Scientia Iranica D (2023) 30(2), 465–476

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

Anomaly Detection Fog (ADF): A federated approach for internet of things

M. Behniafar[a], A. Mahjur[a,*], and A. Nowroozi[b]

a. *Faculty of Electrical and Computer Engineering, Malek Ashtar University of Technology, Tehran, Iran.*
b. *Department of Media Engineering, IRIB University, Tehran, Iran.*

Received 20 February 2021; received in revised form 12 January 2022; accepted 21 November 2022

KEYWORDS
Internet of Things (IoT);
Anomaly;
Fog;
IT security;
Intrusion detection.

Abstract. Heterogeneous data models and resource constraints are the challenging issues of anomaly detection in Internet of Things (IoT). Due to these issues and the complexity of conventional anomaly detection methods, it is necessary to design an anomaly detection approach with IoT-specific concerns. This paper presents a framework for anomaly detection specially designed for IoT called Anomaly Detection Fog (ADF). ADF uses network slicing to present a federation of heterogeneous fog clusters. Federated fog clusters collaborate with each other via anomaly directives (heterogeneous context abstracts) for context-aware and application-independent anomaly detection. Evaluations show that ADF enjoys higher detection accuracy by detecting 95% of false alarms in comparison to conventional anomaly detection methods. ADF reduces energy consumption by 40%. Moreover, it reduces communication overhead and detection latency by preventing cloud offloading.

1. Introduction

Despite using cryptographic and authentication mechanisms, Internet of Things (IoT) is still vulnerable to malicious attacks due to weak security mechanisms in constraint things [1]. The significant security vulnerability of IoT is the physical intrusion in the network [2–4]. It is done by node capturing to disrupt perception procedures and to modify measured values. The intruder captures encryption and authentication keys to follow the security protocols and makes itself a legitimate node. The main target of intruders is to breach data integrity by manipulation of the network data or injection of malicious data to make a deviation in the aggregated data. To detect such data attacks, it is necessary to detect anomalies in the aggregated data of the network.

Conventional anomaly detection solutions are not applicable to IoT due to their resource consumption or computational complexity [5,6]. The inherent features of IoT including resources constraint as well as heterogeneity of things and contexts are the major challenges in detecting data anomalies in IoT [7–10].

The research on anomaly detection in IoT can be investigated from two viewpoints [11]. From the first point of view, the anomaly detection approaches are classified based on the targeted attack network layer. They are divided into two groups. The first group detects attacks in the transport or network layer (for example, routing attacks). These works are out of this article's scope. The second group detects anomalies in the application layer. They have two issues. Most of them are developed for detecting anomalies in a particular application. They cannot be used in a network consisting of heterogeneous contexts and applications. Another issue is the complexity, communication, and computation overhead of them.

*. *Corresponding author.*
E-mail addresses: mbehniafar@mut.ac.ir (M. Behniafar); mahjur@mut.ac.ir (A. Mahjur); alirezanowroozi@iribu.ac.ir (A. Nowroozi)

doi: 10.24200/sci.2022.57774.5410

From another point of view, an anomaly detection solution in IoT can be done in the front end things or in the cloud. Anomaly detection in front-end things is not suitable due to resource constraints in terms of computation and energy. Using the cloud computing approach is more common in IoT, but it has challenges that can be mentioned as follows. Acquisition of the limited bandwidth of IoT communication channels, processing delays, and security challenges are some of the challenges of this approach [12,13]. The issues are more challenging in large-scale geographically distributed networks with low latency requirements [14]. This is due to the cloud offloading delay.

Fog computing was introduced to meet the challenges in latency-sensitive applications and large-scale distributed networks [15–17]. Fog computing provides more power resources, stable network communications, and computing resources to data processing procedures. Fog computing reduces energy consumption by reducing the use of communication links [18]. Fog uses smart routers and gateways for data processing near the edge of the local network. In this method, constraint things gather data from the environment, do pre-processing, and send it to fog for advanced processes (fog offloading) [13,19,20].

The advantages of fog computing over cloud computing encourage the use of fog computing to provide quick analytics and precise IoT anomaly detection. Fog computing can provide context-aware data anomaly detection with minimal latency and communication overhead locally and in a context-aware manner.

Anomaly Detection Fog (ADF) is a fog-based framework for anomaly detection in IoT. ADF slices the network into clusters by the context and application. Each cluster detects data anomalies contextually based on fog computing. ADF architecture is based on edge layer, local fog layer, and core fog layer. The edge layer uses fog offloading to notify local fog for anomaly alerts. The local fog layer provides context-free anomaly directives (introduced by ADF) based on anomalies abstraction and metadata. Anomaly directives are sent to the core fog layer to assist other clusters for better anomaly detection. Finally, anomaly detection in the destination cluster is performed based on the received directives. ADF proposed heterogeneous clusters federation based on anomaly directives for data anomaly detection. The proposed framework deals with data anomaly detection in a heterogeneous and resource-constrained IoT. It presents a framework for context-aware and application-independent data anomaly detection in IoT. The main contributions of ADF are:

- Providing a lightweight framework for detecting data anomalies in a heterogeneous and resource-constrained IoT;
- Reduction of false alarms and increase of accuracy in detecting data anomalies in IoT;
- Providing a cooperative approach to anomaly detection using fog federation;
- Introduction of anomaly directives in order to detect anomalies in the heterogeneous data that are application-independent and context-aware.

This paper is organized as follows. The Section 2 reviews the literature on anomaly detection in IoT. The Section 3 presents the ADF. Section 4 presents the evaluation done by real data anomaly detection, and Section 5 gives the conclusion.

2. Related work

The literature on anomaly detection in IoT has mainly been conducted in two categories [11]. The works of the first category focus on anomaly detection against transport layer related attacks including routing attacks [21–27]. Examples of routing attacks are sinkhole, wormhole, and selective forwarding attacks. These works investigate RPL-based attacks in 6LoWPAN-based networks and attacks in transport and network layers. They generally investigate the neighbor node's connections, communications, routing tables and network behaviour in the lower layers.

The works of the second category focus on anomaly detection against application layer related attacks. This category is divided into two classes: service disruption anomaly detection and data integrity attacks anomaly detection.

Service disruption attacks target the availability and functionality of services. Denial of Service (DoS) and Distributed DoS (DDoS) are the most well-known examples of these attacks in IoT. The literature in this scope monitors parameters of network services traffic flow. If any of the feature values are changed, the deviation from normal behavior is detected.

Data integrity attacks prompt the application layer to destruct, forge, modify, or deviate the application data interactions such as aggregation data. Intruder nodes can breach physical security and weak cryptographic security mechanisms in things to compromise a node and register the adversary node as a legitimate node. Performing insider attacks by node capturing is one of the topmost security issues of IoT. In this case, insider intruders can target data integrity without taking cryptography into account (due to authentication and encryption keys acquisition). Man in the Middle attacks, replay attacks, and injection attacks are some of examples that target perception layer raw data by injecting misinformation to deviate the objective function of the application. According to this, it is very difficult to detect malicious data hidden in the perception layer raw data carried by a legitimate node. The scope of ADF (dash boxes in Figure 1) is

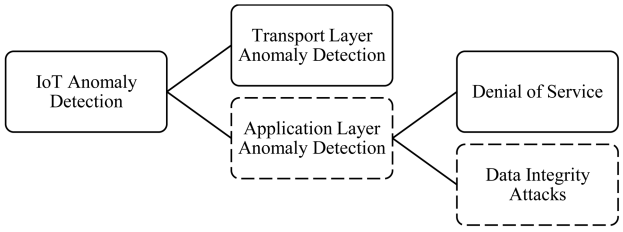

Figure 1. Anomaly detection in IoT.

to present a framework for raw data anomaly detection specifically for heterogeneous constraint IoT.

In the scope of application layer data anomaly detection, some of the researches detect anomalies in industrial data [28–30] and Industrial IoT [31]. In this class, a real application of IoT like smart homes [32,33], or an application of an IoT healthcare system [34] is considered. These works proposed a special-purpose solution for each context and application according to the applied data. ADF presents a context-free approach to data anomaly detection in a heterogeneous IoT.

The proposed approaches in the literature used different algorithms for detecting abnormal or outlier data from a series of perception layer raw data sets [11]. Statistical methods are the most common approaches to detecting the normal behavior of data at the application layer [24,35–45]. In these approaches, a statistical model of the network behavior from the dataflow of the network interactions is comprised implicitly or explicitly as a reference profile [46]. In order to detect anomalies, the obtained data from network behavior is compared with the reference profile and a degree of anomaly or a label is assigned to it according to the amount of deviation. Presenting an anomaly label is done through comparison with a threshold where determining a threshold according to conditions is challenging. Some other studies use artificial immune system [47–49], Support Vector Machine (SVM) classifiers [50,51], neural network [52–54], and PCA-based approaches [55–57] for network data anomaly detection. ADF presents a framework and its architecture for data anomaly detection in IoT. It is independent of the exact algorithm that suffers abnormal computation for raw data.

These works generally do not take into account the heterogeneity and resource constraint features of IoT nodes. Some of them provide a general model without a robust practical evaluation in the context of IoT networks. Some of them provide a special-purpose solution for a special application. ADF is proposed to meet the above-mentioned issues in the defined scope.

3. Anomaly Detection Fog (ADF)

ADF proposed network slicing into homogeneous clusters of things due to the multi-context feature of IoT. Network slicing is mainly business-driven and addresses the contextual requirements of services. ADF proposes a hierarchical fog-based anomaly detection approach (Figure 2(a)). Based on this architecture, different fog nodes collaborate to jointly support the anomaly detection framework. ADF presents this approach due to the similarity in anomalies and due to contexts cross effects.

In ADF, the fog layer acts as a proxy for the resource-constrained edge-layer devices for anomaly detection. The edge layer performs the initial processes for anomaly detection and delivers the anomaly meta information to the local fog layer for further anomaly analysis (Figure 2(b)). Due to the processing power, communication capabilities, and energy resources, the fog layer analyzes the behavior of nearby device by using local and contextual information to detect anomalies on time. We describe ADF in detail below.

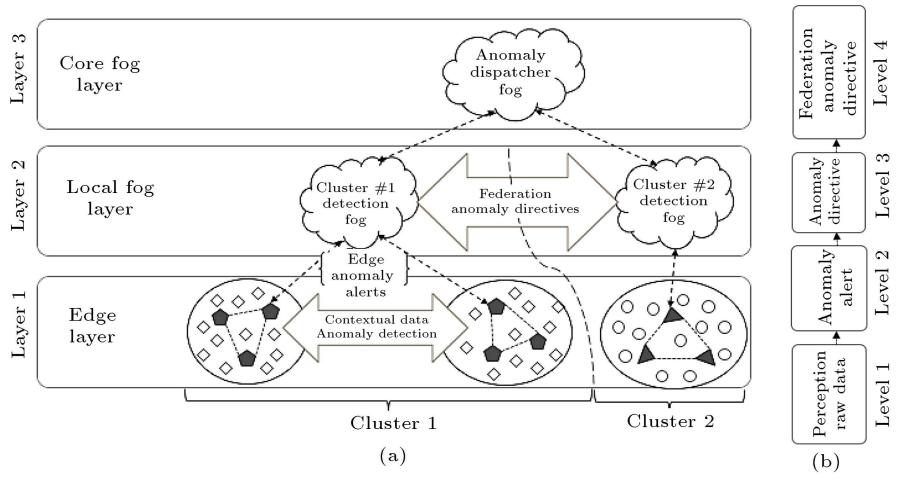

Figure 2. (a) Anomaly detection in IoT and (b) ADF dataflow.

3.1. ADF architecture

By presenting federation of network clusters via anomaly directives, ADF shares the cluster's local anomalies in metadata with other clusters. In the presented architecture, the fog layer is responsible for sharing anomaly directives among clusters. If a fog layer detects an anomaly in the local edge device data (Type A), an alert including an anomaly detection directive is sent to the dispatcher fog. Dispatcher fog sends the anomaly directive effectively to another type (Type B) of fog layer. The purpose of anomaly directive dispatching is to facilitate anomaly detection in the destination cluster. The Type-B fog layer integrates the received anomaly directives and the local anomaly information of clustered things. Finally, according to the gathered information, it takes the final decision on the state of data anomalies. We illustrate the operational procedure of ADF using the sequence diagram in Figure 3 and Figure 2(a). Figure 4 shows the class diagram of ADF to better illustrate the technical aspect of ADF. It consists of 3 main classes that that are equivalent to the layers of the architecture described below.

Things/edge layer consists of front-end edge things that are responsible for lightweight deep packet inspection, raw data aggregation, and data preparation for anomaly detection. Also, simple thresholds are applied to the data to detect outliers and send suspicious anomaly alerts to the upper layer. If any of the edge nodes detects abnormal data, it initiates a process for investigating the suspected data by raising an anomaly alert (1-a#1:n). Configuration of simple outlier detection parameters is done by the fog layer based on federated anomaly directives (1-b). Due to the communication and energy efficiency, the edge layer merely sends raw data on demand or in case of suspicious events to the fog layer. Therefore, the preparation of the requested fog raw (1-c-1) data is the responsibility of the edge layer. The forwarded packet contains the suspected event properties, anomaly properties, and the raw data (1-c-2).

The local fog layer consists of sink and gateway nodes of each cluster. It monitors the behavior of things and performs data analysis according to the cluster's context and its data model. This layer is responsible for the local cluster anomaly alerts aggregation (1-a#1:n) and detection. Time series data anomaly detection (2) is done in this layer by the fog nodes. The selection of an algorithm for the time series anomaly detection method is out of ADF scope and any

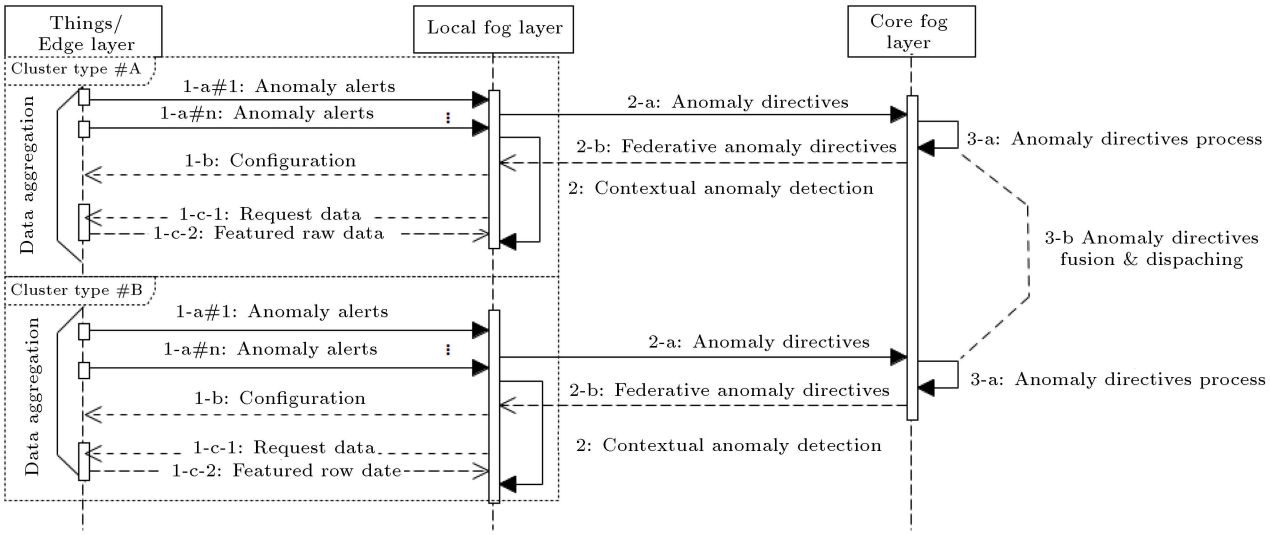

Figure 3. ADF sequence diagram.

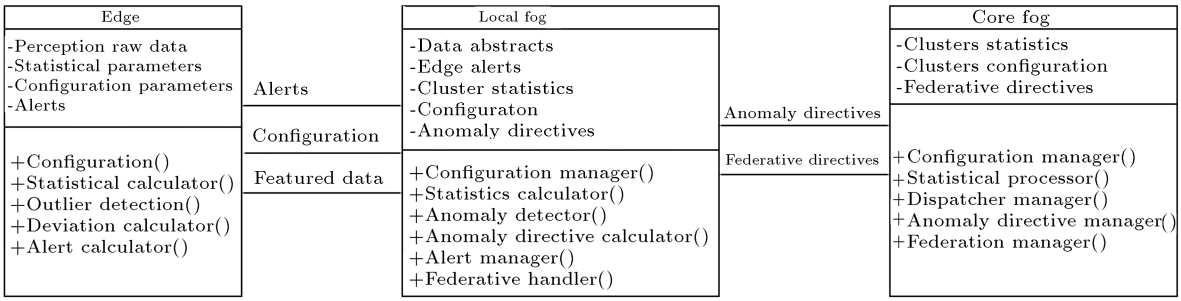

Figure 4. ADF class diagram.

desired algorithms can be used with ADF. For example, stochastic outlier detection algorithms can be used. The fog layer aggregates the diagnostic data related to abnormal alerts from the edge layer (1-c-1). After that, the required analysis is performed to confirm edge layer anomaly alerts. Moreover, this layer is responsible for producing context-free anomaly metadata called anomaly directives. Anomaly metadata is extracted from selected alerts and anomaly directives generated by the fog layer (2-a). Anomaly directives are sent to the upper fog layer for fusion with other clusters' anomaly directives. Applying federated anomaly directives received from the top layer (2-b) and adjusting anomaly detection parameters (1-b) are other tasks of this layer for dynamic anomaly detection.

Layer 3 is the IoT network core fog and the root of the IoT clusters. This layer that we named anomaly dispatcher fog is responsible for managing the cross-cluster federation (3-b). In this layer, the central fog aggregates anomaly directives from the clusters (2-a). This layer determines how anomaly directives are exchanged between clusters based on cross effects of multi-context clusters (3-a). After that, federated anomaly directives are sent to clusters in order to configure anomaly detection parameters contextually (2-b). Also, dispatcher fog determines the effectiveness degree of anomaly directive in each cluster with respect to other clusters.

3.2. ADF dataflow

The dataflow of ADF is shown in Figure 2(b). In the dataflow, environmental raw data are collected from resource-constrained edge devices with a lightweight process. Locally processed data are sent to more powerful fog nodes for further processing. The features to be investigated in ADF are categorized into two categories of data and information features, as described below.

At level 1 of the dataflow, data features are obtained by raw data deep packet inspection to detect malicious and misinformation data hidden in the interaction traffic flow. Intra-cluster communication between things and all inbound and outbound network flows are gathered. The aggregation of the environmental perception data is done by the edge layer with timestamps, suitable for time series analysis.

At level 2 of the dataflow, information features are derived from context-aware anomaly detection from raw data analysis. The results of the analysis of point, cumulative, and contextual data anomalies are shared at the edge layer. This information is reported in the form of an anomaly degree alert in the local cluster of things to improve contextual detection. The local data anomaly analysis is considered as a computational feature to direct the detection and enhance its accuracy in the rest of the network. The set of data items in an anomaly alert is presented in Table 1. Expected deviation represents the permitted interval of the measured value. The alert degree represents the percentage of anomaly certainty.

At level 3 of the dataflow, following the fusion of the heterogeneous anomaly alerts information, meta alerts (which we call anomaly directives) are generated. At this level, the abstraction and generalization of anomaly alerts are performed. Anomaly directives are the output of this process consisting of context-free and application-independent anomalies metadata for federated collaboration among clusters. An anomaly directive packet contains data items to present metadata of anomalous events (Table 2). The sensitivity represents the percentage of the strictness of anomaly detection procedures. Standard Deviation (SD) [58] and Coefficient of Variation (CV) [59] are two statistical dispersion measurement methods. The working status is about the execution mode of anomaly detection in the cluster. The values of this item can be normal, strict, and permissive.

At level 4 of the dataflow, federated anomaly directives are created (Table 3). They are based on the aggregation and fusion of anomaly directives received from heterogeneous clusters. This packet is presented for anomaly detection configuration in clusters. Sensitivity configures the level of strictness of anomaly detection in the destination cluster. Time delay indicates the desired time to apply the configura-

Table 1. Anomaly alert.

Sensor ID	Raw value	Mean value	Expected/permissible deviation	Alert degree

Table 2. Anomaly directive.

Sensor ID	Sink ID	Deviation	Sensitivity	Anomaly's event time
Permissible standard deviation	Coefficient of variation	Anomaly degree	Working status	

Table 3. Federated anomaly directive.

Source cluster	Sensitivity configuration	**Time delay**	Permissible coefficient of variation	Working mode

tion in the cluster anomaly detection. Working mode indicates the execution mode of anomaly detection in the cluster, as described in Subsection 3.3.

3.3. ADF working modes

ADF works in two modes: strict and permissive modes. We will study smart IoT in cities with different weather and traffic sensors as an example to explain ADF working modes. When weather conditions suddenly become abnormal, ADF provides a weather anomaly directive. A sudden change to the weather condition parameters has a direct impact on the traffic situation of streets. In the strict mode, ADF dynamically reduces the sensitivity of the traffic data anomaly detection algorithm to prevent false-positive errors. Depending on the application, anomaly detection directives can be useful in the permissive mode. In the permissive mode, the anomaly detection algorithm runs with a lenient configuration. In this mode, ADF uses anomaly directives to dynamically increase the detection sensitivity for increasing the detection rate and reducing the false-negative error.

4. Implementation and evaluation

For ADF evaluation, a heterogeneous IoT network must be selected. For this purpose, two case studies have been selected as a network of heterogeneous nodes with different contexts: smart city and smart office. In the case studies, different environment features are measured and aggregated. Based on the aggregated data, decisions are made for various services. Therefore, data integrity plays a vital role in services and applications. As explained in the third section, data attacks target data integrity by modification of the perception layer data. The purpose of ADF is to detect data anomalies in order to prevent deviations in aggregated values.

We evaluate our approach against different types of data integrity attacks of different severity. In order to attack data integrity, an intruder may capture the victim node and introduce itself as a legitimate node or it may tamper environmental sensing for wrong measurement. Also, it may sniff or spoof data in the middle of the communication channel. In all attack scenarios, the adversary manipulates data and injects false data into the network. The injection can be as either a single or cumulative point. In the evaluation, the following types of data integrity attacks have been considered:

1. Blind attack: The adversary injects random false data into the network to prevent data aggregation convergence;
2. Constant data attack: The adversary node provides fixed values by forging a message and replaying it;
3. Contextual attack: The adversary node injects a legal value in the acceptable range, but it is wrong according to the context.

As demonstrated by the evaluation, the application of ADF to the data anomaly detection algorithm achieves higher detection accuracy and lower false error than the use of the algorithm alone. In terms of overhead, ADF reduces energy consumption, communication overhead, and detection latency by preventing cloud offloading. The reduction in energy consumption is about 40% based on the analysis performed by Misra and Sarkar [18].

In the implementation, Modified Stochastic Approximation (MSA) algorithm with a combination of sliding window [36] was used for time series anomaly detection. This approach allows estimating integral properties of the stochastic process of the perception raw data (by a sliding window) and tracing the variant dynamics of its stochastic behavior (by MSA). This algorithm is independent of the data distribution model and is appropriate for application-independent anomaly detection. These algorithms are suitable for the point, cumulative, contextual analyses [60] and different types of time series data anomalies. Another reason behind the selection of this algorithm is the low execution overhead for constrained edge devices. ADF is independent of the exact algorithm used to detect data anomalies; hence, it is possible to replace the abovementioned method with other methods. MSA is used as an example for evaluation to prove the efficiency of ADF. The implemented algorithm has the following steps:

0. Input stochastic process parameters (μ, σ, CV)
   $\mu = Mean\,(x)$
   SD or $\sigma = \sqrt{\sigma^2} = \sqrt{Var\,(x)}$
   $CV = \sigma/\mu$
1. Define α and s $(0 \leq \alpha \leq 1)$
   α = Mean Change Ratio
   s = Permissible Deviation Threshold
2. Fill sliding window with values, $x = [v_i . v_{i+n}]$
3. Shift the sliding window, $x_i = x_{i+1}$
4. Update MSA procedure. $\mu_i = \mu_{i-1} + \alpha * (x_i - \mu_{i-1})$; $\mu_i = \bar{x}$ = Sample mean (x)
5. Compute the deviation value, $s = |x_i - \mu_i|$
6. Assess the outlier alert degree
7. If $i < N$, then $i = i\_ + 1$ and go to step 2; else, end.

4.1. Smart city evaluation

To evaluate the result of applying ADF, we used datasets from CityPulse project [61]. This project contains the measurement of various features of Aarhus city, Denmark. CityPulse datasets contain traffic

situations and weather conditions in the city. A large volume of data is obtained from sensors embedded in various urban locations, among which a location was randomly selected and the data collected during a month with a one-hour sampling interval was extracted. Measured data features of the weather condition are wind speed, air temperature, humidity, and dew point. The measured data of traffic situation contain the count of passing cars and their average speed. In the performed evaluation process, the mentioned IoT is sliced into several clusters with homogenous things that contain different types of weather and traffic sensors.

The evaluation process was done through the federation of urban heterogeneous IoT clusters. Federation was based on anomaly directives derived from weather data to facilitate traffic data anomaly detection. The evaluations are done for two parameters of the car's speed and count of cars in the traffic. The applied method is to dynamically change the parameters of the traffic anomaly detection algorithm when a weather data anomaly alert is generated.

We have run MSA algorithm with strict and permissive configurations. In the permissive mode, the detection rate was reduced while the false-negative rate increased. In the strict mode, the false-positive rate increased. Thus, implementation of each MSA configuration mode is subject to shortcomings and deficiencies. ADF anomaly detection with the same algorithm configuration demonstrated that some of the detected anomalies in MSA included false errors. In ADF, through context-aware anomaly directives, we can detect false-positive and false-negative rates to reduce false error while increasing the true detection rate.

After using MSA for anomaly detection, ADF is applied for comparing detection accuracy. The results are presented in Tables 4 and 5. For this purpose, two scenarios were implemented and evaluated. First, we set the parameters of anomaly detection strictly. In this regard, the weight of the MSA algorithm for the mean change ratio was $\alpha = 0.1$. The permissible deviation threshold(s) was considered low (different context varied) (Table 6-Strict Mode). With these configurations, the count of detected anomalies was quite high, such that some might turn out false positive. Following the application of ADF, with an anomaly detected in the weather condition, an anomaly directive was produced and sent to the upper layers to be dispatched to traffic clusters. When traffic clusters receive the permissive anomaly directive, the values of α and permissible deviation threshold are doubled temporarily. In this scenario, by federated use of heterogeneous anomaly directives, the detected false anomalies are removed and, in turn, detection accuracy increases.

Table 4. Results of applying the proposed approach for detecting anomalies of the traffic speed data.

Anomaly detection method	Anomaly counts without ADF	Anomaly counts with ADF
Strict	71	49
Permissive	15	23

Table 5. Results of applying the proposed approach for detecting anomalies of the traffic speed data.

Anomaly detection method	Anomaly counts without ADF	Anomaly counts with ADF
Strict	76	69
Permissive	16	20

In the second scenario, through the adjustment of anomaly detection parameters with greater values for ($\alpha = 0.15$) and admissible variation threshold (different contexts) (Table 6, Permissive Mode), major anomalies are detected and false-negative error increases. Following the application of ADF, when an anomaly is detected in the weather condition, an anomaly directive is produced and sent to upper layers for dispatching to traffic clusters. When traffic clusters receive strict anomaly directives, the values of α and permissible deviation threshold(s) are set to half temporarily to detect anomalies more accurately in the anomalous timestamp. In this scenario, by federated use of heterogeneous anomaly directives, detection accuracy is enhanced.

The results obtained from the application of ADF demonstrate that the fusion of weather anomalies with traffic data has a more significant effect on the accuracy of detecting car-speed data anomalies. Moreover, as predicted, the accuracy of the detection was increased and false-positive errors were reduced in the strict mode. In the permissive mode, the detection accuracy was increased; in turn, false negative errors were reduced and the true positive rate increased.

Validation of the results was performed by comparing the results with the Azure Anomaly Detection Service [62]. We have run azure anomaly detection on the same data with strict mode and, therefore, with permissive configuration to detect anomalies. In the case of traffic speed, our approach states that 22 data points of 71 anomalous points are false-positive errors. Azure experiments confirm the false-positive error for 21 out of the 22 data points. Accordingly, we have achieved 95% accuracy in detecting false-positive errors. ADF managed to reduce the false errors in comparison to the simple conventional anomaly detection approach (Table 7).

Table 6. Time series anomaly detection algorithm configuration.

Working mode		α	s
Strict mode	Strict	0.1	D
	ADF permissive directive	$2 * 0.1$	$2 * D$
Permissive mode	Permissive	0.15	D
	ADF strict directive	$0.5 * 0.15$	$0.5 * D$

Table 7. Improvement of ADF versus conventional anomaly detection.

	Traffic speed	Traffic count
Strict mode (false positive error)	31%	9.2%
Permissive mode (false negative error)	34.8%	20%

4.2. Smart office

We evaluate ADF with another dataset from Intel Berkeley research lab [63]. This dataset contains environmental data collected from 54 sensors deployed at the lab. Measured data contains humidity, temperature, light, and voltage values measured once every 31 seconds for 40 days. This dataset includes about 2.3 million measurements collected from the sensors. For evaluation, multiple sensors were selected and the collected data at an approximate rate of 3 samples per hour was extracted.

The gathered data were evaluated in the presence of the above-mentioned anomalies and attacks (Blind, Constant, and Contextual). Anomalous data were injected into point and cumulative forms or manipulated with the sensory data.

ADF sliced the network into clusters by the sensed feature into voltage of the sensor and reported temperature values by the sensor. According to [63], there is a correlation between sensor voltage and temperature variations. Therefore, ADF proposes the federation of temperature cluster and voltage cluster for anomaly detection using anomaly directives in the voltage cluster to direct temperature cluster anomaly detection. In other words, if there is an anomaly in the sensor voltage, the temperature reported by it is not reliable. Therefore, a proportional abnormal value is predicted in the reported temperature. Therefore, ADF detects false alarms and improves the true detection rate.

The details of the anomaly detection and the evaluation method are similar to the procedures performed for the previous dataset. In this regard, first, anomalies in voltage values are detected. Anomaly directives are extracted from them. Based on the federated anomaly directives, anomaly detection is performed on the temperature values. Moreover, in order to evaluate the accuracy of ADF, anomaly detection for temperature values was performed using a conventional simple anomaly detection algorithm (MSA).

To compare the detected anomalies with and without ADF, the data anomalies of two sensors are shown in different parts of Figures 5 and 6. As shown in Figures 5 and 6, ADF removed false alarms and improved detection accuracy using federated anomaly

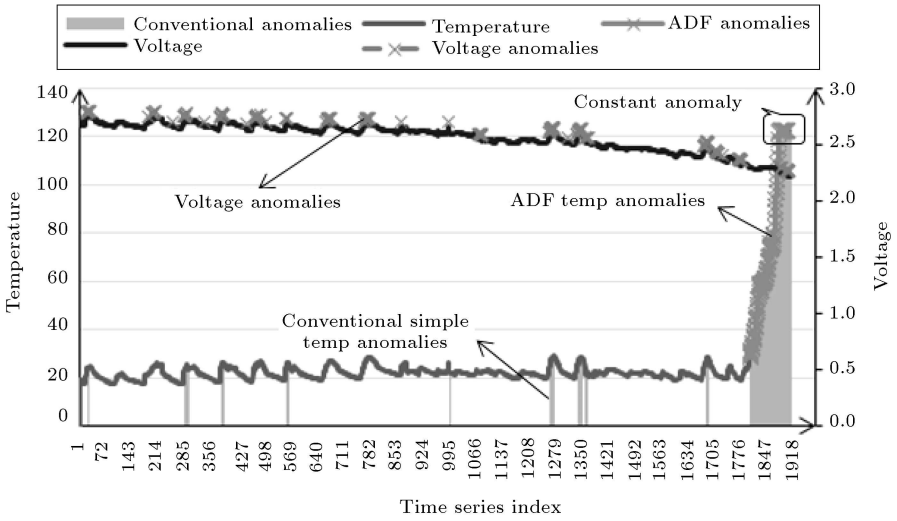

Figure 5. Sensor#1 anomaly detection.

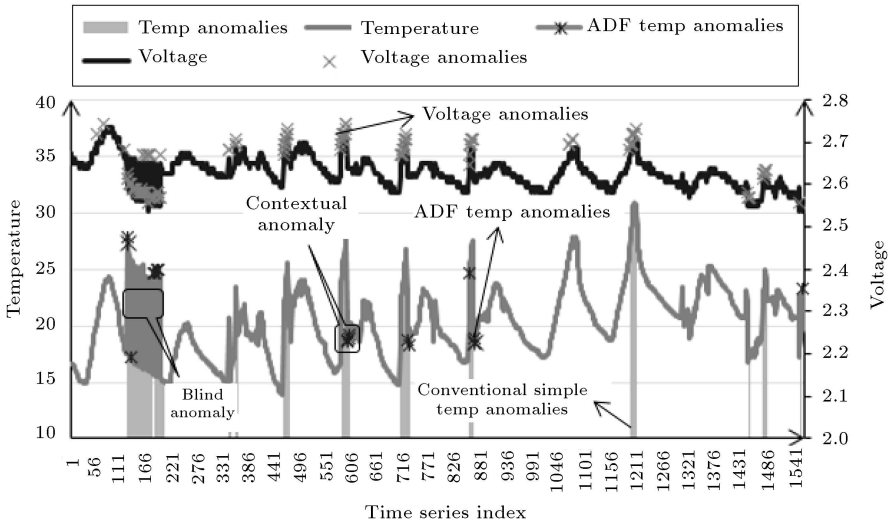

Figure 6. Sensor#2 anomaly detection.

directives. As shown in Figure 5, at the end of the temperature measurement interval, the reported values are subject to constant anomalies detected by ADF. The other sensor data shown in Figure 6 include constant and blind anomalies that have been detected by ADF.

To evaluate smart office data anomaly detection, ADF with the strict mode was used. In this mode, temperature values were strictly evaluated to detect anomalies. It also simultaneously incorporated federated anomaly directives from the voltage cluster to prevent false-positive alerts in the anomaly detection process. Anomaly detection results for different data samples are presented in Table 8. Improvement of ADF versus conventional anomaly detection is presented in Table 9. In the case of Sensor#1 with data anomaly detection, ADF recognized all anomaly alerts as false-positive except for the last interval of temperature data (Figure 5). In the Sensor#2 with data anomaly detection, despite spikes (sudden changes) in voltage values and generation of anomaly directives in the voltage cluster, some temperature values are considered as contextual anomalies. Also, at the early intervals of the data, the sensor voltage experiences spikes with random values resulting from the corresponding changes in the temperature cluster. However, some temperature values experience unexpected sudden changes, considered as an anomaly. As can be seen from Figure 6 and Table 8, ADF recognized most anomaly alerts as false positives based on the extracted information from contextual data (federated anomaly directives).

Table 8. Anomaly detection results.

Dataset	Anomaly counts without ADF	Anomaly counts with ADF
Sensor#1	152	102
Sensor#2	141	18

Table 9. Improvement of ADF versus conventional anomaly detection.

	Sensor#1	Sensor#2
ADF strict mode (False positive error)	32.9%	87.2%

5. Conclusions

This paper presented an approach for an efficient federated fog-assisted anomaly detection framework for Internet of Things (IoT). Anomaly Detection Fog (ADF) detected heterogeneous data anomalies in a context-aware and application-independent manner across a multi-context IoT. This was done through the collaboration of clusters in the form of a heterogeneous fog layer federation by sharing anomaly directives. Using the fog layer processing power, ADF responded to the constraint property of edge devices by the fog layer processing power. ADF presented fog layer federation via anomaly directives to prevent cloud offloading and reduce the communication overhead, energy consumption, and detection latency.

ADF evaluation proved that its application to a data anomaly detection algorithm would increase the true detection rate and decrease the false error rate in comparison to the use of the algorithm alone. Therefore, it is suitable to be applied to a network of resource-constrained things to achieve heterogeneous anomaly detection in a multi-context IoT.

We are developing the proposed framework further for future work. Our focus is on the anomaly dispatcher fog for better fog layer management in the clusters.

References

1. "https://www.owasp.org/index.php/Top_IoT_Vulnerabilities-Google Search". https://www.google.com/search?source=hp&ei=yaScW-aoC4LWkwWJ7ZeYDQ&q=https%3A%2F%2Fwww.owasp.+org%2Findex.php%2FTop_IoT_Vulnerabilities&oq=https%3A%2F%2Fwww.owasp.+org%2Findex.php%2FTop_IoT_Vulnerabilities&gs_l=psy-ab.3..33i160k1.2058.2058.0.2737.1.1.0.0.0.0
2. Du, J. and Chao, S. "A study of information security for M2M of IOT", *in 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, **3**, pp. V3-576-V3-579 (2010). DOI: 10.1109/ICACTE.2010.5579563.
3. Chandola, V., Banerjee, A., and Kumar, V. "Anomaly detection: A survey", *ACM Comput. Surv.*, **41**(3), p. 15 (2009).
4. Calo, S.B., Touna, M., Verma, D.C., et al. "Edge computing architecture for applying AI to IoT", In *2017 IEEE International Conference on (Big Data)*, pp. 3012–3016 (Dec. 2017). DOI: 10.1109/BigData.2017.8258272.
5. Xie, M., Han, S., Tian, B., et al. "Anomaly detection in wireless sensor networks: A survey", *J. Netw. Comput. Appl.*, **34**(4), pp. 1302–1325 (2011).
6. Rajasegarar, S., Leckie, C., and Palaniswami, M. "Anomaly detection in wireless sensor networks", *IEEE Wirel. Commun.*, **15**(4), pp. 34–40 (2008).
7. Aggarwal, C.C., Ashish, N., and Sheth, A. "The internet of things: A survey from the data-centric perspective", In *Managing and Mining Sensor Data*, Springer, pp. 383–428 (2013).
8. Butun, I., Morgera, S.D., and Sankar, R. "A survey of intrusion detection systems in wireless sensor networks", *IEEE Commun. Surv. Tutorials*, **16**(1), pp. 266–282 (2014).
9. Butun, I., Kantarci, B., and Erol-Kantarci, M. "Anomaly detection and privacy preservation in cloud-centric Internet of Things", In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pp. 2610–2615 (2015).
10. Lavin, A. and Ahmad, S. "Evaluating real-time anomaly detection algorithms-the numenta anomaly benchmark", In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 38–44 (Dec. 2015). DOI: 10.1109/ICMLA.2015.141.
11. Behniafar, M., Nowroozi, A., and Shahriari, H.R. "A survey of anomaly detection approaches in Internet of Things", *ISC Int. J. Inf. Secur.*, **10**(2), pp. 79–92 (2018). DOI: 10.22042/isecure.2018.116976.408.
12. "Transform your enterprise with an intelligent edge and IoT", Accessed: May 11, (2019) [Online]. Available: www.nutanix.com
13. Al-Khafajiy, M., Baker, T., Waraich, A., et al. "Iot-fog optimal workload via fog offloading", *Proc. - 11th IEEE/ACM Int. Conf. Util. Cloud Comput. Companion, UCC Companion* 2018, pp. 349–352 Jan. (2019). DOI: 10.1109/UCC-COMPANION.2018.00081.
14. Bonomi, F., Milito, R., Natarajan, P., et al. "Fog computing: A platform for internet of things and analytics", Springer, *Cham*, pp. 169–186 (2014). DOI: 10.1007/978-3-319-05029-4_7.
15. Ai, Y., Peng, M., and Zhang, K. "Edge computing technologies for Internet of Things: a primer", *Digit. Commun. Networks*, **4**(2), pp. 77–86, Apr. (2018). DOI: 10.1016/J.DCAN.2017.07.001.
16. "Cisco innovates in fog computing — The Network". https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1894659 (accessed May 11, 2019).
17. "Open Fog Reference Architecture for Fog Computing", 2017. Accessed: May 11, (2019). [Online]. Available: www.OpenFogConsortium.org
18. Misra, S. and Sarkar, S. "Theoretical modelling of fog computing: a green computing paradigm to support IoT applications", *IET Networks*, **5**(2), pp. 23–29 Mar. (2016). DOI: 10.1049/iet-net.2015.0034.
19. Aazam, M., Zeadally, S., and Harras, K. A. "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities", *Futur. Gener. Comput. Syst.*, **87**, pp. 278–289, Oct. (2018). DOI: 10.1016/J.FUTURE.2018.04.057.
20. Chiang, M. and Zhang, T. "Fog and IoT: An overview of research opportunities", *IEEE Internet Things J.*, **3**(6), pp. 854–864, Dec. (2016). DOI: 10.1109/JIOT.2016.2584538.
21. Raza, S., Wallgren, L., and Voigt, T. "SVELTE: Real-time intrusion detection in the Internet of Things", *Ad Hoc Networks*, **11**(8), pp. 2661–2674 (2013).
22. Le, A., Loo, J., Lasebae, A., et al. "6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach", *Int. J. Commun. Syst.*, **25**(9), pp. 1189–1212 (2012).
23. Thanigaivelan, N.K., Nigussie, E., Kanth, R.K., et al. "Distributed internal anomaly detection system for Internet-of-Things", In *Consumer Communications & Networking Conference (CCNC)*, 2016 13th IEEE Annual , pp. 319–320 (2016).
24. Mayzaud, A., Sehgal, A., Badonnel, R., et al. "Using the RPL protocol for supporting passive monitoring in the Internet of Things", In *Network Operations and Management Symposium (NOMS)*, 2016 IEEE/IFIP, pp. 366–374 (2016).
25. Tsitsiroudi, N., Sarigiannidis, P., Karapistoli, E., et al. "EyeSim: A mobile application for visual-assisted wormhole attack detection in IoT-enabled WSNs", In *Wireless and Mobile Networking Conference (WMNC)*, 2016 9th IFIP, pp. 103–109 (2016).

26. Sarigiannidis, P., Karapistoli, E., and Economides, A.A. "VisIoT: A threat visualisation tool for IoT systems security", In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pp. 2633–2638 (2015).

27. Surendar, M. and Umamakeswari, A. "InDReS: An intrusion detection and response system for internet of things with 6LoWPAN", In *Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on*, pp. 1903–1908 (2016).

28. Han, M.L., Lee, J., Kang, A.R., et al. "A statistical-based anomaly detection method for connected cars in internet of things environment", In *International Conference on Internet of Vehicles*, pp. 89–97 (2015).

29. Kartakis, S., Yu, W., Akhavan, R., et al. "Adaptive edge analytics for distributed networked control of water systems", In *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*, pp. 72–82 (2016).

30. Goodman, D.L., Hofmeister, J., and Wagoner, R. "Advanced diagnostics and anomaly detection for railroad safety applications: Using a wireless, IoT-enabled measurement system", in *2015 IEEE AUTOTESTCON*, pp. 273–279 (2015). DOI: 10.1109/AUTEST.2015.7356502.

31. Da Xu, L., He, W., and Li, S. "Internet of things in industries: A survey", *IEEE Trans. Ind. Informatics*, **10**(4), pp. 2233–2243 (2014).

32. Vijai, P. and Sivakumar, P.B. "Design of IoT systems and analytics in the context of smart city initiatives in India", *Procedia Comput. Sci.*, **92**, pp. 583–588 (2016).

33. Ho, C.-W., Chou, C.-T., Chien, Y.-C., et al. "Unsupervised anomaly detection using light switches for smart nursing homes", In *Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (DASC/PiCom/DataCom/CyberSciTech), 2016 IEEE 14th Intl C, pp. 803–810 (2016).

34. Ukil, A., Bandyoapdhyay, S., Puri, C., et al. "IoT Healthcare Analytics: The Importance of Anomaly Detection", In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 994–997 (2016). DOI: 10.1109/AINA.2016.158.

35. Kasinathan, P., Pastrone, C., Spirito, M.A., et al. "Denial-of-service detection in 6LoWPAN based internet of things", In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 600–607 (2013).

36. Ageev, S., Kopchak, Y., Kotenko, I., et al. "Abnormal traffic detection in networks of the internet of things based on fuzzy logical inference", In *Soft Computing and Measurements (SCM), 2015 XVIII International Conference on*, pp. 5–8 (2015).

37. Eliseev, V. and Gurina, A. "Algorithms for network server anomaly behavior detection without traffic content inspection", In *Proceedings of the 9th International Conference on Security of Information and Networks*, pp. 67–71 (2016).

38. Fu, R., Zheng, K., Zhang, D., et al. "An intrusion detection scheme based on anomaly mining in Internet of Things", In *4th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN 2011)*, pp. 315–320 (2011).

39. Chen, Z., Tian, L., and Lin, C. "A method for detection of anomaly node in IOT", In *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 777–784 (2015).

40. Liu, Y. and Wu, Q. "A lightweight anomaly mining algorithm in the Internet of Things", In *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on*, pp. 1142–1145 (2014).

41. Lyu, L., Jin, J., Rajasegarar, S., et al. "Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering", *IEEE Internet Things J.*, **4**(5), pp. 1174–1184 (2017).

42. Machaka, P., McDonald, A., Nelwamondo, F., et al. "Using the cumulative sum algorithm against distributed denial of service attacks in internet of things", In *International Conference on Context-Aware Systems and Applications*, pp. 62–72 (2015).

43. Ding, J., Liu, Y., Zhang, L., et al. "LCAD: A Correlation Based Abnormal Pattern Detection Approach for Large Amount of Monitor Data", In *Asia-Pacific Web Conference*, pp. 550–558 (2014).

44. Amin, S.O., Siddiqui, M.S., Hong, C.S., et al. "RIDES: Robust intrusion detection system for IP-based ubiquitous sensor networks", *Sensors*, **9**(5), pp. 3447–3468 (2009).

45. Trilles, S., Belmonte, Ó., Schade, S., et al. "A domain-independent methodology to analyze IoT data streams in real-time. A proof of concept implementation for anomaly detection from environmental data", *Int. J. Digit. Earth*, **10**(1), pp. 103–120 (2017). DOI: 10.1080/17538947.2016.1209583.

46. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Comput. Secur.*, **28**(1–2), pp. 18–28 (2009).

47. Liu, C.M., Chen, S.Y., Zhang, Y., et al. "An IoT anomaly detection model based on artificial immunity", *Advanced Materials Research*, (2012). https://www.scientific.net/AMR.424-425.625

48. Greensmith, J. "Securing the internet of things with responsive artificial immune systems", In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 113–120 (2015). DOI: 10.1145/2739480.2754816.

49. intrusion detection system (BMIDS) using internet of things (IoT) behavior-based anomaly detection via immunity-inspired algorithms", In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6 (2016). DOI: 10.1109/ICCCN.2016.7568495.

50. Zheng, Z., Wang, J., and Zhu, Z. "A general anomaly detection framework for internet of things", In *Proc. 41st IEEE/IFIP International Conference on Dependable Systems and Networks*, Hong Kong (2011).

51. Shilton, A., Rajasegarar, S., Leckie, C., et al. "DP1SVM: A dynamic planar one-class support vector machine for Internet of Things environment", In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*, pp. 1–6 (2015).

52. McDermott, C.D. and Petrovski, A. "Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks", *International Journal of Computer Networks and Communications* [online], **9**(4), pp. 45–56 (2017).

53. Thing, V.L.L. "IEEE 802.11 network anomaly detection and attack classification: A deep learning approach", In *Wireless Communications and Networking Conference (WCNC)*, 2017 IEEE, pp. 1–6 (2017).

54. Jain, R. and Shah, H. "An anomaly detection in smart cities modeled as wireless sensor network", In *Signal and Information Processing (IConSIP), International Conference on*, pp. 1–5 (2016).

55. Yu, T., Wang, X., and Shami, A. "Recursive principal component analysis-based data outlier detection and sensor data aggregation in IoT systems", *IEEE Internet Things J.*, **4**(6), pp. 2207–2216 (2017).

56. Hoang, D.H. and Nguyen, H.D. "A PCA-based method for IoT network traffic anomaly detection", In *Advanced Communication Technology (ICACT), 2018 20th International Conference on*, pp. 381–386 (2018).

57. Zhao, S., Li, W., Zia, T., et al. "A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things", In *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, 2017 IEEE 15th Intl, pp. 836–843 (2017).

58. "Standard deviation-Wikipedia". https://en.wiki -pedia.org/wiki/Standard_dev ation (accessed Jul. 27, 2021).

59. "Coefficient of variation-Wikipedia". https://en .wikipe-dia.org/wiki/Coefficient_of_variation (accessed Jul. 27, 2021).

60. Ahmed, M., Mahmood, A., Computer, J. H.-J. of N. and, et al. "A survey of network anomaly detection techniques", *Elsevier*, (2016). DOI: 10.1016/j.jnca .2015.11.016.

61. "CityPulse Smart City Datasets". http://iot.ee.sur -rey.ac.uk:8080/datasets.html (accessed Oct. 23,2018).

62. "Run Anomaly Detection On Your Data-Anomaly Detection in Azure Machine Learning". http://anom -alydetection-aml.azurewebsites. net/Single.aspx (accessed Jan. 25, 2020).

63. "Intel Lab Data". http://db.csail.mit.edu/labdata /labdata.html (accessed Aug. 09, 2021).

Biographies

Morteza Behniafar is a PhD candidate at Malek Ashtar University of Technology, Tehran, Iran. He received his BS and MS degrees in Computer Engineering from Isfahan University, Isfahan, Iran. His research interests include information security, intrusion detection systems, anomaly detection, and trust and reputation models.

Ali Mahjur received his PhD from Sharif university of technology (2006). Since 2011, he has been a faculty member of Malek Ashtar University of Technology. His research interests are operating system design, programming language design, computation theory, micro architecture of processors, and evolutionary algorithms.

Alireza Nowroozi is an Assistant Professor of Computer Engineering at the Engineering Department of IRIB University. Also, he is a consultant advising government and private sector-related industries on information technology. He has eight-year experience as an academic staff member and an IT post-doctoral position in Sharif University of Technology. He is a specialist in cognitive science, software engineering, and IT Security. Also, he has some national and international rewards.