(Article begins on next page)

Semi real-time data cleaning of spatially correlated data in traffic sensor networks

Federica Rollo[a], Chiara Bachechi[b] and Laura Po[c]
"Enzo Ferrari" Engineering Department, University of Modena and Reggio Emilia, Italy
{name.surname}@unimore.it

Keywords: IoT, traffic model, anomaly detection, sensor faults, big data streams, correlation, correlated sensors

Abstract: The new Internet of Things (IoT) era is submerging smart cities with data. Various types of sensors are widely used to collect massive amounts of data and to feed several systems such as surveillance, environmental monitoring, and disaster management. In these systems, sensors are deployed to make decisions or to predict an event. However, the accuracy of such decisions or predictions depends upon the reliability of the sensor data. By their nature, sensors are prone to errors, therefore identifying and filtering anomalies is extremely important. This paper proposes an anomaly detection and classification methodology for spatially correlated data of traffic sensors that combines different techniques and is able to distinguish between traffic sensor faults and unusual traffic conditions. The reliability of this methodology has been tested on real-world data. The application on two days affected by car accidents reveals that our approach can detect unusual traffic conditions. Moreover, the data cleaning process could enhance traffic management by ameliorating the traffic model performances.

1 INTRODUCTION

Public Administrations have begun to capture the large amount of data collected through IoT sensors in order to face the big challenge of sustainable development. Nowadays, many cities are equipped with traffic sensors installed on their road networks. The most diffuse sensor type is the induction loop: static sensors that are embedded under the road surface and provide real-time vehicle count and speed estimation. These data can be used as input to simulate real-time traffic scenarios that can effectively help Public Administration to cope with the mobility challenge – and instantaneously optimizing the transportation flow while sending new instructions to smart city devices like traffic lights. Traffic sensors are of great value for urban traffic modeling. However, they are not free of errors and faults, and the degradation of sensor performance can heavily affect the output of traffic model (Bachechi et al., 2020c). Therefore, detecting faulty traffic sensors is a fundamental step in order to boost the quality of the traffic management system (Bachechi et al., 2020d; Bachechi et al., 2021; Bachechi et al., 2022a; Desimoni et al., 2020). On the other hand, anomalies in traffic sensor observations can also derive from non-conventional traffic conditions such as accidents, slowdowns, or street closures, representing a consequent change in the environment. Road traffic congestion causes a waste of time and money, promptly assessing the occurrence of traffic anomalies can minimize the impact and duration of a road accident and the resulting traffic congestion. The first step to assess the imminent emergence of car accidents is to detect deviations from normal traffic patterns.

The goal of this paper is to define an anomaly detection and classification methodology that can be applied to massive traffic data streams in semi-real-time. Our approach can be applied to traffic sensors that measure both flow and speed. The methodology is discussed and applied in a real scenario, the traffic sensor network of Modena, an Italian city in the Emilia-Romagna region.

The rest of this paper is structured as follows: Section 2 describes related work, while the methodology is detailed in Section 3. The use case and the configuration of the data cleaning process are discussed in Section 4. Section 4.3 describes the results obtained and the comparison with real traffic conditions underlined in newspapers on two different days. Conclu-

[a] https://orcid.org/0000-0002-3834-3629
[b] https://orcid.org/0000-0003-2323-0573
[c] https://orcid.org/0000-0002-3345-176X

sions and future work are sketched in Section 5.

2 RELATED WORK

Anomaly detection in time-series is a research area of data science and machine learning that has received much attention. With sensors pervading our everyday lives, we see an exponential increase in the availability of streaming, time-series data. In the literature, we find supervised, unsupervised, and semi-supervised anomaly detection algorithms (Görnitz et al., 2012; Ramchandran and Sangaiah, 2018). However, several methods are formerly created for processing data in batches, and unsuitable for real-time streaming applications.

Several techniques have also been employed in the context of sensor fault detection (Zamini and Hasheminejad, 2019; Zhang et al., 2020; Chander and Kumaravelan, 2022; O'Reilly et al., 2014). ARIMA (Autoregressive Integrated Moving Average) is a general-purpose technique effective at detecting anomalies in data with regular daily or weekly patterns. Extensions of ARIMA enable the automatic determination of seasonality. ARIMA has also been applied in the context of traffic anomaly detection that considers imbalanced, non-stationary properties of the traffic sensor network (Yu et al., 2016; Zare Moayedi and Masnadi-Shirazi, 2008), and it showed remarkable detection precision and real-time performance. In (Kurian et al., 2015), a system to automatically diagnose faults in induction loop sensors measurements is described. The system is based on an impulse test and thus requires to develop an embedded circuit. In this paper, we will focus on automatic fault recognition methodologies that do not require any embedded system. A possible approach is described in (Zygouras et al., 2015); in this study, faulty readings from traffic sensors are identified by examining the correlations among them and by taking advantage of the ubiquitous citizens through crowdsourced data. The authors evaluate cross-correlation between sensors using the Pearson metric, and then employing a multivariate ARIMA model to detect anomalies considering the correlated sensors. Moshtaghi et al. (Moshtaghi et al., 2011) proposed a clustering method called Forgetting Factor Iterative Data Capture Anomaly Detection (FFIDCAD) for online anomaly detection on normal data. The novel approach of FFIDCAD inspired the development of other algorithms for the identification of events in sensor network (Ali et al., 2015).

From the best of our knowledge, ARIMA has never been tested in combination with FFIDCAD in the context of traffic anomalies detection, nor in combination with the correlation among traffic sensors; our paper is a new example of this combination.

3 METHODOLOGY

Traffic sensors measurements are multivariate spatial time series, since they provide information about two variables: the traffic flow and the average speed of vehicles. Besides, the two variables are not independent: the number of vehicles and their average speed are correlated. The methodology we developed to detect anomalies and distinguish between sensor faults and unusual traffic conditions is composed of three steps:

- *Studying the correlation among traffic sensors*: this phase consists of an analysis of the sensor network. The scope is to identify groups of traffic sensors whose measurements are correlated by studying historical time series, as described in (Bachechi et al., 2022b). Two sensors are considered correlated if their Detrending Cross-Correlation Analysis Coefficient (DCCA) correlation coefficient is higher or equal to 0.7 in an interval of one hour and their distance is lower than 2500 meters.
- *Anomaly detection*: abnormal observations that deviate from the vast common behavior of the sensors are discovered for each sensor.
- *Anomaly classification*: anomalies are classified as sensor faults or unusual traffic conditions. In (Bachechi et al., 2022b), the classification methodology is described in detail. Each anomaly is associated with anomalies identified in an adjacent time interval. The amplitude of this time interval should be defined considering the frequency of observations. Anomalies occurring in adjacent time intervals in correlated sensors are considered unusual traffic conditions. The anomalies observed for sensors with a low number of correlated sensors are more likely to be classified as sensor faults. For this reason, for each anomaly classified as sensor fault, the distance between the sensor it belongs to and each traffic sensor showing a simultaneous anomaly was evaluated. If there are at least two other sensors experiencing an anomaly in a radius of 1500 meters, the anomaly is classified as an unusual traffic condition. The remaining anomalies are sensor faults.

In the following sections, we describe in detail the techniques employed for anomaly detection.

3.1 Anomaly detection techniques

Anomaly is defined as a point in time when the behavior of the system is unusual and significantly different from previous, normal behavior (Chandola et al., 2008). The most common way to detect anomalies is by modeling the average trend of data and detecting deviations from the trend. We are interested in techniques that allow detecting anomalies in real-time or semi real-time. We identified three techniques to find anomalies on the traffic sensors data streams. Firstly, the flow-speed correlation filter is applied to remove the flow and speed values that seem inconsistent if considered related to one another. The filter was described in detail in (Bachechi et al., 2022b) and is based on the idea that, in a fixed time interval, there is a maximum number of vehicles that can pass on a road at a certain speed. Other anomalous measurements can be detected by combining the FFIDCAD model and the ARIMA model.

FFIDCAD is an iterative and multivariate anomaly detection algorithm (Moshtaghi et al., 2011). This algorithm assumes the data fit the multivariate normal distribution and exploits the correlation among different features of data to identify the anomalies. The elements of the input dataset are multidimensional feature vectors. The scope is to find a set of clusters that group the elements of the dataset. The boundary of the clusters are defined by hyperellipsoids. The algorithm employs a continuous learning strategy to estimate the hyperellipsoidal shape that covers the data incrementally; each iteration of the algorithm adjusts the hyperellipsoidal model based on the measurements up to the current time. The values of mean, standard deviation, and covariance of the correlated features are used to build the hyperellipsoids. The out of the bound instances are classified as anomalous data. At iteration k, the elements in the hyperellipsoid are defined by the formula:

$$ell_k(m_k, S_k^{-1}, t) = \{x \in \mathbb{R}^d \mid (x - m_k)^T S_k^{-1}(x - m_k) \leq t^2\}$$

where m_k is the array containing the mean of the features, x is the current data point, and S_k^{-1} is the precision matrix, i.e., the inverse of the covariance. The diagonal elements of the matrix measure how the variables are clustered around the mean, while the off-diagonal elements express the independence of the input features. t^2 is the confidence space of the data distribution, i.e., the range of values that we expect to be non-anomalous. The statistical p-value is used to define the confidence space. For example, if $p = 0.98$, then the ellipsoid will cover the 98% of the data. In other words, a data point has 98% probability to be an acceptable value (near to the mean of the sample). p should be set based on the assumption that anomalies are rare in the dataset. In traffic sensors, the correlated features for the definition of the hyperellipsoids are flow and speed. The detailed analysis of sensor data provided in (Bachechi et al., 2022b) reports that traffic data are non-stationary time series. To increase the tracking capabilities of the model in non-stationary environments, a forgetting factor $\lambda \in (0,1)$ was introduced when updating the parameters of the model. The mean is updated incrementally by the formula:

$$m_k = \lambda m_{k-1} + (1 - \lambda)x$$

At iteration $k+1$, the precision matrix is updated as:

$$S_{k+1}^{-1} = \frac{kS_k^{-1}}{k-1}\left[I - \frac{(x_{k+1} - m_k)(x_{k+1} - m_k)^T S_k^{-1}}{\frac{k^2-1}{k} + (x_{k+1} - m_k)^T S_k^{-1}(x_{k+1} - m_k)}\right]$$

ARIMA is a statistical method for time series analysis and forecast. It is able to model temporal data with seasonality and allows capturing a set of standard temporal structures in time series data to forecast new data (Bianco et al., 2001). For anomaly detection purposes, a model of the sample time series is built, then the anomalies are identified by comparing the forecast with the real data. ARIMA combines an autoregression (AR) model and a moving average (MA) model, and it is integrated (I), which means it exploits the differencing technique to make stationary the non-stationary time series. Indeed, like all the regression models, also the ARIMA model can be applied to non-stationary time series only after making it stationary since trends negatively affect the model. The AR model identifies the dependent relationship between an observation and a variable number of lagged observations. This dependency is explained by the following formula:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \varepsilon_1$$

where Y_{t-1} is the first lag of the series, β_1 is its coefficient estimated by the model and α is the intercept term. On the other hand, the MA model detects the dependency between an observation and the lagged forecast errors ε_t caused by the autoregressive model, following the formula:

$$Y_t = \alpha + \varepsilon_t + \omega_1 \varepsilon_{t-1} + \omega_2 \varepsilon_{t-2} + \ldots + \omega_q \varepsilon_{t-p}$$

The model exploits three configuration parameters: p, the lag order, i.e., the number of lag observations included in the autoregressive model, d, the degree of differencing, i.e., the number of times the raw observations are differenced to achieve stationary and to remove any seasonality or trends, and q, the order of moving average, i.e., the number of lagged forecast errors for the prediction. To find the parameters

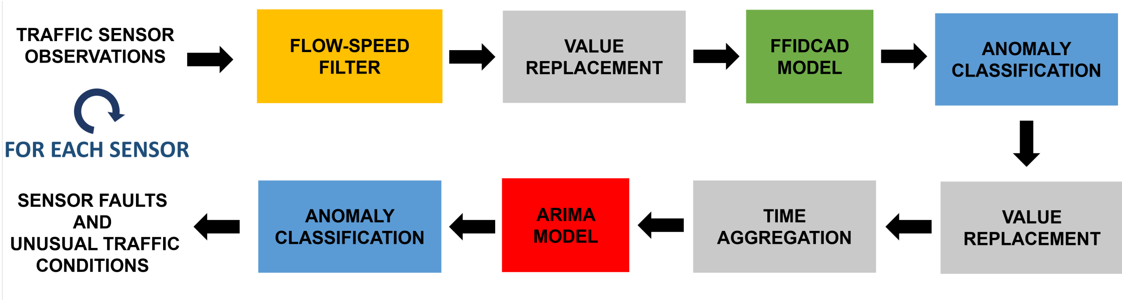


Figure 1: Overview of the data cleaning process.

which fit better the sampled data, an iterative process can be set up for building multiple models with different parameters and checking the result. The Python implementation of ARIMA in the statsmodels library allows discovering the optimal configuration parameters for each model automatically. Therefore, it is not necessary to find these values with manual tests.

3.2 Data cleaning process

The anomaly detection techniques are combined in a complex data cleaning process, as illustrated in Figure 1. Firstly, data coming from sensors are filtered through the flow-speed correlation filter. The "filtered" observations are replaced by the average of the reliable proximal observations. The resulting measurements are given as input to the FFIDCAD model; then, the model results are classified and divided into sensor faults and unusual traffic conditions considering the correlation between sensors. The obtained collection of observations, labeled with anomaly classification, is then processed to remove sensor faults, which are replaced with an average of proximal observations. Then, the obtained modified observations are aggregated with a certain time interval and given as input to the ARIMA model. Anomalies detected by ARIMA are then classified, and the final result is produced. The process is repeated for each traffic sensor.

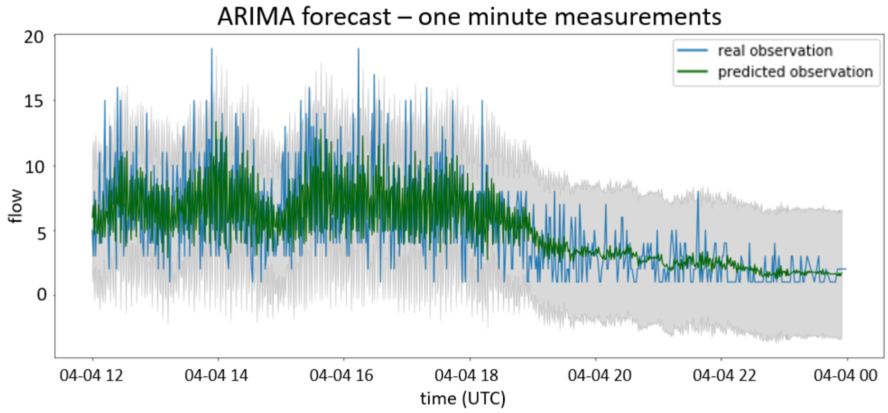
The anomaly detection techniques are employed as complementary techniques to find all the possible anomalies. We were not expecting to find a significant intersection between the anomalies found by the different techniques. However, to verify this assumption, we compare the anomalies found by the flow-speed correlation filter, and by ARIMA and FFIDCAD on a subset of real-world traffic sensor data. We noticed that the anomalies discovered by the flow-speed correlation filter were not detected by the FFIDCAD model. This happens because FFIDCAD works on the correlation between flow and speed, but it does not know the meaning of the values and the constraint of their relationship. In conclusion, the flow-speed correlation filter cannot be replaced by FFIDCAD; it is a complementary technique. Comparing the results of FFIDCAD and ARIMA, we observed that only a low percentage (less than 0.04%) of the total number of sensor faults and unusual traffic conditions detected by the two techniques have been identified by both of them. This was evidence of the complementing behavior of the methods. We conclude that all the methods have to be applied to the sensor measurements since there is not a significant overlap.

4 USE CASE

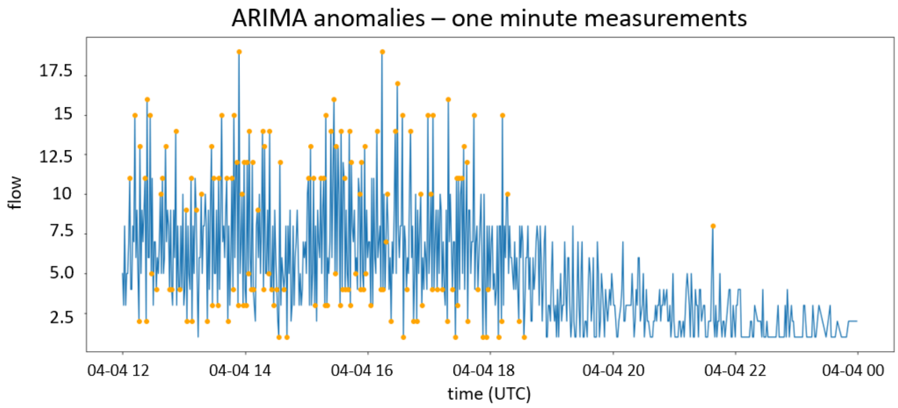
Our methodology has been applied to the road traffic sensor network in the city of Modena. Around 400 traffic sensors (induction loops) are spread in different locations of the city, in a single lane of the street, usually near traffic lights.[1] The sensors measure in real-time the number of vehicles (flow) and their average speed with a certain frequency. Sensors data are collected into a PostgreSQL database and exploited to emulate real routes of vehicles in an urban traffic model (Po et al., 2019b; Po et al., 2019a; Bachechi et al., 2020c; Bachechi et al., 2020a; Bachechi and Po, 2019). The frequency of measurement is 1 minute for sensors located in urban roads and 15 minutes for sensors in provincial area. From September 2018 till April 2022, the database collected more than 550 million traffic observations. Since traffic sensors are installed under the surface of the street, their maintenance cannot be continuously granted, and sensors can be faulty and provide erroneous information. Thus, an anomaly detection process is essential for two reasons: excluding outliers from the traffic model input and discovering unusual traffic conditions.

In the next subsections, we discuss the configuration of the data cleaning process to fit our use case.

[1] Modena Sensor Map: https://trafair.eu/modenasensormap/



(a) Time series representing the measurements of the sensor (blue line) and the prediction of ARIMA (green line).



(b) Anomalies (orange points) detected by ARIMA.

Figure 2: ARIMA applied to one minute measurements of one sensor in some days of April 2019.

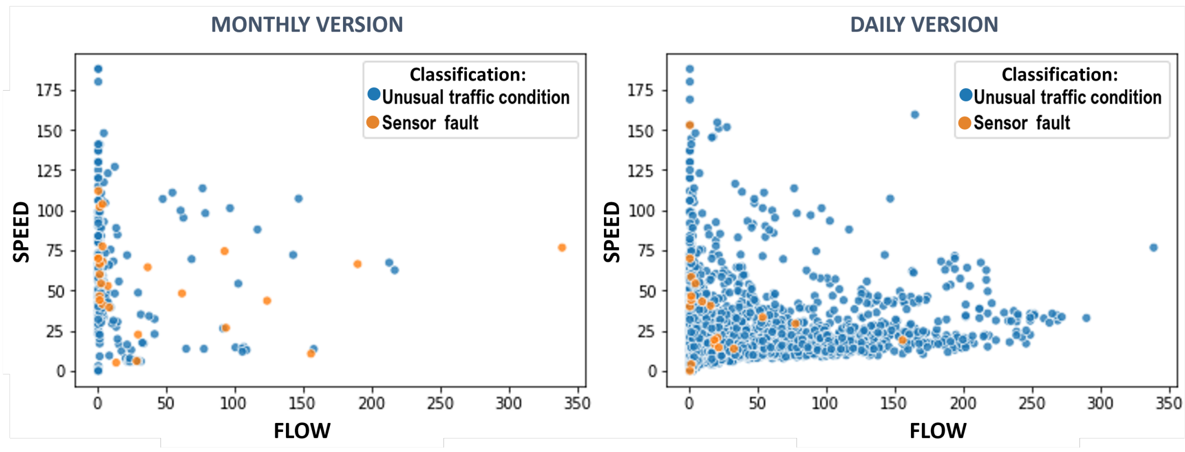


Figure 3: FFIDCAD anomaly detection over the whole month of November 2018 (monthly version) and the only 8th November 2018 (daily version).

4.1 Aggregation of the input data

The three implemented anomaly detection techniques are applied to the measurements of each sensor separately. The flow-speed correlation filter has to be applied to the measurements as they are provided by the sensors, without aggregation, since the application of the average speed could modify the result significantly. To probe this decision, we could consider two measurements, each of them related to one minute time interval, the first with *flow = 1* and *speed = 150* and the second with *flow = 50* and *speed = 60*. The first measurement is considered non-anomalous by the filter, while the second is an anomaly. The weighted average of the two values of speed is calculated by the formula:

$$weighted\_average\_speed_i = \frac{\sum_{i=1}^{n}(flow_i * speed_i)}{\sum_{i=1}^{n} flow_i}$$

where $flow_i$ and $speed_i$ are the values of flow and speed of the i^{th} measurement. In this case, the weighted average speed is around 62. The aggregated measurement, considering the weighted average, would be detected as anomalous by the filter. In conclusion, it is better to apply the filter to the "raw" observations, i.e., data not aggregated, to avoid excluding some data that are instead valid measurements.

FFIDCAD and ARIMA have been applied to the "raw" data and to the data aggregated every 15 minutes. When the data are aggregated, the values of the flow are summed up for sensors with 1 minute frequency, since they represent the actual number of vehicles in the time interval of one minute. The weighted average is evaluated to obtain a representative value of average speed in the aggregated interval. The results obtained by using different aggregation intervals were compared in order to define the best choice for each anomaly detection technique. Firstly, FFIDCAD was tested on the measurements of one day (8^{th} November 2018) with 388,800 observations. The total number of anomalies found by FFIDCAD in data aggregated every 15 minutes is 2286. Instead, the number of detected anomalies on the same, not aggregated input data, is 11358. By applying classification, the total number of sensor faults on aggregated data is 19 (0.008%), a very low number, and 77 (0.0067%) on not aggregated data. Considering another day (15^{th} April 2019), the anomalies detected aggregating data are 3618, and 61 of them are classified as sensor faults; without the aggregation of data, the total number of anomalies grew to 12,129 and the sensor faults are 414. FFIDCAD, as described in Section 3.1, studies the correlation between speed and flow. When aggregating data every 15 minutes, some anomalies cannot be detected since the relationship between flow and speed changes when evaluating the sum of the flow and the weighted average of the speed in 15 minutes. For this reason, FFIDCAD should be applied to raw input data.

We also evaluated the application of ARIMA to both the "raw" measurements and the measurements aggregated by 15 minutes. We compared the results, and we found out that the configuration, which considers the "raw" measurements, detects many anomalies. Plotting these anomalies and analyzing the values of flow and speed, they did not seem to be anomalous values. An example is provided by Figure 2a which represents the measurements of one traffic sensor on a day of April 2019 (the blue line) and the prediction of ARIMA (the green line). The gray area is delimited between the lower and upper bounds of prediction found by the model. The measurements with a flow value outside this area are detected as anomalies. Figure 2b highlights with orange points the measurements considered as anomalies by the model. As can be seen, many measures are anomalies, by the ways they seem to be non-anomalous. We suppose this erroneous behavior happens because the sensors we consider are installed near traffic lights; therefore, it is very common the flow value grows fast and then again takes on lower values. Studying the "unsteady" trend of the time series, the ARIMA model is not able to predict the one-minute measurements in the right way. Therefore, we decided to apply the ARIMA model to measurements aggregated by 15 minutes.

4.2 Time interval of concern

While the flow-speed correlation filter consider the "raw" measurements once a time, FFIDCAD and ARIMA are applied to a set of measurements related to a certain time interval. For the FFIDCAD model, this means that the model can take as input the measurements related to periods of different lengths, i.e., one day, one week, one month, and so on. Mean, standard deviation, and covariance are calculated on the entire dataset provided as input. In this way, the model finds anomalies based on the whole period. For the ARIMA model, the different duration of the time interval is related to the training set.

FFIDCAD algorithm was applied to the entire month of November 2018 and then on a single day: 8^{th} November 2018. The anomalies detected by the algorithm trained on the entire month are different from the ones detected by the same algorithm trained only on November 8^{th}. The total number of anomalies detected in the whole month was 7226 and only 281 of them on November 8^{th} (only 26 classified as sensor faults). Instead, with a daily interval of application, the detected anomalies for the only 8^{th} November were 2286 (19 sensor faults). The reduction in the number of sensor faults in the daily version is because more anomalies are detected w.r.t. the monthly version and some of these anomalies are simultaneous with the one previously erroneously classified as

Table 1: Experimental results.

	November 8th, 2018	April 15th, 2019
Available sensors	256	335
Raw measurements	383421	442802
Flow-speed filter anomalies	14076 (3.7%)	14461 (3%)
FFIDCAD anomalies	11147 (3%)	16975 (4%)
FFIDCAD sensor faults	204 (0.02%)	2207(13%)
ARIMA anomalies	1431 (18%)	2485 (8%)
ARIMA sensor faults	96 (14.9%)	263 (9.5%)

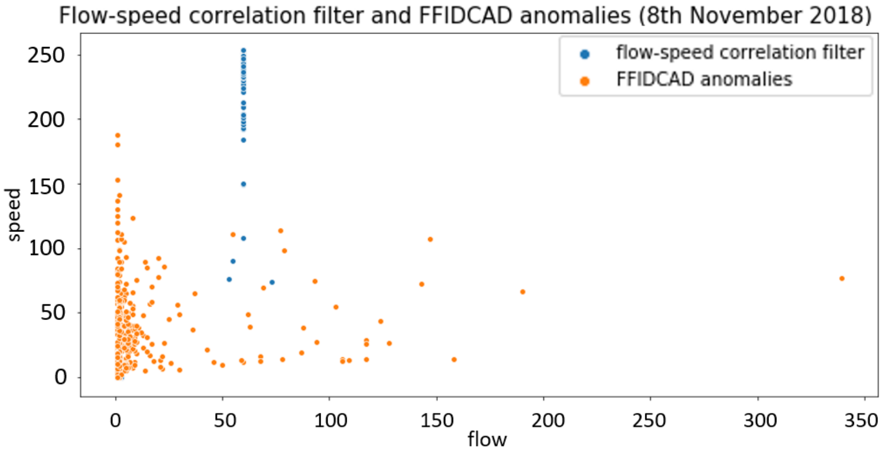


Figure 4: Anomalies found by the flow-speed correlation filter and the FFIDCAD model on the 8th November 2018.

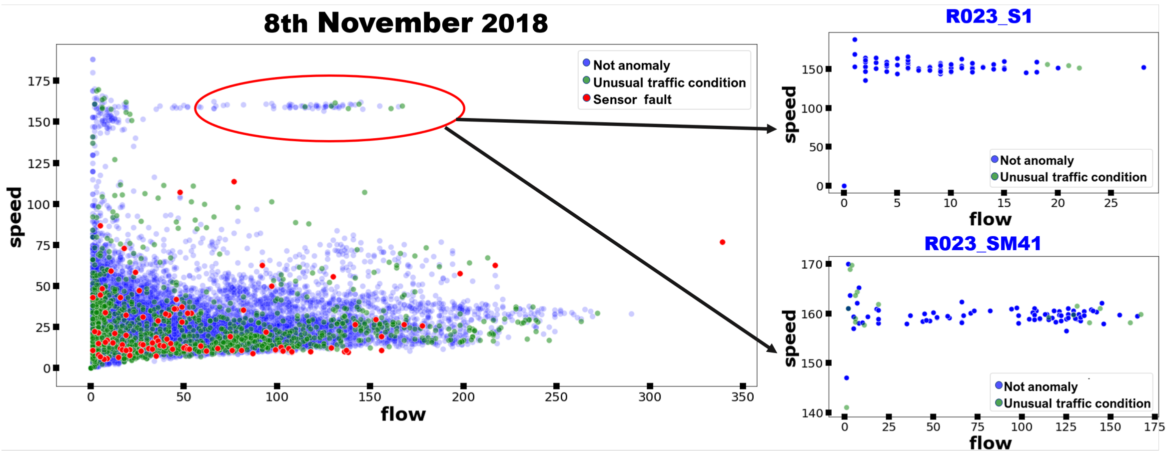


Figure 5: Distribution of detected anomalies on the 8th November 2018 observed by all traffic sensors and particulars for two specific traffic sensors (R023_S1 and R023_SM41).

sensor faults and now classified as unusual traffic conditions. The two different intervals of application detect different anomalies, only 185 anomalies were detected by both the monthly and daily version. 21 of the 26 anomalies classified as sensor fault by the monthly version are also detected by the daily version. However, only 3 sensor faults detected by the daily version are detected also by the monthly version. In Figure 3, the difference between the anomalies detected for all the sensors in the two versions are represented in a two-dimensional space considering their flow and speed. In this case, data are aggregated every 15 minutes for both monthly and daily versions. The reason why some anomalies are not detected in the monthly version is that training the model on the whole month means also considering holidays and weekends that have a singular trend and normally a lower flow and can influence the detection of anomalies in regular working days. For this reason, we choose the daily version approach for FFIDCAD.

In the ARIMA model, instead, the train is made on the entire month to predict one day. In this way, different trends can be included in the model, i.e., the daily trend, but also the weekly trend, the different behavior of the sensors in holidays, and so on. Thus, the ARIMA model is used to predict sensor observations related to one day, but the model is trained on the whole month.

4.3 Traffic accident analysis

The application of our methodology was evaluated on two specific days: 8$^{\text{th}}$ November 2018 and 15$^{\text{th}}$ April 2019. We selected these two days since there were reported road accidents in streets controlled by our sensors in Modena, so it was possible to check if our methodology can distinguish between sensor faults and unusual traffic conditions. Table 1 reports the number of available sensors, the number of observations, and the number and percentage of anomalies detected by the different steps of the data cleaning process for each of the two days.

In the first experiment (on 8$^{\text{th}}$ November 2018) the number of available sensors was lower than in the second experiment; as a consequence, fewer observations were collected. The flow-speed correlation filter detects a similar percentage of anomalies in the two days. In both experiments, the forgetting factor was set to 0.999, as suggested by the authors of (Moshtaghi et al., 2011) and an FFIDCAD model was generated for each sensor. The time required by FFIDCAD to find anomalies is less than 1 second for each sensor. Even if the percentage of detected anomalies is similar in the two experiments, the percentage of anomalies classified as sensor faults is significantly higher in the second experiment.

Figure 4 shows the anomalies found in the first experiment by the flow-speed correlation filter and the FFIDCAD model in a flow-speed scatter plot. As can be seen, most of the anomalies found by FFIDCAD are related to low values of flow, while the flow-speed correlation filter detects anomalies related to very high values of speed.

Before applying the ARIMA model we replaced the measurements filtered by the flow/speed correlation filter and the ones identified as sensor fault by FFIDCAD with the average of proximal measurements. The ARIMA model was trained on the measurements of the previous 30 days aggregated every 15 minutes to forecast the measurements of the next hour. Then, the model was retrained with the real measurements of the predicted hour to forecast the next hour and so on. We used this approach to allow anomaly detection in real-time. The model requires less than 10 minutes to predict the measurements of the whole day. The percentage of anomalies detected by the ARIMA model is halved in the second experiment even if the absolute number of anomalies is higher.

In the first experiment, the anomalies classified as sensor faults are related to 46 sensors, and the ones classified as unusual traffic conditions to 189 sensors. While, in the second experiment, the anomalies classified as sensor faults are related to 72 sensors, and the ones classified as unusual traffic conditions to 227 sensors.

All the aggregated measurements of all the available sensors on the 8$^{\text{th}}$ of November 2018 are displayed in Figure 5. It can be observed that the majority of sensor faults are detected when the speed has low values. Moreover, the values with very high speed and flow (the ones indicated by the red circle) appear to be anomalies observing the whole population of sensors. The 96% of these measures with speed higher than 150 km/h belongs to two sensors (R023_S1 and R023_SM41). It was not possible to identify their unrealistic measurements as anomalies because anomaly detection is performed individually for each sensor. Figure 7 displays the anomalies detected in the second experiment. Like in the first experiment, the majority of sensor faults are detected when the speed has low values. The values with very high speed and flow (the ones indicated by the red circle) belong again to the two sensors R023_S1 and R023_SM41. Since the majority of their observations have high speed and flow these sensors should be checked to investigate the presence of malfunctions or drifts. The data in the light-blue circle instead come from another sensor's observation, R009_SM19. This sensor was not employed during the first experiment and shows always the same value of speed for very different values of flow; this suspect behavior needs to be further analyzed. Therefore, our anomaly detection methodology fails to detect a constant drift or malfunction of the sensor that can emerge only when its observations are compared with the ones of the other sensors. On the 8$^{\text{th}}$ of November 2018, two reported car accidents happened in Modena. We tested the ability of our solution to detect accidents as unusual traffic conditions. For each anomaly classified as an unusual traffic condition, an *evidence score* has been evaluated. The evidence score is the number of one-minute observations that were classified as unusual traffic conditions by the FFIDCAD model in

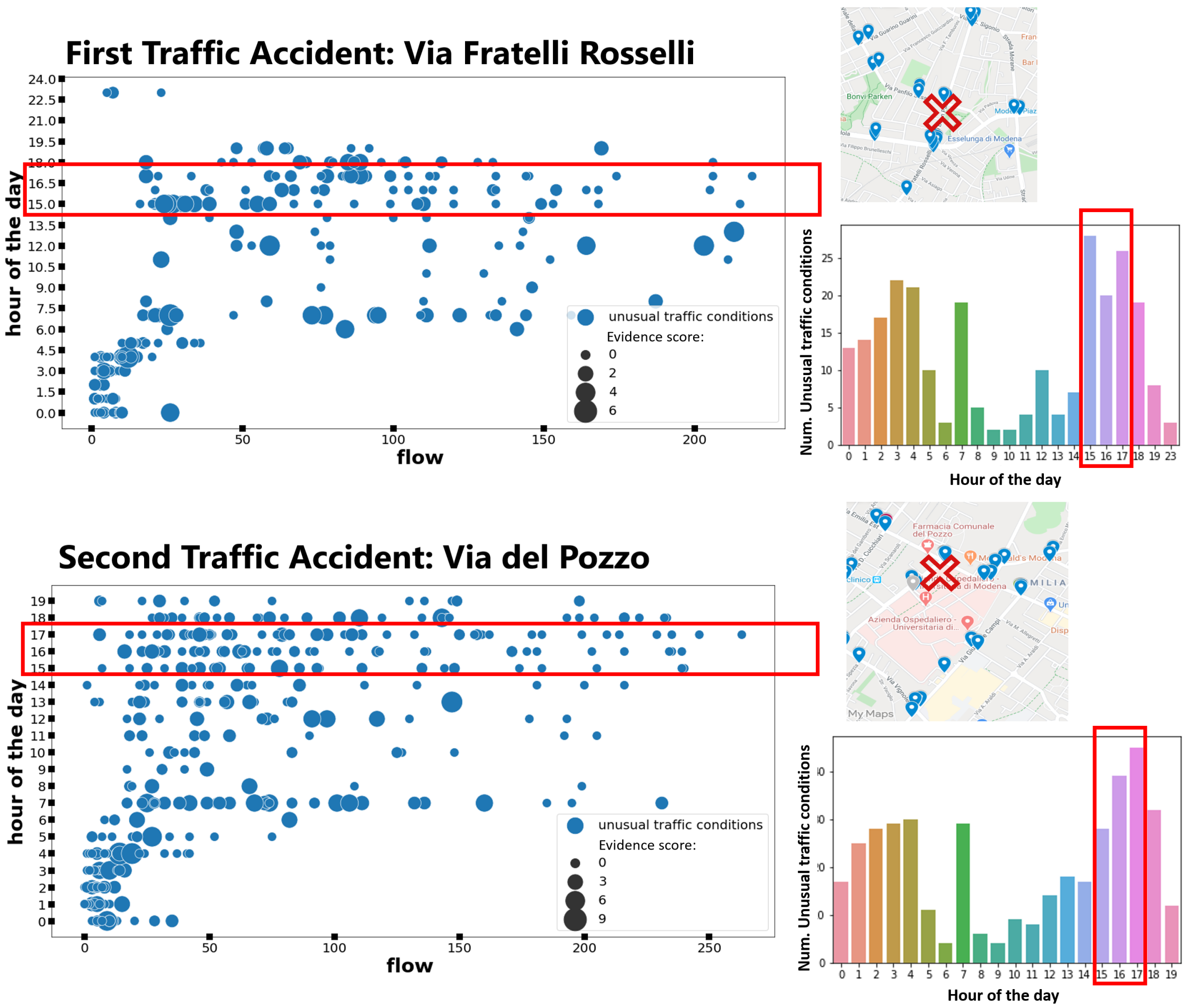


Figure 6: Distribution of the unusual traffic conditions detected on the 8th November 2018 in sensor measurements near the location of the car accident.

the aggregated time interval. In Figure 6, the unusual traffic condition anomalies observed in sensors with a distance from the accident location lower or equal to 1500 meters are displayed for each accident. The size of the spots is proportional to their evidence score. The two accidents occurred both at 03:30 PM UTC in two different areas of the city, in the time interval around that time (highlighted by red rectangles in Figure 6) we observe more unusual traffic conditions in sensors located nearby than in the other hours of that day.

On 15th April 2019, there was one reported car accident around 7 AM UTC. In Figure 8, the unusual traffic conditions observed in the sensors with a distance from the accident location lower or equal to 1500 meters are displayed. The graph shows the unusual traffic conditions with a size proportional to their evidence score evaluated considering the number of unusual traffic conditions detected by the FFIDCAD model at the same time interval. There are several unusual traffic conditions in the area around 6 AM in the moments preceding the accident. These anomalies have a high evidence score and summing the value of this score, the criticality of the traffic condition is evident.

Comparing Figure 6 to Figure 8 and also considering other plots of the same graph for a different group of sensors, it is evident that our methodology can detect slow-moving traffic that usually interests morning and mid-afternoon hours. However, the relation between slowdowns and car accidents should be further analyzed to be able to successfully identify car accidents.

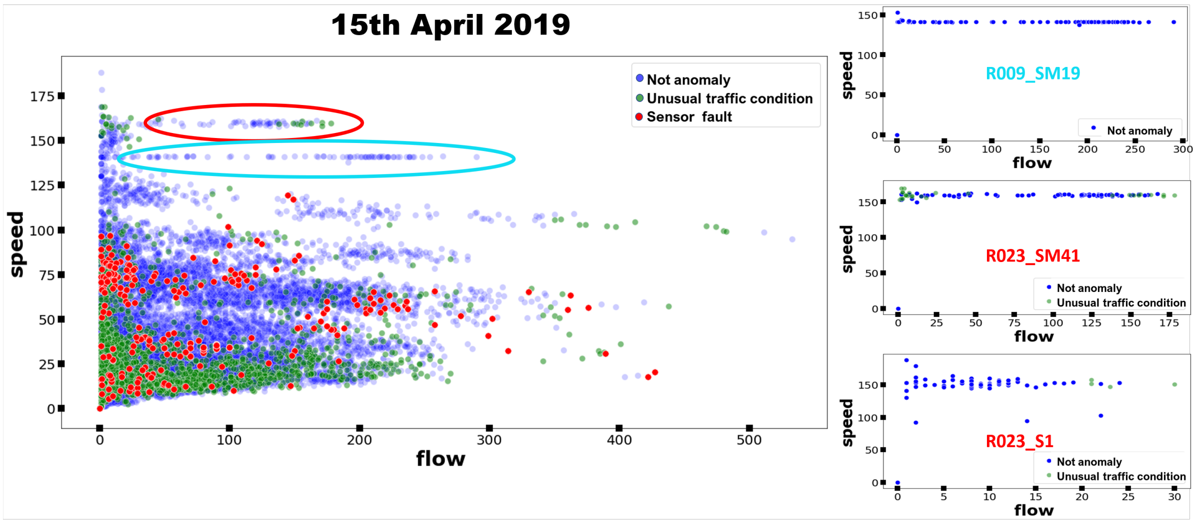


Figure 7: Distribution of detected anomalies the 15th April 2019 considering the flow (number of vehicles in 15 minutes interval) and speed (Km/h) observed by traffic sensors.

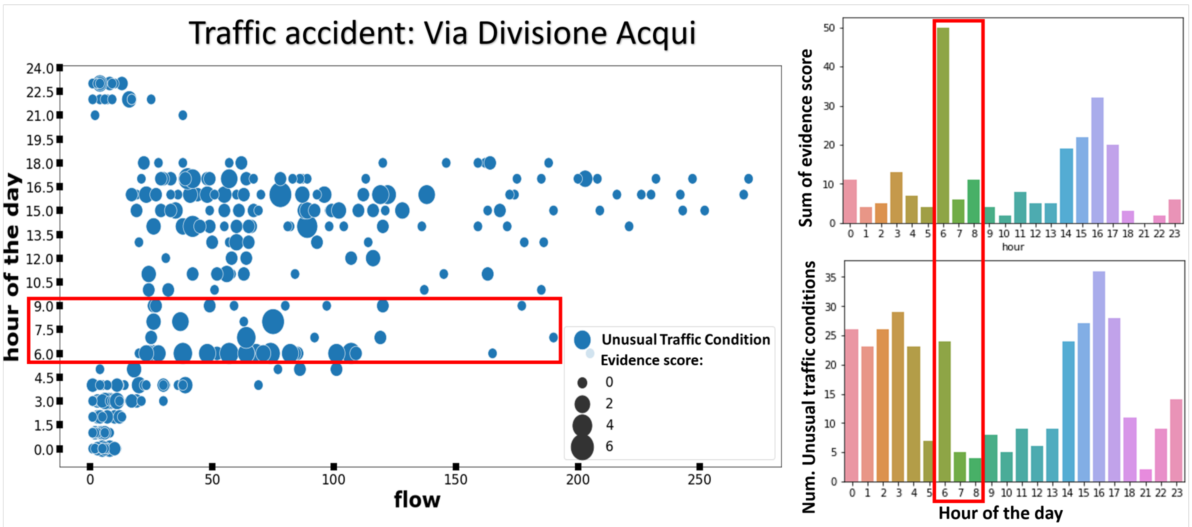


Figure 8: Distribution of the unusual traffic conditions detected on the 15th April in sensor measurements near the location of the car accident.

5 CONCLUSION AND FUTURE WORK

This paper has introduced a combined method for the detection and classification of anomalies on traffic sensors. Two anomaly detection algorithms, a filtering technique, and an anomaly classifier were combined to detect unusual traffic conditions and sensor faults. Sensor faults are observations collected from induction loop sensors that should be discarded in order to evaluate real traffic data. Unusual traffic conditions, instead, provide useful information to detect deviations from traffic trends and to identify critical situations such as car accidents. Having clean and correct traffic data is very important for the study and management of road traffic; it also ameliorates the predictions of the traffic flows that can be generated using a traffic model (Bachechi and Po, 2019; Po et al., 2019a).

In the future, we will analyze the impact, on traffic model performance, of the detection and classification of anomalies on traffic sensors. We plan to compare the current implementation of the traffic model that uses all sensor data observations with a traffic model that uses a pre-processed input without sensor faults. Moreover, we intend to implement the prediction of traffic congestion, thus exploiting multi-modal data streams that combine the IoT data, weather conditions, and social media data streams.

Detecting anomalies and sensor faults is an important aspect of a wide variety of sensors used in smart cities. The proposed anomaly detection methodology for multivariate time series is designed for traffic sensors, but can be easily adapted for different applications by modifying the flow-speed filter considering the given use case. As a future work, we aim to deepen the problem of anomalies produced by air quality monitoring sensors. Such sensors are more sensitive to environmental changes than traffic sensors, their observations are strongly affected by the values of humidity, temperature and also the concentrations of other gases or particles (Rollo and Po, 2021; Rollo et al., 2021; Bachechi et al., 2020b). Moreover, air quality sensors are subject to rapid deterioration over time.

ACKNOWLEDGEMENTS

Research reported in this paper was partially supported by the TRAFAIR project 2017-EU-IA-0167, co-financed by the Connecting Europe Facility of the European Union. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the EU Commission. The authors would like to thank in particular the partners that contribute to the collection and management of traffic sensor data: the City of Modena and Lepida S.c.p.A..

REFERENCES

Ali, K., Anwar, T., Naqvi, I. H., and Jafry, M. H. (2015). Composite event detection and identification for wsns using general hebbian algorithm. In *2015 IEEE International Conference on Communications (ICC)*, pages 6463–6468.

Bachechi, C., Desimoni, F., Po, L., and Casas, D. M. (2020a). Visual analytics for spatio-temporal air quality data. In *2020 24th International Conference Information Visualisation (IV)*, pages 460–466.

Bachechi, C., Desimoni, F., Po, L., and Casas, D. M. (2020b). Visual analytics for spatio-temporal air quality data. In Banissi, E., Khosrow-shahi, F., Ursyn, A., Bannatyne, M. W. M., Pires, J. M., Datia, N., Nazemi, K., Kovalerchuk, B., Counsell, J., Agapiou, A., Vrcelj, Z., Chau, H., Li, M., Nagy, G., Laing, R., Francese, R., Sarfraz, M., Bouali, F., Venturini, G., Trutschl, M., Cvek, U., Müller, H., Nakayama, M., Temperini, M., Mascio, T. D., Sciarrone, F., Rossano, V., Dörner, R., Caruccio, L., Vitiello, A., Huang, W., Risi, M., Erra, U., Andonie, R., Ahmad, M. A., Figueiras, A., Cuzzocrea, A., and Mabakane, M. S., editors, *24th International Conference on Information Visualisation, IV 2020, Melbourne, Australia, September 7-11, 2020*, pages 460–466. IEEE.

Bachechi, C. and Po, L. (2019). Implementing an urban dynamic traffic model. In Barnaghi, P. M., Gottlob, G., Manolopoulos, Y., Tzouramanis, T., and Vakali, A., editors, *2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019, Thessaloniki, Greece, October 14-17, 2019*, pages 312–316. ACM.

Bachechi, C., Po, L., and Rollo, F. (2022a). Big data analytics and visualization in traffic monitoring. *Big Data Res.*, 27:100292.

Bachechi, C., Rollo, F., Desimoni, F., and Po, L. (2020c). Using real sensors data to calibrate a traffic model for the city of modena. In Ahram, T. Z., Karwowski, W., Vergnano, A., Leali, F., and Taïar, R., editors, *Intelligent Human Systems Integration 2020 - Proceedings of the 3rd International Conference on Intelligent Human Systems Integration (IHSI 2020): Integrating People and Intelligent Systems, February 19-21, 2020, Modena, Italy*, volume 1131 of *Advances in Intelligent Systems and Computing*, pages 468–473. Springer.

Bachechi, C., Rollo, F., and Po, L. (2020d). Real-time data cleaning in traffic sensor networks. In *17th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2020, Antalya, Turkey, November 2-5, 2020*, pages 1–8. IEEE.

Bachechi, C., Rollo, F., and Po, L. (2022b). Detection and classification of sensor anomalies for simulating urban traffic scenarios. *Clust. Comput.*, 25(4):2793–2817.

Bachechi, C., Rollo, F., Po, L., and Quattrini, F. (2021). Anomaly detection in multivariate spatial time series: A ready-to-use implementation. In Mayo, F. J. D., Marchiori, M., and Filipe, J., editors, *Proceedings of the 17th International Conference on Web Information Systems and Technologies, WEBIST 2021, October 26-28, 2021*, pages 509–517. SCITEPRESS.

Bianco, A., Ben, M. G., Martínez, E., and Yohai, V. (2001). Outlier detection in regression models with arima errors using robust estimates. *J. Forecast.*, 20:565–579.

Chander, B. and Kumaravelan, G. (2022). Outlier detection strategies for wsns: A survey. *Journal of King Saud University - Computer and Information Sciences*, 34(8):5684–5707. Cited By :3.

Chandola, V., Mithal, V., and Kumar, V. (2008). Comparative evaluation of anomaly detection techniques for sequence data. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pages 743–748.

Desimoni, F., Ilarri, S., Po, L., Rollo, F., and Trillo-Lado, R. (2020). Semantic traffic sensor data: The trafair experience. *Applied Sciences*, 10(17).

Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2012). Toward supervised anomaly detection. *Journal of Artificial Intelligence Research (JAIR)*, 45.

Kurian, N., Thomas, A., and George, B. (2015). Automated fault diagnosis in multiple inductive loop detectors.

11th IEEE India Conference: Emerging Trends and Innovation in Technology, INDICON 2014.

Moshtaghi, M., Leckie, C., Karunasekera, S., Bezdek, J. C., Rajasegarar, S., and Palaniswami, M. (2011). Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks. In Cook, D. J., Pei, J., Wang, W., Zaïane, O. R., and Wu, X., editors, *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pages 467–476. IEEE Computer Society.

O'Reilly, C., Gluhak, A., Imran, M. A., and Rajasegarar, S. (2014). Anomaly detection in wireless sensor networks in a non-stationary environment. *IEEE Commun. Surv. Tutorials*, 16(3):1413–1432.

Po, L., Rollo, F., Bachechi, C., and Corni, A. (2019a). From sensors data to urban traffic flow analysis. In *2019 IEEE International Smart Cities Conference, ISC2 2019, Casablanca, Morocco, October 14-17, 2019*, pages 478–485. IEEE.

Po, L., Rollo, F., Viqueira, J. R. R., Lado, R. T., Bigi, A., López, J. C., Paolucci, M., and Nesi, P. (2019b). TRAFAIR: understanding traffic flow to improve air quality. In *2019 IEEE International Smart Cities Conference, ISC2 2019, Casablanca, Morocco, October 14-17, 2019*, pages 36–43. IEEE.

Ramchandran, A. and Sangaiah, A. K. (2018). Chapter 11 - unsupervised anomaly detection for high dimensional data—an exploratory analysis. In Sangaiah, A. K., Sheng, M., and Zhang, Z., editors, *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Intelligent Data-Centric Systems, pages 233 – 251. Academic Press.

Rollo, F. and Po, L. (2021). Senseboard: Sensor monitoring for air quality experts. In Costa, C. and Pitoura, E., editors, *Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, March 23, 2021*, volume 2841 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rollo, F., Sudharsan, B., Po, L., and Breslin, J. G. (2021). Air quality sensor network data acquisition, cleaning, visualization, and analytics: A real-world iot use case. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, UbiComp '21, page 67–68, New York, NY, USA. Association for Computing Machinery.

Yu, Q., Jibin, L., and Jiang, L. (2016). An improved arima-based traffic anomaly detection algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 12(1):9653230.

Zamini, M. and Hasheminejad, S. M. H. (2019). A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare. *Intell. Decis. Technol.*, 13(2):229–270.

Zare Moayedi, H. and Masnadi-Shirazi, M. A. (2008). Arima model for network traffic prediction and anomaly detection. In *2008 International Symposium on Information Technology*, volume 4, pages 1–6.

Zhang, M., Guo, J., Li, X., and Jin, R. (2020). Data-driven anomaly detection approach for time-series streaming data. *Sensors*, 20(19):5646.

Zygouras, N., Panagiotou, N., Zacheilas, N., Boutsis, I., Kalogeraki, V., Katakis, I., and Gunopulos, D. (2015). Towards detection of faulty traffic sensors in real-time. In *MUD@ICML*.