IJICTR
International Journal of Information & Communication Technology Research

Volume 13- Number 3 – 2021 (12 -23)

# Anomaly Detection in Non-Stationary Water Distribution Grids Using Fog Computing Architecture

**Sara Mirzaie**
Department of Computer Engineering and Information Technology
Shiraz University of Technology
Shiraz, Iran
s.mirzaie@sutech.ac.ir

**Mohammad Reza AvazAghaei**
Department of Computer Engineering and Information Technology
Shiraz University of Technology
Shiraz, Iran
avaz.aghaei.sutech@gmail.com

**Omid Bushehrian***
Department of Computer Engineering and Information Technology
Shiraz University of Technology
Shiraz, Iran
bushehrian@sutech.ac.ir

Received: 5 March 2021 - Accepted: 10 May 2021

*Abstract*— **Efficient monitoring and quick feedback control are the main requirements of smart cities to guarantee the stability and safety of urban infrastructures. Real-time monitoring in order to detect anomalies leads to the intensive data processing and hence requires a new computing scheme to offer large-scale and low latency services. Fog architecture by extending computing to the edge of the network, provides the ability to accurate and fast detection of abnormal patterns. A hierarchical fog computing architecture and an efficient hyperellipsoidal clustering algorithm presented in previous studies have been applied to identify anomalous behaviors in water distribution grids. We created an urban water distribution grid dataset using Epanet2w simulator software by measuring grid features: pressure and head for several scenarios. We created 12 distinct events (unexpected behavior) with different scales during the simulation time. To evaluate the effectiveness of the hierarchical anomaly detection model in water distribution grids, the data and computing nodes at different layers were executed as docker containers. The evaluation results proved the efficiency of the proposed hierarchical anomaly detection model with a significant reduction in latency compared to the centralized scheme, while reaching a significant detection accuracy compared to the centralized one.**

**Keywords**: Internet of things; anomaly; clustering; fog computing; water distribution grid.

**Article type:** Research Article



* Corresponding Author



[ DOI: 10.52547/ijict.13.3.12 ]

## I. INTRODUCTION

The performance, sustainability and safety of smart cities are achieved by the integration of massive infrastructure components and services in the areas of energy, transportation, healthcare, education, smart homes, smart lighting, and utilities. Fulfilling these objectives, requires the real-time monitoring and analysis of the behavior of different infrastructure components and a quick feedback control system [1]. For monitoring the critical infrastructures in smart cities like bridges, gas/oil/water pipelines, roads, and subways, wireless sensor networks (WSNs) consists of small, cheap and intelligent sensor node with the ability to collecting raw data from a large scale area are the most suitable platform The sensor nodes monitor system behavior and measure required parameters. Malfunctions, faults or unexpected events in the environment may cause unexpected measurements by the sensors called anomaly. The anomaly is unusual observations that contradict the distribution of the majority of the data. In this context, it is important to identify and report those erroneous measurements to provide reliable and safe network performance. The process of detecting unusual behavior or hazardous events in the system is known as anomaly detection [2]. Identification of anomalous behaviors necessitates having a model for the majority of normal data, and then detection of the anomalies based on those data vectors, which are significantly differ from the normal model. Measurements collected by the sensor nodes form time-ordered data and anomalies can be detected by analyzing the time series data [3], [4]. However, sometimes, during the lifetime of data collection, the underlying phenomenon that is being observed may alter (concept drift) [5]. This will cause a change in the data distribution of the nodes; thus the data distribution will no longer be stationary rather a non-stationary one. If a system has a stationary data distribution, the model of the data from which anomalies are identified only needs to be constructed once. In contrast, in an environment with a non-stationary data distribution, it is necessary to construct a new model at certain time intervals in order to account for changes in the data distribution. In non-stationary systems, the data are temporally correlated, with correlation increasing as temporal distance decreases. Therefore, in order to achieve the best generalization error, the model needs to be formed from data that are temporally close to the data that will form the testing set [6].

Beside the non-stationary data distribution, another point to consider is the concept of big data. The large-scale distributed sensor networks generate a huge volumes of data, which leads to the challenge of processing big data. The centralized processing of high volume of data results in high processing delay, which is in conflict with the timing requirements of the time-sensitive applications [7]. Moreover, the large quantity of data causes high transmission traffic on the communication networks, and consequently the high communication delay. Location-awareness requirement are also necessary for some applications. Therefore, the cloud computing paradigm faces great challenges with the explosive amount of big data, the network bandwidth limitation, the low speed of data transmission, and the additional need for location awareness. Fog computing concept proposed by Cisco [8], is an efficient alternative to the cloud computing to meet these requirements. Fog computing extends the cloud computing architecture to the edge of the network to perform large-scale services throughout the network [9], [10]. Offloading some portions of the computing tasks to the fog nodes with computation and storage capability at the edge of the network will satisfy the requirements of low latency, low communication and location awareness in our applications. The Fog paradigm is well positioned for real-time big data analysis.

The concepts of smart monitoring and anomaly detection can be utilized to obtain the real-time control of smart grid conditions, and make decisions towards more efficient resource management. It has the potential to reduce peak demand, improve energy conservation, and enable the integration of renewable energy sources, which guarantees sustainable energy resource [11]. In this paper, smart monitoring is leveraged to detect anomalies in an urban water distribution grid. The urban water distribution grids as one of the most important infrastructures in the cities play significant role in the water supplying. However, several threats like aging and unexpected environmental events (sudden air temperature change) endanger the performance of water pipelines. These threats may result in corrosion, leakage, and failure of the grid, and consequently severe economic and urban problems [12].

In this paper, a four-layer hierarchical fog computing architecture is applied to detect anomalies in a non-stationary water distribution grid. In this research, anomalies are supposed as abnormal observations occurred via unexpected events at the water grids, which can be detected at any layers based on the extent of the area in which the data deviation happens. We define three different types of anomalies to be detected using the gateway, fog and cloud levels as follows. ***Local anomaly***, which is due to an element failure or malfunction like pipeline leakage as a prevalent failure in the water pipelines. This failure affects the measured data of one or more sensor nodes locally where anomaly has occurred and makes their data as outlier. ***Regional anomaly***, in which adverse events lead to failures or malfunctions like water supply disconnection or water pressure reduction at a region. Regional anomalies affect one or more locations of a region and change the normal behavior of sensor nodes in those locations. ***Global anomaly*** commonly caused by some global events like earthquake or extremely hot or cold weather that make global failures or malfunctions like water supply disconnection in the grid. Regarding these three types of anomalies, the task of detection could be offloaded at each corresponding layer of the hierarchical architecture.

The main contribution in this research is the evaluation of applying a hierarchical fog computing model for anomaly detection in a non-stationary water



[ DOI: 10.52547/ijict.13.3.12 ]

distribution grid compared to the centralized scheme in terms of the accuracy of detection and the amount of data transmissions. We exploited spatial/temporal correlation of data at each layer to detect existing anomalies in the created dataset. The results showed that the hierarchical model have made significant decrease in the data transmission compared to the centralized schemes, while achieving a comparable detection accuracy compared to the centralized one.

The rest of this paper is organized as follows: In Section II, we introduce some related works on anomaly detection, fog computing architecture and data correlations. In Section III, an overview of the hierarchical fog computing scheme and related methods for anomaly detection is described. Section IV illustrates the model for anomaly detection in water distribution grids. Evaluation results are presented in Section V. Finally, section VI concludes the paper.

## II. RELATED WORKS

### A. *Anomaly Detection with Fog Computing Scheme*

The challenges of analyzing the big data created by smart cities require using novel and high-performance architecture of fog computing. The key objective of fog computing architecture is distributing workloads throughout the network in order to reach low delay, less communication network overhead and higher performance computing capability. Bonomi *et al.* [13] described the fog computing advantages, which make it an appropriate choice for a number of real-time applications with low latency in Internet of Things (IoT) and big data processing. The Fog paradigm is well positioned for real-time big data analysis by supporting densely distributed data collection points, and providing advantages in terms of superior user experience.

Detecting interesting or unusual events as anomalies is an open issue in the data mining community. Non-parametric anomaly detection methods, does not have any prior knowledge about the distribution of the collected data at each time window. These methods are proper for dynamic environments where the condition and consequently, the data distribution may change frequently over the time (non-stationary). Lyu *et al.* [14] introduced a non-parametric distributed fog-empowered method for anomaly detection in large-scale systems. Authors utilized the fog computing advantages along with a hyperellipsoidal clustering algorithm and a scoring mechanism (ENOF) to detect anomalies at the vicinity of the network. Their research focus is using fog architecture for anomaly detection in order to diminish the latency and communication overheads.

Water distribution grids are prone to various types of threats, failures and unexpected events. Conventional anomaly detection techniques have been widely utilized for detecting anomalous measurements at these infrastructures. Daniel *et al.* [15] used the full label BATADAL dataset [16] to identify anomalies in the water distribution grids by applying several traditional anomaly detection approaches and proposing an ensemble technique. This technique uses a quadratic discriminant analysis (QDA) process that combines the output of a distance-based shared nearest neighbors (SOD) algorithm designed to detect outliers in high-dimensional data [17] with a local outlier factor (LOF) algorithm [18] to detect outliers in low-dimensional data to classify data points into anomalous or normal classes. Authors considered stationary systems and used supervised methods for centralized training. In [19], time series data modeling were applied by researchers to detect anomalies in smart power grids. They used statistical methods to detect outliers in the low volume data and applied RNN to recognize the normal behavior at a stationary grid with a centralized scheme. The authors in [20] focus on real-time identification of cyber-physical attacks on water distribution grids. They applied supervised machine learning anomaly detection techniques in stationary water grids by creating four modules. The first layer checks whether the given observations follow the right rules specified for the system, while the second layer finds statistical outliers. The third module has an Artificial Neural Network Model (ANN) that predicts the anomalies. The fourth module contains Principle Component Analysis (PCA) to classify data as normal or anomalies.

The aforementioned methods for the anomaly detection in the smart grids mostly applied the supervised machine learning methods for model classification without considering the concept drift in the data distribution. Moreover, these works mainly analyzed the system behavior based on a central scheme, which suffers from the scalability issues, the high latency and the high communication overhead. At this research, a hierarchical architecture along with an unsupervised detection method has been applied for the anomaly detection problem in a water distribution grid.

### B. *Correlation of Data*

In WSNs, in order to certify the full coverage of a monitored environment, a spatially dense deployment of sensor node is required [21], [22]. This deployment results in observing same condition by multiple sensor nodes. For example, in water distribution grid, sensor nodes measure same physical features for water at pipes; consequently, they have the same data distribution. A set of sensor nodes within a spatial proximity, which measure the same phenomenon have the spatial correlation of their data. These spatial correlation among gathered data could be used to detect anomaly at that time.

In addition to spatial correlation, temporal correlation of data may occur. When the underlying features of the phenomenon that is being recorded change gradually over the time, temporal correlation arises between consecutive data points. Data



[ DOI: 10.52547/ijict.13.3.12 ]

measurements on an individual sensor node become temporally correlated due to the nature of the phenomenon that is being monitored; for example, in distribution grids, pressure measurements at each consumer node exhibit a predictable behavior pattern (gradual change) during the lifetime of simulation. Temporal correlation can also be used to detect anomalies by comparing data point of several sequential time windows [6].

Therefore, spatial-temporal correlation of data may occur in WSNs where data collected on different nodes and at different times, exhibit a predictable relationship. The spatial, temporal and spatial-temporal correlation of data can be exploited to identify an anomaly and determine its cause [6]. Vuran *et al.* [23] studied data correlations in order to reduce energy consumption in a WSN. The objective of this research is to exploit spatial/temporal correlation of the WSN paradigm to enable the development of efficient communication protocols. They use spatial and temporal correlation for efficient medium access and reliable event transport in WSN, respectively.

Anomalies caused by errors occur independently, whereas anomalies caused by events exhibit spatial and/or temporal correlation. At this paper, we defined anomalies occurred by unexpected events with spatial/temporal correlation.

## III. FOG COMPUTING ARCHITECTURE

### A. *Distributed Schemes*

In WSNs, raw data are recorded by the individual sensor nodes, which are dispersed in a physical environment and monitor the environmental conditions of their vicinity. The spatial correlation of sensor nodes ensures the similar experiences of one sensor node to the other close nodes, hence it is useful for these nodes to share identified characteristics of their data for better perception of the system behavior. This may lead to a distributed learning structure where information describing the data of one sensor node is communicated with other nodes to build a comprehensive model of the environment to identify the outliers and the anomalous sensor nodes accurately [24]. Learning in a distributed environment is divided into two distinct categories; hierarchical and central.

In the centralized approach [15], [19], [20], all sensor data are transmitted via multiple hop communication to a central node. The central node constructs the data model using the whole data, and anomalies are detected by analyzing the created model. High accuracy in the anomaly detection process is attained due to the computational power of the central node that enables it to run more computationally complex anomaly detection algorithms on the immense amounts of data. Though, the communication costs in transmitting all local nodes data measurements to a central node could be prohibitive. In addition, scalability issues when the measurement numbers scales up become a noticeable problem. Finally, the delay incurred by the transmission of the massive data to a central node and processing that big data to detect anomalies increases the response time for online applications. Therefore, cloud computing architecture (centralized approach) cannot meet the requirements of scalability, communication cost and timely response in the large-scale real-time applications.

Hierarchical learning attempts to limit the transmissions to a central node by building sub-models locally and merging sub-models to a complete model as data goes up in the hierarchy. An intermediate node in the hierarchy run the same instance of the model fusion and anomaly detection algorithms to first build a parent model from received sub-models and then check for anomalies. Intermediate nodes merely transmit information about the local models to the parent node in the higher layer rather than the whole data from sensor nodes in the network. Summarized information that contains the form of model parameters and/or anomalies, is transmitted to ensure a reduction in transmission time and load in transmitting nodes. Furthermore, the hierarchical scheme supports location-awareness of anomalies and allows different types of anomalies to be detected based on the range of anomaly, namely, local anomalies, regional anomalies and global anomalies, using the fog and cloud level cluster information. However, the hierarchy can affect the accuracy of anomaly detection by indirect information exchange among all the end nodes either.

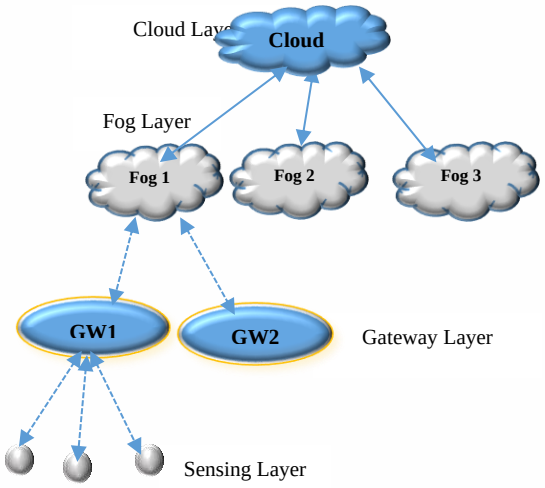



Fig 1. Hierarchical four-layer fog computing architecture.

Considering the scale of water grids, we use a four-layer hierarchical fog computing architecture (Fig. 1) for anomaly detection as follows [1]:

- Sensing layer: which is comprised of numerous sensor nodes monitoring the environment features and record required data at the regular time intervals. At



[ DOI: 10.52547/ijict.13.3.12 ]

the end of each time window, sensor nodes forward its raw data into the upper layer (gateway layer).

- Gateway layer: which contains intermediate nodes with relatively low-power and high performance capability. Each gateway devices is connected to and responsible for a local group of sensors in its vicinity. Gateway nodes receive the collected raw data from their local sensors and process them to detect anomalies. The gateways output involves: (1) summary information of the processed data to be sent to the next upper layer in the hierarchy, (2) a signal message to be sent to one or more of its local sensors to alarm about the detected malfunction or fault at them.
- Fog layer: consists of a number of nodes with intermediate storage and computing capability connected to a group of gateway nodes. Fog nodes should process transmitted summary information from the gateways to identify the potential anomalous events. They merge received information from the gateway layer, analyze them and send the summaries to the cloud layer. They also make quick control response to the gateway layer when an unexpected event is detected.
- Cloud layer: All the results of cluster analysis at the fog layer are communicated to the cloud layer, for more comprehensive and precise system analysis and global monitoring. This layer provides city-scale monitoring and centralized controlling via a cloud computing data center. Next, we describe a clustering algorithm uses hyperellipsoidal clusters to model the collected data of sensor nodes.

### B. *Clustering Algorithm*

Based on the statistical and machine learning algorithms, the anomaly detection approaches are classified as follows: clustering-based approaches, classification-based approaches, dimension-reduction-based approaches, and hybrid approaches that combine multiple technologies together [25]. The clustering approach, which is the process of dividing data points to several groups such that each group contains highly similar data points, has been broadly applied as a non-parametric knowledge discovery tool in the systems with restricted resources, such as the wireless sensor networks. In the data clustering based approaches, the data are first clustered and then anomaly detection method is applied to detect the outliers and anomalous clusters [26]. We applied unsupervised HyCARCE [26] algorithm as a computationally efficient clustering algorithm in our anomaly detection platform. This algorithm can model many different data distributions including hyperspherical to linear. Besides, the number of clusters is chosen by an automatic mechanism, and a linear computational overhead is imposed in terms of the number of data vectors processed. This algorithm has an input parameter: initial grid cell width $w$. The main steps of the HyCARCE algorithm are as follows [26]:

Step 1. At first, the input space is divided into a set of same size cells with d-dimension. Then, empty cells are removed.

Step 2. The grid cells that contain a small number of data points are removed. If the cardinality of an initial cell is less than the mean value minus the standard deviation value of all data points, the grid cells is deleted. Then, the clusters is created over the data points of the remaining cells. The mahalanobis distance with the sample mean μ of data points in a cell having covariance matrix Σ is used to make hyperellipsoid clusters around the mean in each cell (Eq.1). The threshold $t = (\chi_d^2)_p$ (i.e., the inverse of the chi-squared statistic with d-degrees of freedom) as the effective radius results in a hyperellipsoids with at least $p = 95\%$ coverage of the data points of a cell. This threshold is used to make the boundary of hyperellipsoids. Each data point $x$ that satisfies the following equation falls inside the hyperellipsoid $e$.

$$e(\mu, \Sigma^{-1}, t) = \{x \in \Re^d \| (x - \mu)^T \Sigma^{-1} (x - \mu) \leq t\} \quad (1)$$

Step 3. At this step, the algorithm enlarges the ellipsoids to better fit the shape of the cluster. The enlargement is achieved by scaling the inverse covariance matrix $\Sigma^{-1}$ by the scaling factor $S_f$ as shown in Eq. (2). The amount of the scaling factor mainly depends on the distribution of the data. In a very dense data distribution, a value close to one can be chosen for this factor, in contrast to sparser data distributions the smaller value is more suitable [25]. New mean and new covariance matrix are recalculated based on the new data points inside the enlarged ellipsoids and ellipsoid boundaries are adjusted to incorporate the new data points. These processes continue until the number of new added data points to the new clusters becomes less than a threshold.

$$\Sigma^{-1}{}_{enlarged} = S_f \times \Sigma^{-1} \quad (2)$$

Step 4. At the last step, the algorithm identifies the redundant ellipsoids which their center are very close to each other and delete the one with less number of the data points. After removing the redundant ellipsoids, the remaining ellipsoids mark the boundaries of the clusters. Next, we discuss an algorithm that analyzes these hyperellipsoidal clusters and provides an outlierness score to identify anomalous clusters and detect anomalies.

### C. *Spatial Correlation*

Once a set of hyperellipsoidal clusters are created, a scoring mechanism should be applied to identify the normal and anomalous clusters. Regarding the spatial correlation of data at each time window, we use ENOF [26] algorithm to classify clusters as the normal and anomalous base on an outlierness score for each ellipsoid. ENOF mainly relies on the distance metric and the use "focal distance" between two ellipsoids to find close neighborhoods of each ellipsoid [27]. Then, the outlying ellipsoids are identified relative to their



[ DOI: 10.52547/ijict.13.3.12 ]

close neighborhood, with respect to the densities of their neighborhoods. ENOF mechanism calculates an outlierness scoring parameter by comparing the reachability density of each ellipsoid with the average reachability density of its close neighbors in order to identify the ellipsoids which are outlying relative to their close neighborhoods. In particular, an ellipsoid that belongs to a dense group of ellipsoids has a smaller outlier score than an ellipsoid that is far from this group of ellipsoids. This is a ratio between the average neighborhood reachability density of the neighbors and the ellipsoids' own neighborhood reachability density. This ratio becomes 1 when an ellipsoid becomes comparable to its neighboring ellipsoids. For the faraway ellipsoids from their neighbors, the ENOF becomes significantly higher than 1. ENOF scores are used to determine the anomalous clusters via comparing cluster scores with a Threshold computed using the ENOF scores.

The ENOF procedure can only work efficiently, when an event affects a number of nodes at one layer and a part of ellipsoids or data deviate from the rest of ellipsoids or data. Then, the procedure detects the outliers in comparison with the normal data of each time window. Although, some events similarly affect the behavior of all nodes at a location and consequently, the whole data model of that location. Hence, the ENOF algorithm cannot detect these anomalies effectively. At the next, we discuss temporal correlation and introduce a method to detect these anomalies based on the temporal similarity.

### *D. Temporal Correlation*

As mentioned in introduction, alterations in the condition of underlying environment that is being monitored make a non-stationary system, in which collected data changes during the lifetime of the environment. Data distribution of sensor nodes gradually changes along with the environment changes. Since the data measurements at close intervals are expected to be more correlated, temporal correlation can be exploited to detect unexpected and sharp data changes portending hazardous events. Therefore, we compare the data models of two consecutive time windows in order to detect abrupt changes and accordingly the temporal anomalies.

The procedure uses the similarity of two consecutive models to detect temporal changes. It compute these similarity as follows: At first, the procedure checks whether two models are exactly same or there is a change. If the model has changed, it calculates the amount of change (model shift) by computing the average focal distances of each ellipsoid in the current model from all ellipsoids of the previous one. Then, it calculates the average of these average distances computed for each ellipsoid. This value is compared with the temporal threshold $T$ to detect abrupt change and unusual events. The temporal change greater than $T$ notifies an anomaluy.

**Algorithm 1:** Temporal Change Detection

**Role:** Computing nodes (gateway, fog, cloud) calculate the temporal similarity between two models (i , j)

{

Compare tow models (i , j)

If (not copy)

{

for each ellipsoid $e_k$ in model i

compute the focal distance of ellipsoid $e_k$ from the all ellipsoids of model j.

calculate the average $t_k$ of these focal distances.

compute $t_{ij}$ as the average of all computed $t_i$ for each ellipsoid.

if ($t_{ij} \leq T$)

"no temporal anomaly detected."

else

"temporal anomaly detected."

}

Fig. 2 shows the measured accuracy values for the spatial and spatial-temporal correlations. A four-layer fog computing architecture and a water grid dataset were used.

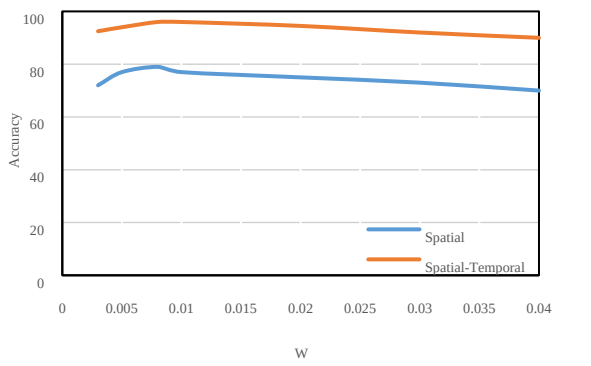



Fig 2. accuracy in anomaly detection with spatial and spatial-temporal correlations.

### E. *Merging Algorithm*

Fog or cloud nodes at the upper layers should firstly merge their received clusters from the lower layers in order to obtain their own level clusters. Clusters can be merged in a pairwise manner. For each pair of clusters $e_i$ and $e_j$, with mean vectors $\mu_i$ and $\mu_j$, covariance matrices $\Sigma_i$ and $\Sigma_j$, and the number of cluster elements $N_i$ and $N_j$, the merged (hyperellipsoidal) cluster $e_m$ will have the mean vector $\mu_m$, covariance matrix $\Sigma_m$ and the number of cluster elements $N_m$ computed as follows [27]:



[ DOI: 10.52547/ijict.13.3.12 ]

$$N_m = N_i + N_j \quad (3)$$

$$\mu_m = \frac{N_i}{N_m}\mu_i + \frac{N_j}{N_m}\mu_j \quad (4)$$

$$\Sigma_m = \frac{N_i-1}{N_m-1}\Sigma_i + \frac{N_j-1}{N_m-1}\Sigma_j + \frac{N_i N_j}{N_m(N_m-1)}[(\mu_i - \mu_j)(\mu_i - \mu_j)^T] \quad (5)$$

Many researchers choose Euclidean distances as a metric for measuring the similarity of two clusters. Two clusters should be close to one another and they should have a large number of data points in common to be merged; there should not be a significant gap where no data sample exists [28]. Accordingly, we used focal distance [27] between two ellipsoids as a metric for choosing two clusters to merge (Fig. 3). If the focal distance of two ellipsoid is less than a merging threshold *R*, two ellipsoid are merged. Next, we present the introduced anomaly detection methods to identify anomalies in the water distribution grids with the four-layer fog computing architecture.

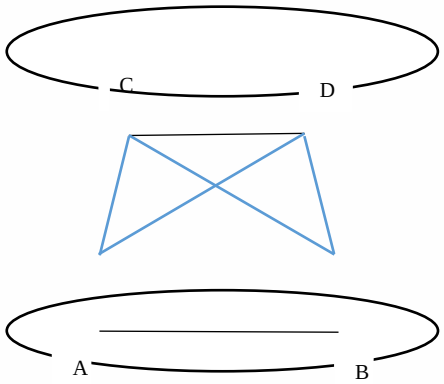



Fig 3. Focal distances between two ellipsoids.

## I. WATER DISTRIBUTION GRID MODELING

In this section, the overview of the hierarchical model for anomaly detection is presented. A smart water grid should have ability to monitor the safety of pipelines throughout the grid and detect potential dangerous events. By monitoring and analyzing the physical parameters of water (pressure, flow and head) during a time window, unexpected behaviors can be detected. These behaviors may indicate a system failure or an emergency event. Indeed, the basis of the scheme is the real-time surveillance of the water distribution grid, processing the received data and accurate and quick detection of unexpected events as anomalies.

The hierarchy of the scheme alleviates the computational overhead imposed at the cloud. Further, using the computing nodes at the edge of the network helps early identification of anomalies along with minimizing the data transmission. The detailed functions of each node in the hierarchy are as follows:

- Each sensor node gathers the raw data of water pipelines during a time window and transmits its measured data to the upper connected gateway node.

- Gateway nodes at each time window, after receiving all the raw data from their associated sensor nodes, perform clustering on the data using the hyperellipsoidal clustering algorithm (HyCARCE) to model the gateway level clusters. Then, the ENOF algorithm is applied on the clusters to find the outliers and anomalous clusters as the local level anomalies. Gateway raises alarm messages for any possible detected anomalies. However when an undesirable event affects all the sensor nodes in a time window the gateway will not be able to detect anomalies by applying ENOF on the clusters. This is where the temporal correlation comes to play. Comparing the temporal difference of two constitutive models, abrupt change in the current model could be detected. Abrupt change indicates an unexpected event in the underlying environment monitored. This detection is alarmed by the gateway nodes. Then, the gateway clusters summary information (ID, mean, and covariance matrix) are communicated to the fog nodes at the next upper layer for regional analysis.

- The Fog nodes merge the received cluster data of their sub-ordinate gateway nodes based on the procedure explained in the previous section by using a user-defined parameter *R* as the merging threshold . ENOF is applied on the merged clusters to classify the anomalous and normal clusters and find the spatial regional level anomalies. Then temporal similarity is calculated to detect temporal anomalies at this level. Since the regional anomalies affect nearly all parts of a region, all sensor data of one or more gateways are affected; the sensor nodes of these gateways are labeled as anomalous and the fog node creates an alarm signal to their connected gateways. After that, the fog level clusters will be transmitted to the upper layer for global analysis.

- The received clusters at the cloud layer are merged in order to form the cloud layer clusters. Then, ENOF is exploited to find spatial anomalies at the cloud layer and consequently the temporal similarity is exerted to identify abrupt temporal changes. If a global anomaly have been occurred in the water distribution grid, the cloud can detect that and send an alarm signal to the related nodes.

## IV. EVALUATION RESULTS

In this section, the accuracy and percentage of the communication saving for the hierarchical fog computing scheme are evaluated compared to the centralized scheme in the previous studies ([15], [19], and [20]). An evaluation test-bed were crated as a set of Docker [29] containers for emulating the four-layer fog and centralized architectures. Any intermediate nodes in the hierarchy executed as a container with predefined resource capacity. For emulating the sensor nodes functionality, a containerized Node-Red [30] process were used that successively queries measurements from a database node and transmits to



[ DOI: 10.52547/ijict.13.3.12 ]

the upper layer interval by interval using MQTT [31] protocol. All the transmitted messages are JSON strings containing the sensor data, intermediate models or anomaly alerts.

A. *Data Sets*

For this research, we made a real-world dataset by simulating a sample of water distribution grid in a city [32]. Epanet 2 + WaterNetGen simulation software [33] was used for designing and implementing an urban water distribution grid. The software is able to perform extended-period simulation of the hydraulic and water quality behavior within pressurized pipe networks, which consist of pipes, consumer nodes, storage tanks, and so on. It can be used to track the flow of water in each pipe, the pressure and head of water at each consumer node, the height of the water in each tank, a chemical concentration, the age of the water, and source tracing throughout the network during a simulation period.

A water distribution grid consists of three distinct three streets regions were designed by the software, in each region each including six consumer nodes was deployed in the software. The hierarchical fog architecture configuration used for the aforesaid data set are illustrated in Fig. 4. Each consumer node was monitored by a sensor node. Hence a total of 54 sensor nodes were used to cover this grid. Streets and regions are monitored by a gateway and fog nodes respectively. The whole grid is under the surveillance of one cloud node. In contrast, in the centralized configuration, all 54 sensor nodes are monitored by a central node and the data is transmitted to it by multi-hop communications.

The Epanet software was set to record water parameters at the consumer nodes every 30 seconds during the simulation time to make the dataset. The sensor nodes send the collected measurements during a 30 minutes time window to the gateway nodes. The Simulation lasted six hours and at each time window, 4300 two-dimension data vectors were measured by each sensor node, having normal distribution (proved by the Kolmogorov–Smirnov test) with various cluster overlaps degree.

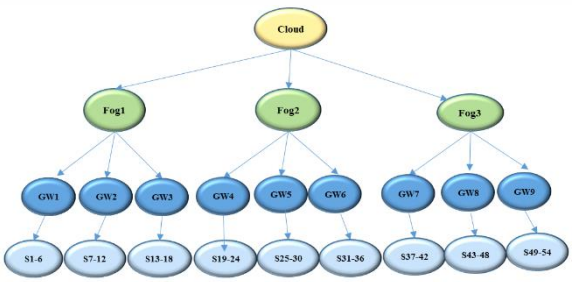



Fig 4. The hierarchical fog architecture configuration used for the evaluation.)

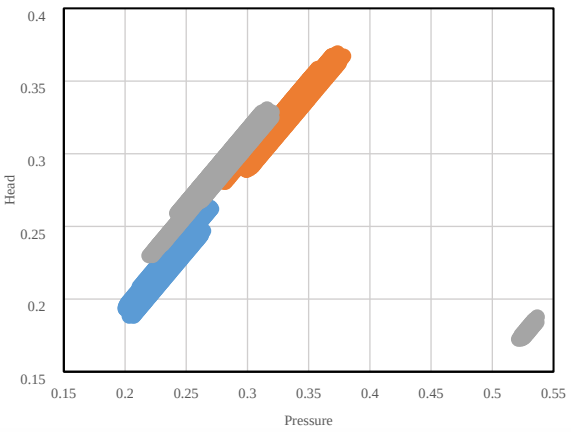



Fig 5. Scatter plot of all combined data at a time window.

Fig. 5 shows the scatter plot of the measurements made by all 54 sensor nodes in a time window. This plot shows three distinct colors denoting the measured data at each region during a time window. We used the maximum/ minimum value of the combined data of all sensors to normalize data to the range [0, 1]. As you can see, an anomaly is observed in the plotted data. It is a small collection of data vectors in the lower right hand corner of the plot that differs significantly from the majority of the data. This anomalous data constitutes a part of the data from the region 3 (fog node 3). We labeled these visually obvious anomalous data vectors as anomalies, and the rest of the data vectors as normal for our evaluation purposes.

B. *Accuracy*

Here, we compared the detection accuracy of the proposed hierarchical scheme with the centralized anomaly detection scheme considering the temporal and spatial correlations. In the centralized detection, anomalies were identified based on all the 54 sensors' data at each sliding window. The accuracy of the scheme was assessed based on these two parameters: $(i)$ true positives $(TP)$, and $(ii)$ true negatives $(TN)$. The number of correctly detected anomalous behavior are defined as $TP$, and the number of correctly detected normal behavior are defined as $TN$. Using these at each time window, the accuracy parameter was computed as $accuracy = (TP + TN)/n$, where $n$ was the number of data vectors in each sliding window [14]. Finally, the overall accuracy was considered as the average of all computed accuracies at each time window.

Experiments repeated using different values for two parameters of the HyCARCE and merge algorithms: cell size $w$ and the merging threshold R [14]. Fig. 6 shows the results for the accuracy measurement with different window sizes, while keeping the merging threshold $R$ fixed at 0.005. The scaling factor of HyCARCE is set to 0.95, the z and k values of the ENOF are set as 3 and 25% respectively, and the temporal threshold $T$ is set .005 in this



[ DOI: 10.52547/ijict.13.3.12 ]

research. The research findings affirm the close accuracy to the centralized scheme; the smaller cell sizes result in the closer accuracy to the centralized scheme. Because, using lower $w$ produces more number of clusters leading to similar results to the centralized scheme.

In addition to the cell size, the merging threshold can also affect the detection accuracy. Fig. 7 illustrates the accuracy results for a range of $R$ with the fixed cell size $w = 0.01$. As you can observe in the Fig. 7, in the bigger $R$ values more number of the clusters will be merged which causes much more information loss and less accuracy. While in the lower $R$ values, less merged clustered results in a better accuracy.

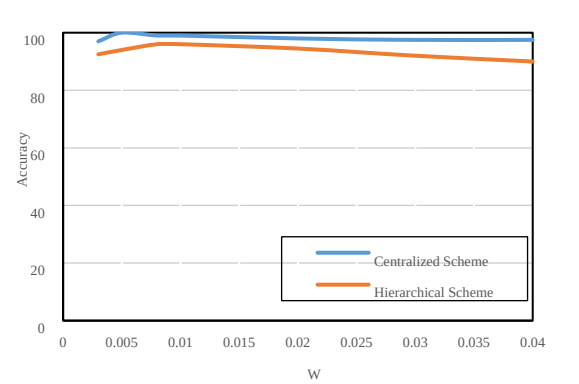



Fig 6. Example Anomaly detection accuracy with different values of $w$.

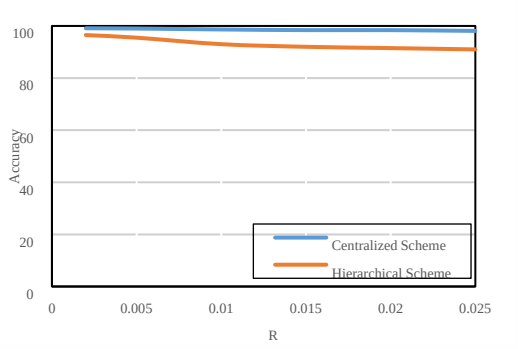



Fig 7. Example Anomaly detection accuracy with different values of $R$.

## C. *Communication Traffic*

We compared the hierarchical fog computing scheme with the centralized cloud computing scheme based on the number of inter-layer data transmissions. In the fog model, the raw data is just transmitted between the sensor and the gateway layers, after that only the clustering information (model of the data) is communicated along the hierarchy. Hence, it is clearly expected that a notable reduction in communication traffic will be achieved in this scheme. This is in contrast to the centralized scheme where no intermediate data model is constructed and all transmissions involve detailed sensor measurements.

We performed simulations for different cell sizes $w$ ranging from 0.007 to 1 in 0.05. Each sensor node in the centralized scheme is assumed to be three hops away from the cloud and the raw data vectors are passed through three communication hops to reach the cloud. The saving percentage in communication load was calculated based on Eq. (6) that NTH denotes the total number of the transmissions (data and cluster information) in the hierarchical scheme and NTC denotes the number of the transmissions in the centralized Scheme.

$$\text{Saving Percantage} = \left(1 - \frac{NTH}{NTC}\right) * 100 \qquad (6)$$

The total reduction in traffic for different cell sizes and different values of $R$ are shown in Fig. 8 and 9 respectively. In Fig. 8, the fixed merging threshold R = .005 and in the Fig. 9, the fixed cell size $w = 0.01$ were used.

It was observed that the larger the cell size is chosen, the higher reduction in communication traffic is achieved. Larger cell sizes result in a clustering with fewer numbers of clusters which causes a smaller data model to transmit to the upward layer (Fig. 8). It argument is also true for explaining the results in Fig. 9.

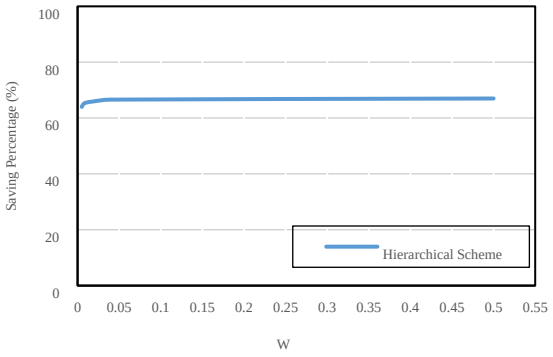



Fig 8. The percentage of communication saving in the hierarchical architecture for different cell sizes.

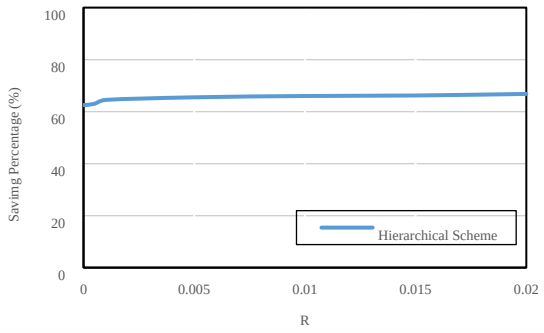



Fig 9. The percentage of communication saving in the hierarchical architecture with different R values.



[ DOI: 10.52547/ijict.13.3.12 ]

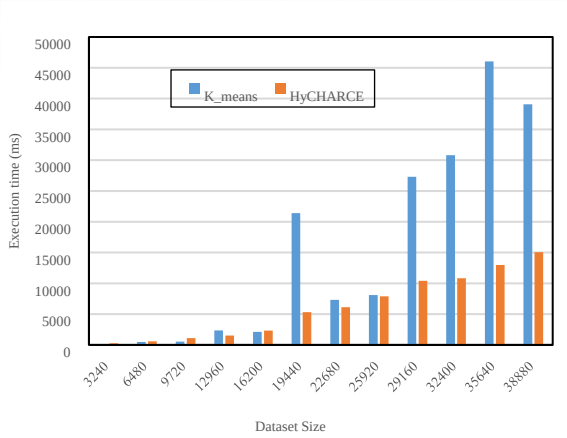


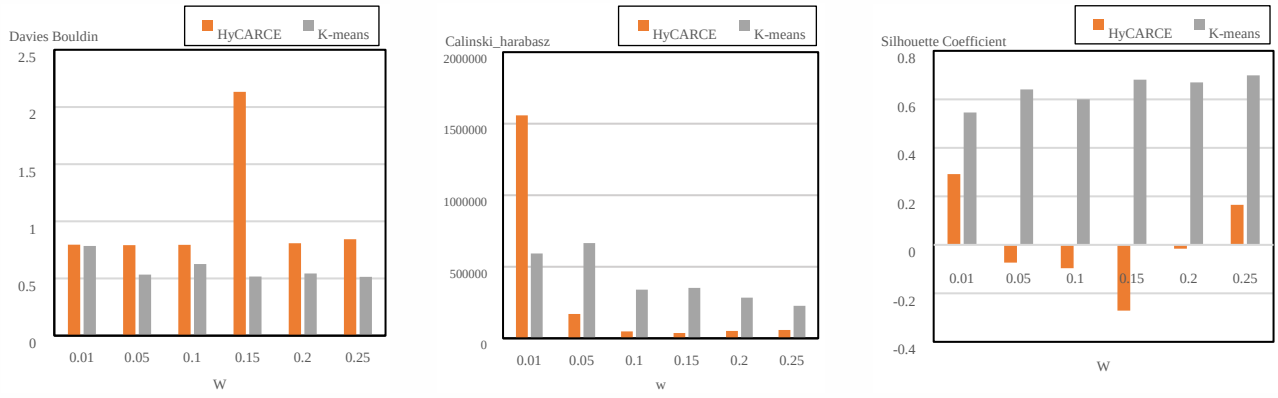

Fig 10. Comparison of the execution times of HYCARCE and K-means Algorithms for different dataset sizes.

### D. *Clustering Performance*

The results of comparing HyCARCE clustering algorithm with the well-known K-means algorithm in terms of execution time and accuracy are presented here. Both HyCARCE and K-means algorithms were applied on the dataset to create clusters. As shown in Fig. 10, for the small dataset sizes, two algorithms had similar execution times, however as the size of dataset increases the HyCARCE algorithm create clusters much faster. To compare the accuracy of algorithms, three internal metrics: Silhouette Coefficient (the higher value means better quality), Calinski_Harabasz (the higher value means better quality) and Davies Bouldin (the lower value means better quality) [34] were used. Fig. 11 shows the result of comparison of two algorithms in terms of accuracy. Obviously, opposed to k-means, the accuracy of the HyCARCE algorithm is very sensitive to the selection of cell size. As shown in this figure w=0.01 worked the best for HyCARCE. However overall, k-means outperformed HyCARCE in terms of accuracy.

## V. CONCLUSIONS

Fog computing is an interesting scheme for collecting and processing the expanding amounts of IoT data in large scale surveillance applications. In the time-critical applications, quick and accurate detection of the anomalous behaviors in the environment is the most important challenge. We used a four-layer fog computing architecture in order to detect anomalous consumption patterns in water distribution grids. The hierarchical architecture makes possible the early and accurate identification of various ranges of anomalies. This scheme resulted in real-time detection of anomalies with low inter-layer communication traffic compared to the centralized schemes. Evaluation results proved that the hierarchical fog computing architecture could reach to acceptable anomaly detection accuracy compared to the centralized scheme. As the future work, we aim to apply the hierarchical clustering algorithm to locate faulty elements in distribution grids.




(a) (b) (c)

Fig 11. Example Comparisons of the accuracy of HYCARCE and K-means algorithms based on: (a) Davies Bouldin metric, (b) Calinski_Harabasz metric and (c) Silhouette Coefficient metric for different values of $W$.

## REFRENCES

[1] Tang, Zhen Chen, G. Hefferman, S. Pei, T. Wei, H. He, Q. Yang, "Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities," IEEE Transactions on Industrial Informatics, vlo.13, No. 5, October 2017.

[2] A. Gaddam, T. Wilkin, and M. Angelova, "Anomaly detection models for detecting sensor faults and outliers in the IoT-a survey," 13th International Conference on Sensing Technology (ICST). IEEE, pp. 1–6, 2019.

[3] Andrew A.Cook, Göksel Mısırlı, Zhong Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," IEEE Internet Things J., vol. 7, pp. 6481–6494, 2020.

[4] Jiuqi Zhang, Di Wu, Benoit Boulet, "Time Series Anomaly Detection for Smart Grids: A Survey," IEEE Canadian Electrical Power and Energy Conference, 2021.

[5] C. H. Tan, V. C. Lee, and M. Salehi, "MIR_MAD: An Efficient and On-line Approach for Anomaly Detection in Dynamic Data Stream," 2020 International



[ DOI: 10.52547/ijict.13.3.12 ]

Conference on Data Mining Workshops (ICDMW), pp. 424-431, 2020.
[6] O'Reilly, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Anomaly detection in wireless sensor networks in a non-stationary environment," IEEE Communications Surveys Tutorials, vol. 16, no. 3, pp. 1413–1432, Third 2014.
[7] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," IEEE Comput. Intell. Mag., vol. 9, no. 4, pp. 62–74, Nov. 2014.
[8] Cisco, C. V. N. I. (2015), Global Mobile Data Traffic Forecast Update, 2019 (white paper).
[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.
[10] Miorandi, S.Sicari, Fracesco D.Pellegrini, I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks journal, vol. 10, pp.1497-1516, 2012.
[11] R. Moghaddass and J. Wang, "A Hierarchical Framework for Smart Grid Anomaly Detection Using Large-Scale Smart Meter Data, " in *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5820-5830, Nov. 2018.
[12] L. Mart´ı, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," Sensors, vol. 15, no. 2, pp. 2774–2797, 2015.
[13] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," Big Data and Internet of Things, 2014.
[14] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in internet of things using hyperellipsoidal clustering," IEEE Internet of Things Journal, 2017.
[15] Daniel Ramotsoela, Gerhrad Hancke, and Adnan M.Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," Hum. Cent. Comput. Inf. Sci., 2019.
[16] Taormina, R, Galelli, S, Tippenhauer, NO, Salomons, E, Ostfeld, A, Eliades, DG, Aghashahi, M, Sundararajan, R, Pourahmadi, M, Banks, MK, "Battle of the Attack Detection Algorithms: Disclosing cyber-attacks on water distribution networks," Journal of Water Resources Planning and Management, vol. 144, no. 8, 2018.
[17] C.C. Aggarwal, "High-Dimensional Outlier Detection: The Subspace Method. In Outlier Analysis," Springer New York: New York, NY, USA, pp. 135–167, 2013.
[18] Markus M.Breunig, Hans-Peter Kriegel, Raymond T.Ng, Jorg Sander, "LOF: Identifying Density-Based Local Outliers, " In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, Dallas, TX, USA, 16–18 May 2000; Association for Computing Machinery: New York, NY, USA, pp. 93–104, 2000.
[19] Q. Wei et al., "GLAD: A Method of Micro-grid Anomaly Detection Based on ESD in Smart Power Grid, " 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 103-107, 2020.
[20] Abokifa, Ahmed A., Kelsey Haddad, Cynthia Lo, and Pratim Biswas, "Real-time identification of cyber-physical attacks on water distribution systems via machine learning–based anomaly detection techniques, " Journal of Water Resources Planning and Management, vol.145, no. 1, 2019.
[21] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in Proc. 1st ACM, Int. Workshop on Wireless Sensor Networks and Applicat., Atlanta, GA, pp. 42–48, Sept. 2002.
[22] Susana C.Gomes, Susana Vinga, and Rui Henriques, "Spatiotemporal Correlation Feature Spaces to Support Anomaly Detection in Water Distribution Networks," Water, vol. 13, no. 18: 2551, 2021.
[23] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," Comput. Netw., vol. 45, no. 3, p. 245, 2004.
[24] S. Rajasegarar, C. Leckie, M. Palaniswami, J.C. Bezdek, "Distributed anomaly detection in wireless sensor networks," 10th IEEE International Conference on Communication systems, Singapore, pp. 1–5, 2006.
[25] Jinfang Jiang, Guangjie Han, Li Liu, Lei Shu, Mohsen Guizani, "Outlier Detection Approaches Based on Machine Learning in the Internet-of-Things, " IEEE Wireless Communications, vol. 27, no. 53-59. 2020.
[26] M. Moshtaghi, S. Rajasegarar, C. Leckie, and S. Karunasekera, "An efficient Hyperellipsoidal clustering algorithm for resource-constrained environments," Pattern Recog., vol. 44, no. 9, pp. 2197–2209, 2011.
[27] Sutharshan Rajasegarar, A. Gluhak, M. A. Imran, M. Nati, M. Moshtaghi, C. Leckie, M, Palaniswami, "Ellipsoidal neighborhood outlier factor for distributed anomaly detection in resource constrained networks," Pattern Recognition, vol. 47, no. 9, pp. 2867–2879, 2014.
[28] P. M. Kelly, "An algorithm for merging hyperellipsoidal clusters," Los Alamos National Laboratory, Tech. Rep, 1994.
[29] Docker Desktop on Windows, https://docs.docker.com/desktop/windows/install/
[30] "Node-RED Tools", https://nodered.org/.
[31] "Mosquitto Documentation", https://mosquitto.org/.
[32] S. Mirzaie, M. AvazAghaei and O. Bushehrian, "Anomaly Detection in Urban Water Distribution Grids Using Fog Computing Architecture," *2021* 29th Iranian Conference on Electrical Engineering (ICEE), pp. 591-595, 2021.
[33] "Epanet User Manual", https://epanet22.readthedocs.io/en/latest/, 2020.
[34] https://scikit-learn.org/ stable/ modules / clustering.html#clustering-performance-evaluation.

**Sara Mirzaie** received her B.Sc. degree in Computer Software Engineering from Shahid Chamran University, Ahvaz, Iran, in 2007, and M.Sc. degree in Computer Network Engineering from the Yazd University, Yazd, Iran, in 2010. She is a Ph.D. candidate in Computer Engineering at Shiraz University of Technology, Shiraz, Iran. Her research interests are: IoT, Smart Cities, Anomaly Detection and Data Mining.

**Mohammad Reza AvazAghaei** received his B.Sc. degree in Computer Engineering from Fasa University, Fasa, Iran, in 2018, and M.Sc. degree in Computer Network Engineering from Shiraz University of Technology, Shiraz, Iran, in 2021. His research interests are IoT, Anomaly Detection and Network programing.



[ DOI: 10.52547/ijict.13.3.12 ]

**Omid Bushehrian** received his B.Sc. in Software Engineering from Amirkabir University of Technology (Tehran poly-techniques) in 2001. He received his M.Sc. and Ph.D. degrees from Iran University of Science and Tech (IUST) in Software Engineering in 2003 and 2008 respectively. He is currently an Associate Professor at Shiraz University of Technology working on different areas related to the Distributed Computing. His research interests are IoT, Application Migration to Cloud and Distributed and Large-Scale Systems. He also has been working in telecom companies since 2008 as software project manager and consultant.



[ DOI: 10.52547/ijict.13.3.12 ]