (2018)
12. Jeong, S.Y., Koh, Y.S., Dobbie, G.: Phishing detection on Twitter streams. In: Cao, H., Li, J., Wang, R. (eds.) PAKDD 2016. LNCS (LNAI), vol. 9794, pp. 141–153. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-42996-0\\_12](https://doi.org/10.1007/978-3-319-42996-0_12)
13. Li, X., Li, Z., Han, J., Lee, J.-G.: Temporal outlier detection in vehicle traffic data. In: IEEE 25th International Conference on Data Engineering (ICDE), pp. 1319–1322 (2009)
14. Limthong, K.: Real-time computer network anomaly detection using machine learning techniques. *J. Adv. Comput. Netw.* **1**(1), 1–5 (2013)
15. Nidhal, A. Ngah, U.K., Ismail, W.: Real time traffic congestion detection system. In: 5th International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1–5 (2014)
16. Pukelsheim, F.: The three sigma rule. *Am. Stat.* **48**(2), 88–91 (1994)
17. Rajasegarar, S., Bezdek, J.C., Moshtaghi, M., Leckie, C., Havens, T.C., Palaniswami, M.: Measures for clustering and anomaly detection in sets of higher dimensional ellipsoids. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2012)
18. Rajasegarar, S., et al.: Ellipsoidal neighbourhood outlier factor for distributed anomaly detection in resource constrained networks. *Pattern Recogn.* **47**(9), 2867–2879 (2014)
19. Reddy, R.R., Ramadevi, Y., Sunitha, K.: Enhanced anomaly detection using ensemble support vector machine. In: ICBDAC, pp. 107–111 (2017)
20. Shi, Y., Deng, M., Yang, X., Gong, J.: Detecting anomalies in spatio-temporal flow data by constructing dynamic neighbourhoods. *Comput. Environ. Urban Syst.* **67**, 80–96 (2018)
21. Tu, J., Duan, Y.: Detecting congestion and detour of taxi trip via GPS data. In: IEEE Second International Conference on Data Science in Cyberspace (DSC), pp. 615–618 (2017)
22. Tukey, J.W.: *Exploratory Data Analysis* (1977)
23. Witayangkurn, A., Horanont, T., Sekimoto, Y., Shibasaki, R.: Anomalous event detection on large-scale GPS data from mobile phones using Hidden Markov Model and cloud platform. In: Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, pp. 1219–1228 (2013)
24. Zhou, Z., Meerkamp, P., Volinsky, C.: Quantifying urban traffic anomalies. arXiv preprint [arXiv:1610.00579](https://arxiv.org/abs/1610.00579) (2016)