Adaptable Anomaly Detection in Traffic Flow Time Series

Md Rakibul Alam*, Ilias Gerostathopoulos*, Sasan Amini†, Christian Prehofer* and Alessandro Attanasi§
* Department of Informatics, Technical University Munich, Munich, Germany
Email: {rakibulmd.alam, ilias.gerostathopoulos,christian.prehofer}@tum.de
† Department of Civil, Geo and Environmental Engineering, Technical University Munich, Munich, Germany
Email: sasan.amini@tum.de
§ PTV SISTeMA, Rome, Italy
Email: alessandro.attanasi@ptvgroup.com

***Abstract*—Analysis of traffic data is an essential component of many intelligent transportation system applications where the quality of data plays an important role. Traffic data collected through sensors (e.g. loop detectors) often contain anomalies due to different reasons such as malfunctioning detectors or anomalous traffic conditions. Regardless of their rooting cause, such data heavily affect the results of the subsequent analysis (e.g. traffic prediction). There are a few barriers that make the anomaly detection troublesome including absence of universal definition of anomaly, change of traffic pattern over time, unavailability of labeled data, use-case driven analysis. In this paper, a new anomaly detection method for traffic univariate time-series is proposed which does not assume labeled historical data yet uses expert feedback to deal with the fluid definition of anomaly. The method is exemplified and evaluated by applying it on real traffic time series data collected through loop detectors installed in an urban road network in Europe. Employing the proposed method as a pre-process of traffic state estimation can increase the accuracy measure as well as ease the learning of different traffic patterns.**

***Index Terms*—anomaly detection, traffic flow time-series, loop detectors, clustering**

I. Introduction

Traffic measurements on different segments of a road network are essential for a wide range of applications in intelligent transportation system (ITS) such as traffic management and advanced traveler information systems (ATIS). Traffic data are usually obtained from a wide variety of sensors such as inductive loop detectors, surveillance cameras, vehicle number plate re-identification. However, such data are often noisy, erroneous, incomplete and contain outliers. Therefore, it is essential to perform data cleansing (or data cleaning) before analysing the data.

One of main goals of data cleansing is to identify anomalies in the data. According to [1] anomalies are defined as observations that are inconsistent with the rest of the data. Generally, anomalies can be categorized in two main groups. First, outliers in the data that are due to malfunctioning sensors, and second, unexpected measurements that arise because of non-recurrent traffic congestion on the road e.g. incidents or adverse weather. Detecting anomalous traffic condition has been well studied specially for intercity freeways where many automatic incident detection (AID) algorithms have been developed. Deniz et al. provide an overview of the most frequently used AID algorithms [2].

Since labeling large-scale data is tedious task, labeled anomalous data sets are rare [3]. Moreover, often subtle difference between noise and anomaly in the data makes anomaly detection very challenging. A comprehensive list of challenges are presented in [4]. Thus, anomaly detection is usually conducted considering only the internal characteristics of the data. The main challenge is to define the *normal* boundary for a certain type of data. As there is no universal solution, usually subjective judgments from domain experts are required to set the boundaries for each detection point.

In response to this, this paper describes a methods for quick and effortless annotation of anomalies in a large body of traffic time series data. The method incorporates subjective judgment in identifying appropriate thresholds that can be used for labeling points as anomalies, but can also be employed without such subjective inputs. The *normal* thresholds are identified for only a small number of time series and can be re-used in other time series (and corresponding traffic sensors) that have similar characteristics.

To evaluate the method, it is applied for identifying anomalies in time series of 15-minutes aggregated vehicle counts from the loop detectors of the city of Vienna, Austria.

The remainder of the paper is structured as follows. In section II background together with an overview of the existing literature is provided. Section III provides a detailed explanation of the proposed method together with a short description of the data used in this paper. Section IV evaluates the accuracy of our anomaly detection method and the feasibility of reusing knowledge for anomaly detection across sensors. Finally, Section V discusses important implications and Section VI concludes.

II. Background & Related Work

Conceptually anomalies should be rare occurrences. Anomaly (outlier or deviation) detection is primarily referred to a process of data mining aiming at identifying data points which do not conform to a notion of normal behavior. Noise

Current affiliaton: DENSO Automotive Germany, c.prehofer@denso-auto.de

and novelty are two concepts which are closely related with anomaly. Noise refers to values in data which are of no interest to analyst and novelty refers to patterns that were never seen before [4]. Noise removal [5], noise accommodation [6], novelty detection [7]—these are all related to anomaly detection since the process that is used for one can be also used for the other as well.

Perhaps the simplest way to detect outliers is to exclude extreme implausible values such as negative flow values or extreme positive values. Even though such plausibility checks are essential to exclude outliers, but they are not capable of detecting all anomalies. The earliest approaches of anomaly detection were based on statistical methods [8].

The most common statistical approach in the existing literature is to use box and whisker approach proposed by [9] in which all the values that extend beyond 1.5 times the interquartile are considered as outliers. Kieu et al. [10] use a similar method where they use the median absolute deviation technique to remove outliers in travel time. In this method a lower and an upper bound value are defined as: median $\pm\sigma\times$ f. Here f is a factor that defines the scatter in the data and has to be selected based on the subjective judgments.

Different statistical tests such as Grubbs test [11] and Extreme Studentized Deviation (ESD) test [12] are also used for anomaly detection. Grubb's test is used to detect anomalies in a univariate data set under the assumption of Gaussian distribution that determines the largest anomaly in a given data set. ESD is a generalized version of Grubb's to detect multiple anomalies in the given data set. Twitter developed a framework [13] recently based on ESD with ngong research to further improve the algorithm [14].

Detecting anomalies using time-series analysis is another well-studied approach in the literature. Handling outliers in a time series includes use of bayesian method [15], consideration of outliers as contamination generated from a given probability distribution [16], use of two parametric models to study outliers [17] among others. Different iterative procedures have been used in [18], [19] and [20] which are based on the two parametric models technique. Another approach was using sequential detection methods assuming probabilistic models for handling outliers [21], [22]. Using past events to detect events in the present time was used in anomaly detection of sequential data in [23]. Here, if the actual event does not match the predicted event based on the past events then actual event is marked as rare.

Scoring technique for anomalies is frequently used in regression based methods where magnitude of residual errors are used as scores. Residuals refer the part of the instance which is not explained by the regression model. Scores are used with certain confidence intervals [24] [16] [25]. Another technique to mark anomalies is analysis of the Akaike Information Content (AIC) during model fitting [26].

Using robust regression techniques was proposed in [6] with the argument that the robust regression techniques not only hide the anomalies, but can also detect the anomalies, because the anomalies tend to have larger residuals from the robust fit. Similar robust anomaly detection approach has been applied in Autoregressive Integrated Moving Average (ARIMA) models [27] [28]. An empirical approach to anomaly detection in time-series is presented most recently in [29], which is data driven and focuses on user-and problem-specific parameters instead of using pre-defined models. This anomaly detection method is composed of two stages where in the first stage, all the potential global anomalies are selected based on the data density, and in the following stage, local anomalies are marked based on the data clouds formed from the potential global anomalies. This paper takes a similar conceptual approach where local anomalies are identified by concentrating on the temporal changes in the given time-series.

Density based approaches have also shown promising results in detecting anomalies. For instance, k nearest neighbor (kNN) [30]–[32], isolation forest [33] and DBSCAN clustering [34].

Machine learning techniques for anomaly detection can be categorized in three ways based on the availability of labeled data-set. These are supervised, semi-supervised, unsupervised. A non-exhaustive list machine learning techniques can be found in [4] [8]. Recently, neural networks, especially, Long-short-term-memory (LSTM) neural network has been used to detect anomalies in time-series [35] [36]. A probabilistic method for time-series anomaly detection in time-series is presented in [37] where a Bayesian maximum likelihood classifier is used to learn the temporal sequence and detect anomalies based on that learning. Disadvantages of machine learning techniques are that they require either exhaustive labeling (for classification techniques) or depend on the initial hyper parameters setting (clustering based approaches). Besides, these techniques are usually resource greedy and often not optimized to detect anomalies.

III. Anomaly detection in traffic flows

The data that have been used for the evaluation of the proposed method contains vehicle flow data. The estimation of the traffic flow, based on counting the number of vehicles that cross a given location during some time interval is an important descriptor of a complex traffic system. The vehicle flow usually varies over the day and between different days of the week. The data were collected by inductive loop detectors (LD).

PTV Group SISTeMA [38] has provided total 4 years (2011-2014) of traffic data which were collected from the road network of Vienna, Austria. A total of 280 LD data were made available to us. For each LD, there are records captured in 15-minutes interval, from January 2011 to December 2014. Hence, each day yields 96 records and each year 35,040 records for each LD (an exception is 2012 which, being a leap year, yields 35,136 records for each LD). For this paper, we considered data from 95 such LDs within 2011.

Almost all LDs have given 0 as the minimum vehicle count (missing values are not treated as a record) whereas maximums have differed and include some very high numbers (e.g. 12000 per 15 mins). These type of maximums are certainly produced

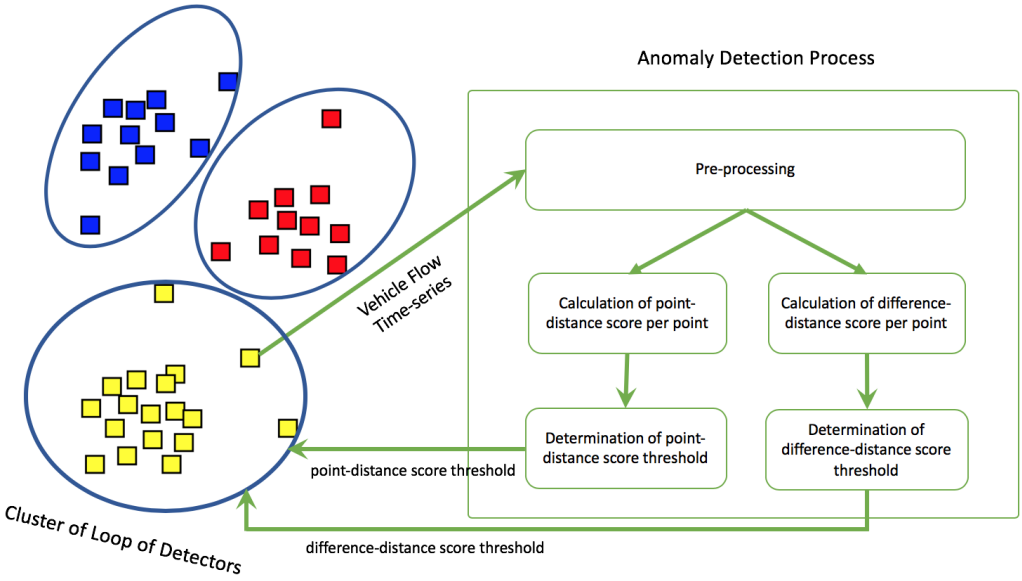

Fig. 1. Overview of anomaly detection process

by some sort of malfunctioning in the LD according to several studies on vehicle counts in different road networks [18].

A. Overview

Our proposed anomaly detection method for traffic flow time series has different components. Figure 1 provides an overview of the different components, which work in the following way. First all LD data are clustered based on their internal characteristics (i.e. without considering any context data such as the position of each LD) (Section III.B). After some basic pre-processing that removes null and out-of-bounds values is performed ((Section III.C)), for each identified cluster one LDs is selected for identifying anomalies in their time series. For a time series, anomaly detection is performed by relying on its seasonality and calculating two scores for each point, point-distance and difference-distance, that measure the anomalousness of the point (Section III.D).

In the next step, if an expert user is able to participate, an interactive threshold selection process for each score is employed. If an expert user is not available or user interaction is not desired, an automatic threshold selection process is employed (Section III.E).

When score thresholds are identified for a single LD, the same thresholds are employed in identifying anomalies in all the LDs that belong to the same cluster. This way, we provide a tunable and scalable anomaly detection method.

B. Clustering of loop detectors

In this section, we describe the method we followed in grouping or clustering loop detectors (LDs) into groups of similar ones. The motivation for doing this was to be able to use the same–learned via user interaction with a single LD–thresholds across all LDs in the same group. Our approach relies solely on the time series data available for each LD and not on the detector's characteristics or context (e.g. physical position).

For clustering, we considered 95 LDs for which we obtained 15-min measurements for each day in 2011. As a first step, we transformed the raw time series for each LD by calculating, for each time slot in a day of the week (00:00 on Mondays, 00:15 on Mondays, ..., 23:45 on Sundays) the median over all the weeks in 2011. In each case, the median was calculated over *at most* 52 points (number of weeks in a year), since some measurements were just absent in the raw data. As a result, for each LD we obtained a new time series–median time series–with 672 points (96 points in a day multiplied by 7 days). We consider these time series as representatives of the normal behavior of each LD.

We then focused on clustering the set of median time series. For this, we used *k-means*, a well-known algorithm for clustering of datasets [39]. In k-means, given a number of clusters k, the algorithm iteratively tries to find a centroid for each cluster so that the sum of the squared Euclidean distances between an observation assigned to the cluster and the clusters centroid is minimized. For each LD, we calculated the Dynamic Time Warping (DTW) [40] distance of its median time series from the median time series of all LD, including itself. DTW distance is a well-known algorithm for measuring the similarity between two time series by pair-wise comparison of their (potentially slightly shifted in time) data points. For each LD, the 95 DTW distances were the *features* fed into the k-means algorithm. We note here that alternatives exist both the clustering algorithm (e.g. xmeans) and on the features to consider (e.g. one could select the median, max, min, and variance of each median time series as features). Our intuition for using the pairwise comparison with the DTW distance was to utilize as much information about the shape and the

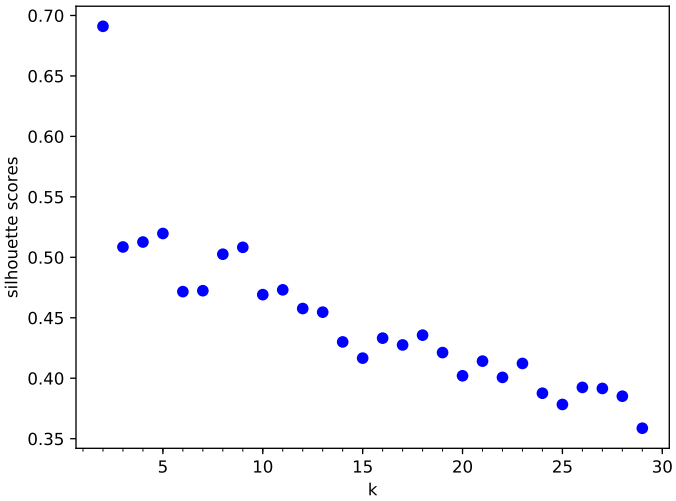

Fig. 2. Average Silhouette scores for different number of clusters: the higher the better.

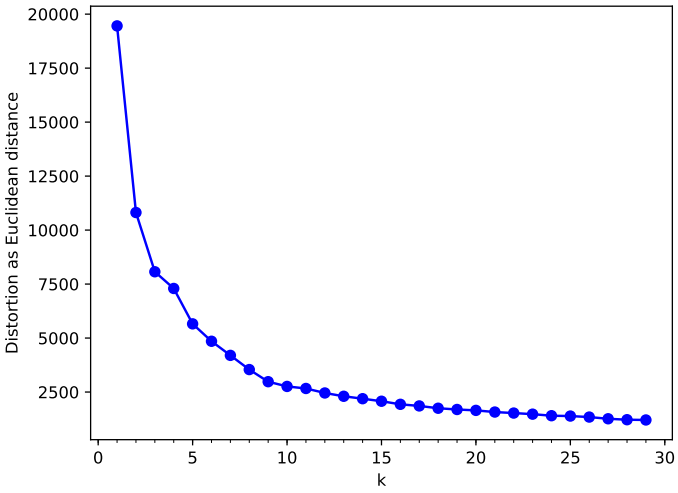

Fig. 3. Elbow method.

evolution of the time series as possible.

One important consideration when using k-means is which number of clusters–*k*–to use. We applied k-means with different number of clusters starting from 2 to 30. For each application, we calculated the average Silhouette score of the clustering [41]. The Silhouette score for a datum is calculated by:

$$a = \frac{b - a}{max(a, b)} \tag{1}$$

where a is the average Euclidean distance between the datum and other data in its cluster and b is the average Euclidean distance between the datum and other data in the next nearest cluster. Silhouette scores take values within [-1,1] with values close to 1 indicating a good match of the datum to the cluster. We calculated the average Silhouette score for each clustering by considering all data points– this provides a measure of how well the data are assigned to clusters in that clustering [41]. The average Silhouette scores for the different number of clusters we considered are depicted in Figure 2. As can be seen, the best number of clusters according to the Silhouette method is 2, followed by 5.

A popular alternative to this method is the elbow method, which relies on visually inspecting the graph of number of clusters versus a distortion metric (e.g. sum of squared distances of each point to its cluster's centroid) and identifying the point of inflection in the graph. When applying the elbow method we obtained results similar to the Silhouette method, namely that the best number of clusters are 2 and 5 (Figure 3).

As can be seen from the application of the above clustering criteria and especially of the Elbow method, many times the optimal number of clusters is not clear-cut. Nevertheless, in the following, we consider that this number is 2. In this case, the 95 LDs are split into 2 clusters of 77 and 18 LDs. Figure 4 shows the formation of the two clusters in a three-dimensional space obtained via applying Principal Component Analysis

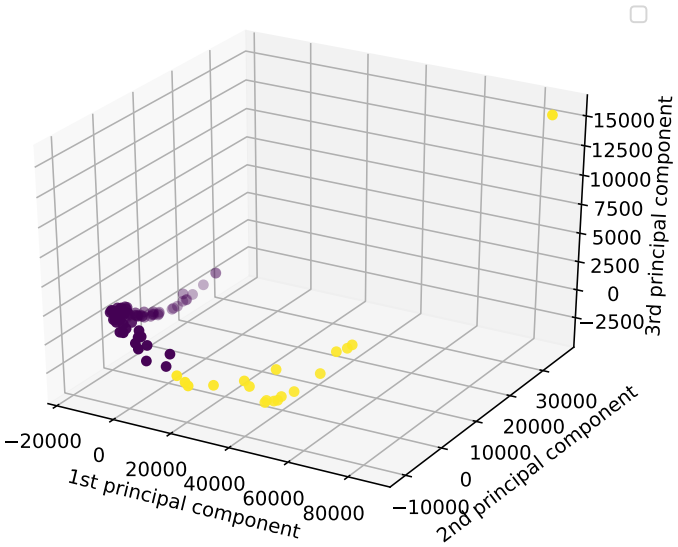

Fig. 4. The formation of the two clusters in the 3D space of the first three principal component dimensions returned by PCA. Yellow points belong to the small cluster while violet to the big cluster.

(PCA), a well-known method for dimensionality reduction, on the median time series data.

The two clusters differ from each other mainly on the maximum of the medians reached for a particular slot in a day of the week: in the large cluster (77 LDs) such peaks reach 100-400 counts, whereas in the small cluster (18 LDs) they range from 400 to 600 counts, with a single exception (depicted in the top right corner of Figure 4 of a LD having peaks at 800-1000 counts).

For illustration, Figure 5 depicts the median time series from five LD belonging to the large cluster and five belonging to the small cluster (dotted lines).

C. Pre-processing

Each loop detector has null values presented in the time-series due to probable malfunctioning of hardware, communi-

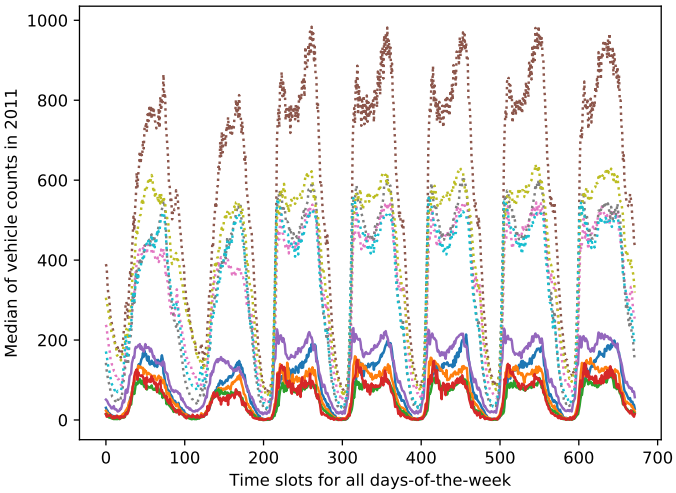

Fig. 5. Examples of median time series for each cluster. Time series of loop detectors of the small cluster are marked in dotted lines and of the big cluster in solid lines.

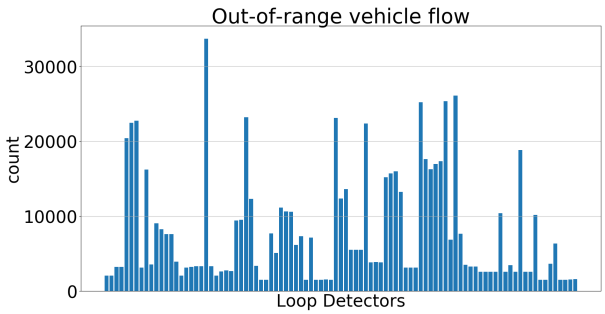

Fig. 6. Count of out of range vehicle flow records

cation failure etc. Figure 6 shows count of such null values for the 95 loop detectors we have studied.

The identification of outliers in the pre-processing phase can be done with the knowledge of lower bound of vehicle flow. As lower bound of the range, zero can be used since negative vehicle flow is not possible. Higher bound is not necessary since these outliers are going to be detected by next phases of the process. Since our datasets did not contain any negative values, the pre-processing steps only removed null values.

D. Anomaly scores

One important component of time-series is seasonality. Frequent seasonalities that are observed in time-series over a year span are hourly, daily, weekly, bi-weekly, monthly, quarterly etc [42]. A time-series with interval less than a day usually shows complex seasonality—multiple (e.g. both weekly and daily) seasonality in the same time-series. Based on results of our previous work [43], the time series in our study show stronger weekly than daily seasonality.

Based on this, for each time slot in a day of the week (e.g. 00:00 on Mondays) we construct a partial time series that contains all the corresponding measurements for each week

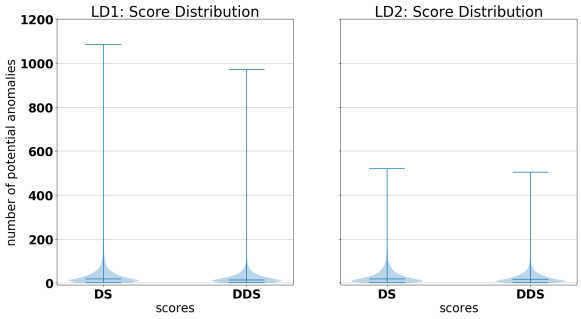

Fig. 7. Violin plots of score distributions for LD1 and LD2.

in the year (a maximum of 52 points). We then calculate the 25th and 75th percentile of the partial time series. Finally, we calculate the distance of each point that lies outside of the 25th-75th band from the 25th percentile, if the point is below the band or the 75th percentile, the point is above the band. We call this distance *point-distance score*. Points that lie within the 25th-75th band get assigned a point-distance score of 0.

Similarly, for each time slot in a day of the week we construct a differenced partial time series by taking the difference from the previous point in the original time series. We then again calculate the 25th and 75th percentile of each differenced partial time series and calculate the distance from points that lie out of this band from the border of the band. We call this distance *difference-distance score*. Points that lie within the 25th-75th band get assigned a difference-distance score of 0.

For illustration, we calculated the two scores for two LD, *LD1* and *LD2*. Figure 7 depicts violin plots showing their distribution. As can be seen, the majority of scores are less than 100 while only a handful of scores are more than 400. Clearly, a high point-distance score for a point in a time series indicates that the point lies far from the behavior that is expected due to the weekly seasonality of the time series. Similarly, a high difference-distance score indicates that there has been a rise or drop from the previous measurement that is unusual if one considers the expected rises or drops due to the seasonality of the time series.

The question that is still open is when is a point-distance and difference-distance score of a point high enough so that it can be consider an anomaly. We tackle this by identifying thresholds for each score, above which points are detected as anomalies in our method as described next.

E. Identification of score thresholds

We consider two types of threshold selection process depending on expert user availability: interactive threshold selection and z-score-based threshold selection.

Interactive threshold selection: To select a threshold for each score an interactive process is used which follows the idea of binary search and comprises several steps for each score. For each score, at each step, two plots for a number of potential anomalies (we considered 5) are presented to the expert user. The first plot is a line chart of the time series

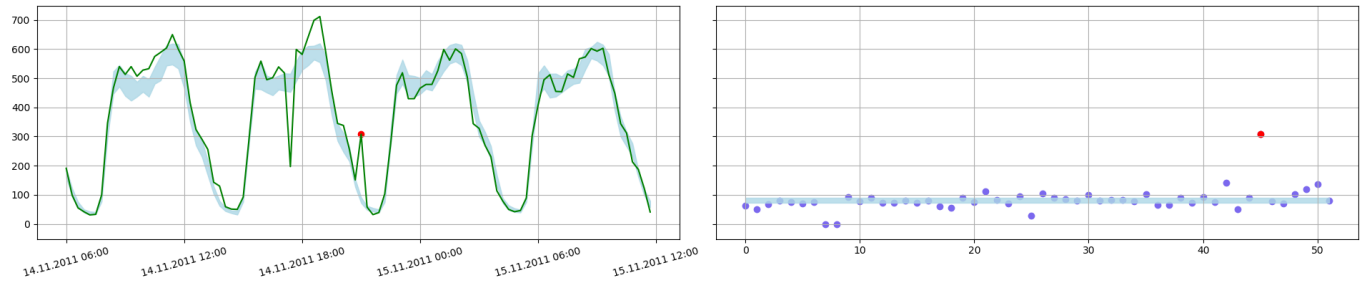

Fig. 8. Point-distance threshold selection scatter plot example

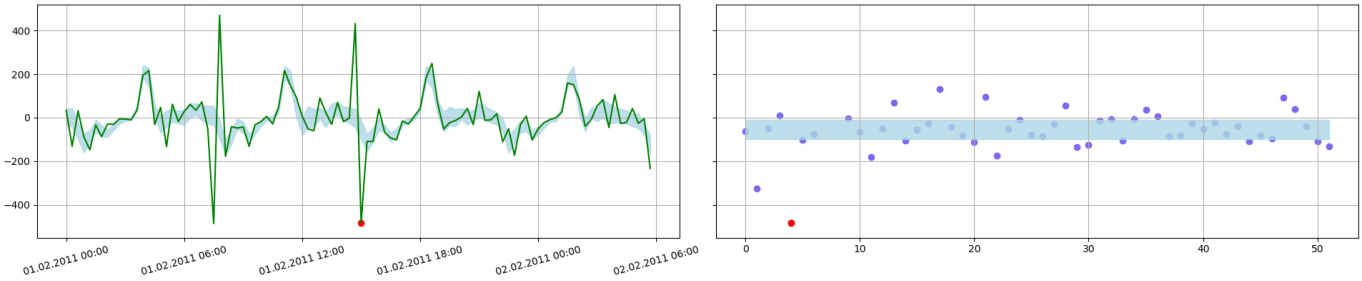

Fig. 9. Difference-distance threshold selection scatter plot example

around the potential anomaly, whereas the second plot is a scatter plot of the partial time series the point belongs to. In both plots, the potential anomaly is marked in red and the 25th-75th band is depicted in the background to help the user take an informed decision. An example of such a plot pair for the point-distance score is depicted in Figure 8, whereas an example for the difference-distance score is depicted in Figure 9.

After showing these plots, the expert is asked to decide whether most of the points marked as red are anomalies or not. Such judgment may be based on the reason for which anomaly detection is performed: in some cases it would make sense to tolerate more anomalies than in others. The expert answers with yes or no and based on this the threshold is updated and new plots are shown.

The process starts by setting the threshold to the average between the maximum and the minimum score values. In a positive answer from the expert, the new threshold becomes the average of the old threshold and the minimum value used in the calculation of the old threshold, so essentially anomaly detection becomes stricter. Contrary, in a negative answer, the threshold moves up and the detection becomes less strict.

The interactive threshold selection process has at most $log_2 n$ steps, where n is the difference between the maximum and minimum of the scores. For instance, for LD1, the plots are shown *at most* 10 times. When the process finishes, the threshold used in the last positively answered step becomes the final one.

For illustration, Figure 10 depicts the different steps (corresponding to different slices of the data) in the interactive

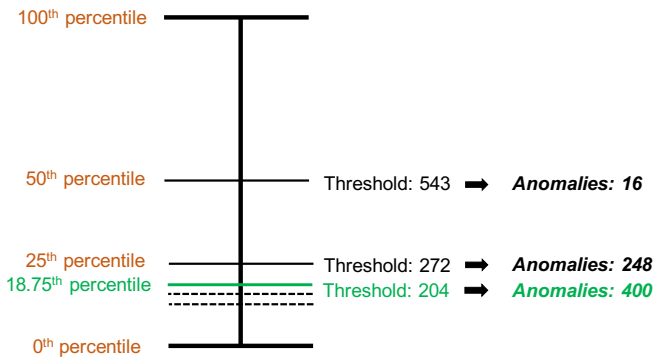

Fig. 10. Interactive threshold selection for point-distance score for LD1. Thresholds answered with "yes" are solid lines, with "no" are dashed. The final threhold is in green.

threshold selection in LD1. As can be seen, after 5 steps, 3 of which were positively answered, the threshold for point-distance score for LD1 was set to 204. In this process, the expert was assumed to take a "high tolerance" approach towards anomaly detection, which might miss some anomalies but should have high precision. We also experimented with "low-tolerance" approaches of expert users. Table I depicts the final thresholds for both scores following a high tolerance versus a low tolerance approach for LD1 and LD2. Clearly, the number of identified anomalies depends a lot on the aggressiveness of the expert user when following the interactive threshold selection process.

Threshold selection based on modified z-score: Our method also offers the option of determining the threshold for

		High Tolerance		Low Tolerance	
		Threshold value	# Anomalies	Threshold value	# Anomalies
LD1	DST	204	400	47	3372
	DDST	183	224	31	2809
LD2	DST	98	843	70	1589
	DDST	85	288	35	2416

TABLE I
DIFFERENT THRESHOLDS AND CORRESPONDING REPORTED ANOMALY COUNTS.

each score via a statistical method that requires no (expert) user input. In particular, for each potential anomaly pa we calculate its modified z-score, a standardized score for outlier detection:

$$\frac{0.6745(score_{pa} - median)}{median\_absolute\_deviation}$$

If the modified z-score of a point is greater than 3.5, then the point should be marked as anomalous [44]. With this in mind, we set the threshold of the point-/difference-score to the point-/difference-score of the first potential anomaly whose z-score score is greater than 3.5.

F. Reuse of score thresholds

Once the score threshold for a single LD is determined, they can then be reused for the other LDs of the same cluster. To do this, we first normalize them to percentages using the following formula:

$$threshold_{norm} = \frac{threshold}{max(score) - min(score)} \times 100 \quad (2)$$

Then, we perform the opposite operation to derive a threshold from a percentage for each LD in the same cluster:

$$threshold = \frac{threshold_{norm}}{100} \times (max(score) - min(score)) \quad (3)$$

IV. EVALUATION

In evaluating our method, we were interested in determining (a) the accuracy, precision, recall and F1 measure of our method in determining anomalies for a single LD, and (b) whether the thresholds identified for a LD can be reused for other LDs in the same cluster with comparable performance metrics.

A. Method's accuracy, recall, precision and F1 measure

To quantify the accuracy, recall, precision and f1 measure) of our method we needed to obtain a baseline. For this, one of the authors, a traffic engineer, manually annotated anomalies for LD1 and LD2 for 2011. These two LDs belonged to the same cluster and in particular to the small cluster out of the two clusters derived in the clustering analysis of the 95 LDs described in Section III.B.

We note that the engineer worked with their own preferred and without knowing how our anomaly detection method was performed to detect anomalies in the same data-sets. As a first indicator of the effort gains of our method, the manual labeling process took about 90 minutes, whereas the interactive labeling process in our method took approx. 5 minutes only.

	High Tolerance	Low Tolerance	Modified z-score
Accuracy	98.5%	86.7%	95.7%
Recall	83.9%	98.3%	95.3%
Precision	90.8%	31.2%	59.5%
F1Score	87.2%	47.3%	73.2%

TABLE II
ACCURACY, RECALL, PRECISION AND F1SCORE FOR EACH THRESHOLD FOR LD1.

	High Tolerance	Low Tolerance	Modified z-score
Accuracy	96.9%	90.5%	96.1%
Recall	88.8%	94.1%	90.1%
Precision	75.2%	44.1%	68.2%
F1Score	81.4%	60.0%	77.6%

TABLE III
ACCURACY, RECALL, PRECISION AND F1SCORE FOR EACH THRESHOLD FOR LD2

The accuracy, precision, recall and F1 score for LD1 and LD2, for three variants of our method (high tolerance, low tolerance, modified z-score) are presented in Tables II and III respectively. The same data are also graphically depicted in Figures 11 and 12.

In general, our method consistently shows high accuracy across all variants in both loop detectors. As expected, the low tolerance approach increased the number of both true and false positives hence increasing recall and reducing precision. Interestingly, the modified z-score approach stroke a balance between the two approaches that required human input in the considered performance metrics.

B. Reuse of thresholds

As an example of re-use of thresholds in the same cluster we re-used the thresholds from LD1 to LD2 and compared with thresholds of LD2 decided through the interactive selection process. Figure 13 shows the comparison. As can be seen, the reused threshold have almost the same impact to the performance metrics as the thresholds derived from the interactive process. This is a first proof that thresholds can indeed be reused in our method.

V. DISCUSSION

Accuracy of detection: Measuring the accuracy of different anomaly detection use cases (that map e.g. to our simplified low/high tolerance approaches) requires a labelled data set for each use case. Based on our evaluation, we are confident that our approach can be used to detect anomalies in different use cases that deal with univariate traffic flow time-series with high accuracy. We note that performing purely manual anomaly detection is itself prone to wrongly annotated anomalies as well as missing actual anomalies due to the fact that seasonal patterns require dedicated visualization support to comprehend. In case of large number of time series from LDs where a purely manual approach is not really an option, our approach provides a viable alternative without sacrificing

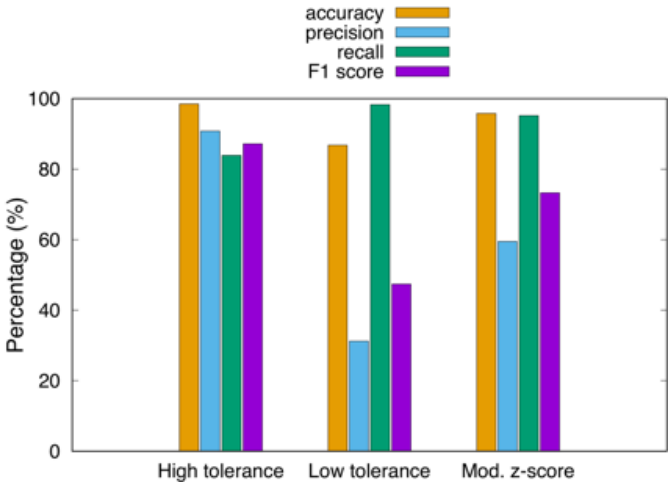

Fig. 11. Evaluation of different thresholds against manually annotated baseline for LD1.

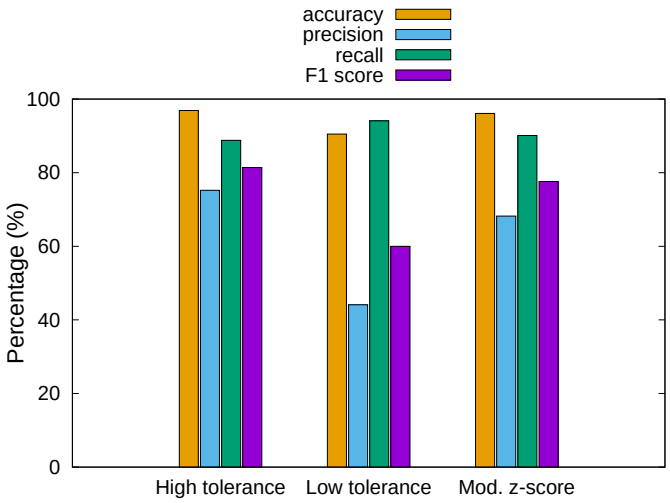

Fig. 12. Evaluation of different thresholds against manually annotated baseline for LD2.

accuracy to a big extent and scales with the amount of manual effort needed (since only one LD per cluster has to analyzed).

Quantification, interpretability: An advantage of our anomaly detection method is that its results are easy to interpret. In essence, an anomaly is detected when a value does not follow a certain seasonal pattern or its deviation does not follow a certain seasonal pattern. By providing different scores for each case we are able to better understand the reason and degree of anomalousness of a point.

Knowing the reason and degree of anomalousness helps in deciding what to do with the detected anomaly. For example, anomaly with a low score can still be included in forecasting. A sequence of low scored anomalies could be further analyzed and mapped to a specific event. Large-score anomalies can be removed from dataset for better modeling the traffic behavior as well as for more accurate predictions.

Adaptability: By separating the anomaly scores from the corresponding thresholds we offer the opportunity to set

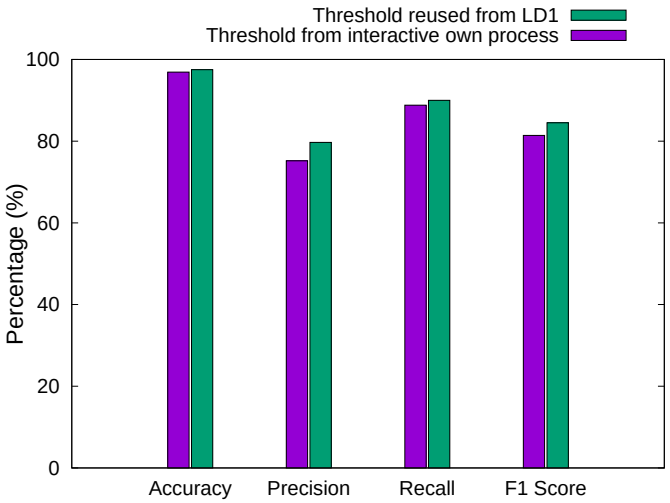

Fig. 13. Using high tolerance threshold from LD1 to LD2.

different thresholds that serve different needs or use cases, effectively adapting or tailoring the anomaly detection process in a natural way. For example, different types of alerts can be set out based on different use cases. An important point here is that our method does not need to run the whole process of calculating scores again in order to be tailored to a different use case.

Limitations: The proposed anomaly detection method has also few limitations:

- It can only with regularly spaced, univariate timeseries. In many real-life cases, there is a clear need for correlating data from more than one time series (e.g. vehicle speeds and counts) in order to identify anomalies. This is a topic of our future work.
- The way we perform clustering of time series taking into account all pairwise comparisons is computationally intensive and may pose scalability challenges for really large number of LDs. Fortunately, (i) clustering needs to be performed only once, and (ii) instead of pairwise comparisons, statistical metrics over each time series (e.g. median, variance) can be used in as clustering features for performance reasons.

VI. Conclusion

Traffic measurements from sensors such as loop detectors typically contain a number of irregularities or anomalies that can severely impact traffic analysis and prediction. In this paper, we described a lighweight method for detecting anomalies in traffic flow time series that relies on quantifying the degree of anomalousness of individual points and on reusing expert knowledge to expedite the process of identifying anomalies across time series with similar characteristics. Among the strengths of the approach is that the results are easily to interpret and the method itself is easy to tune to one's needs.

References

[1] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley Sons, 1994.

[2] O. Deniz and H. B. Celikoglu, "Overview to some existing incident detection algorithms: a comparative evaluation," *Procedia-Social and Behavioral Sciences*, vol. 2, pp. 153–168, 2011.

[3] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[5] H. S. Teng, K. Chen, and S. C. Lu, "Adaptive real-time anomaly detection using inductively generated sequential patterns," in *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*. IEEE, 1990, pp. 278–284.

[6] A. M. Leroy and P. J. Rousseeuw, "Robust regression and outlier detection," *J. Wiley&Sons, New York*, 1987.

[7] M. Markou and S. Singh, "Novelty detection: a reviewpart 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[8] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[9] J. W. Tukey, *Exploratory data analysis*. Reading, Mass., 1977, vol. 2.

[10] L. M. Kieu, A. Bhaskar, and E. Chung, "Bus and car travel time on urban networks : integrating Bluetooth and bus vehicle identification data," in *25th ARRB Conference : Shaping the future: Linking Policy, Research and Outcomes*, Perth, WA., 2012.

[11] F. E. Grubbs *et al.*, "Sample criteria for testing outlying observations," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 27–58, 1950.

[12] B. Rosner, "On the detection of many outliers," *Technometrics*, vol. 17, no. 2, pp. 221–227, 1975.

[13] B. Agrawal, T. Wiktorski, and C. Rong, "Adaptive real-time anomaly detection in cloud infrastructures," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 24, p. e4193, 2017.

[14] R. G. Vieira, M. A. Leone Filho, and R. Semolini, "An enhanced seasonal-hybrid esd technique for robust anomaly detection on time series," in *Simpósio Brasileiro de Redes de Computadores (SBRC)*, vol. 36, 2018.

[15] B. Abraham and G. E. Box, "Bayesian analysis of some outlier problems in time series," *Biometrika*, vol. 66, no. 2, pp. 229–236, 1979.

[16] R. D. Martin and V. J. Yohai, "Influence functionals for time series," *The annals of Statistics*, pp. 781–818, 1986.

[17] A. J. Fox, "Outliers in time series," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 350–363, 1972.

[18] I. Chang, G. Tiao *et al.*, "Effect of exogenous interventions on the estimation of time series parameters," in *Proceedings of the American Statistical Association, Business and Economics Statistics Section*, 1983, pp. 532–537.

[19] W. R. Bell and S. C. Hillmer, "Modeling time series with calendar variation," *Journal of the American statistical Association*, vol. 78, no. 383, pp. 526–534, 1983.

[20] R. S. Tsay, "Time series model specification in the presence of outliers," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 132–141, 1986.

[21] A. Smith and M. West, "Monitoring renal transplants: an application of the multiprocess kalman filter," *Biometrics*, pp. 867–878, 1983.

[22] M. West, P. J. Harrison, and H. S. Migon, "Dynamic generalized linear models and bayesian forecasting," *Journal of the American Statistical Association*, vol. 80, no. 389, pp. 73–83, 1985.

[23] G. M. Weiss and H. Hirsh, "Learning to predict rare events in event sequences." in *KDD*, 1998, pp. 359–363.

[24] F. J. Anscombe, "Rejection of outliers," *Technometrics*, vol. 2, no. 2, pp. 123–146, 1960.

[25] P. H. Torr and D. W. Murray, "Outlier detection and motion segmentation," in *Sensor Fusion VI*, vol. 2059. International Society for Optics and Photonics, 1993, pp. 432–444.

[26] K. Kadota, D. Tominaga, Y. Akiyama, and K. Takahashi, "Detecting outlying samples in microarray data: A critical assessment of the effect of outliers on sample classification," *Chem-Bio Informatics Journal*, vol. 3, no. 1, pp. 30–45, 2003.

[27] A. M. Bianco, M. Garcia Ben, E. Martinez, and V. J. Yohai, "Outlier detection in regression models with arima errors using robust estimates," *Journal of Forecasting*, vol. 20, no. 8, pp. 565–579, 2001.

[28] D. Chen, X. Shao, B. Hu, and Q. Su, "Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra," *Analytical Sciences*, vol. 21, no. 2, pp. 161–166, 2005.

[29] P. P. Angelov and X. Gu, "Anomaly detectionempirical approach," in *Empirical Approach to Machine Learning*. Springer, 2019, pp. 157–173.

[30] E. M. Knorr, R. T. Ng, and V. Tukacov, "Distance-based outliers: Algorithms and applications," *International Journal on Very Large Data Bases*, vol. 8 (3-4), pp. 237–253, 2000.

[31] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD '00*, p. 427, 2000.

[32] F. Angiulli and C. Pizzuti, "Fast Outlier Detection in High Dimensional Spaces. Principles of Data Mining and Knowledge Discovery," *Lecture Notes in Computer Science*, vol. 2431, p. 15, 2002.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[34] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[35] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.

[36] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[37] H. N. Akouemo and R. J. Povinelli, "Probabilistic anomaly detection in natural gas time series data," *International Journal of Forecasting*, vol. 32, no. 3, pp. 948–956, 2016.

[38] "PTV Group SISTeMA," https://www.ptvgroup.com/it/, via Bonghi 11b, Rome, Italy.

[39] S. Na, L. Xumin, and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, 2010, pp. 63–67.

[40] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[41] A. Starczewski and A. Krzyak, "Performance Evaluation of the Silhouette Index," in *Artificial Intelligence and Soft Computing*, ser. Lecture Notes in Computer Science. Springer, 2015, pp. 49–58.

[42] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[43] M. R. Alam, I. Gerostathopoulos, C. Prehofer, A. Attanasi, and T. Bures, "A framework for tunable anomaly detection," in *Proceedings of International Conference on Software Architecture, to appear.*, 2019.

[44] B. Iglewicz and D. C. Hoaglin, *How to Detect and Handle Outliers*. ASQC Quality Press, 1993, google-Books-ID: siInAQAAIAAJ.