A modified LOF based approach for outlier characterization in IoT

Lynda Boukela, Gongxuan Zhang, Meziane Yacoub, Samia Bouzefrane, Sajjad Bagheri, Hamed Jelodar

A modified LOF based approach for outlier characterization in IoT

Lynda Boukela[1*] · **Gongxuan Zhang**[1] · **Meziane Yacoub**[2] · **Samia Bouzefrane**[2] · **Sajjad Bagheri Baba Ahmadi**[1] · **Hamed Jelodar**[1]

Abstract The Internet of Things (IoT) is a growing paradigm that is revolutionary for Information and Communication Technology (ICT) because it gathers numerous application domains by integrating several enabling technologies. Outlier detection is a field of tremendous importance, including in IoT. In previous works on outlier detection, the proposed methods mainly tackled the efficacy and the efficiency challenges. However, a growing interest in the interpretation of the detected anomalies has been noticed by the research community, and some works have already contributed in this direction. Furthermore, characterizing anomalous events in IoT-related problems has not been conducted. Hence, in this paper, we introduce our modified Local Outlier Factor (LOF)-based outlier characterization approach and apply it to enhance the IoT security and reliability. Experiments on both synthetic and real-world datasets show the good performance of our solution.

L. Boukela
E-mail: lyndaboukela@njust.edu.cn

G. Zhang
E-mail: gongxuan@njust.edu.cn

M. Yacoub
E-mail: meziane.yacoub@cnam.fr

S. Bouzefrane
E-mail: samia.bouzefrane@cnam.fr

S. Bagheri
E-mail: s.bagheri@njust.edu.cn

H. Jelodar
E-mail: Jelodar@njust.edu.cn

* Corresponding author
1 School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei Street, Nanjing 210094, China
2 CEDRIC lab, Conservatoire National des Arts et Métiers, 292 rue Saint Martin 75141, Paris Cédex 03, France

Keywords Outlier characterization · Internet of Things · Local Outlier Factor · Security

1 Introduction

Anomaly detection is a sub-field of the data analysis field. It is of great importance for various application domains, such as fraud detection in bank transactions, fault detection in control systems, and intrusion detection in computer security. Outlier detection includes the techniques that allow one to find the data points that show unexpected patterns and that do not follow the majority of the data. Several algorithms have been proposed to tackle the issues related to the detection efficiency and efficacy. However, a challenge to which few contributions have been arisen is the anomaly characteristics finding and reporting. In other works, the characterization of outliers is referred to as outlying aspect mining [1] and outlying property detection [2]. Outlier characterization is a new concept that has not been widely applied; however, it is of great utility. Indeed, finding the characteristics of the anomalies will provide the analyst with more information about the outlier, especially about the reason behind its anomalous aspect. In addition, automatically discovering the outlier characteristics helps in saving time and effort. Furthermore, the discovered information can be reused for reinforcing detection methods, by targeting the same outlier types with their characteristics.

The Internet of Things (IoT) is a new paradigm that integrates several enabling technologies such as sensing, wireless networks, communication protocols, embedded computing, data storage, and distributed services. IoT is believed to be revolutionary for Information and Communication Technology (ICT) and for personal and professional life. Indeed, this paradigm offers numerous applications in health care, environmental monitoring, smart cities, the commercial industry, etc.[3]

Outlier detection techniques have been widely used for anomalous and faulty event identification in IoT. However, there has been no work on the characterization of the detected outliers, while this can be of an enormous importance. If we take the example of fault detection then, it would be interesting to know why a sensor is faulty, whether it is due to an external or an internal factor, sensor ageing, hardware malfunction, battery drain, cyber attack, etc. [4]. By gathering data with appropriate features describing each property of the sensors, if an anomalous behavior is characterized after detection, the network administrator will be able to automatically and rapidly identify the reason behind the anomalous behavior, and will be able to take the necessary measures to ensure the reliability of the network and to limit the loss.

In the majority of works, outliers are characterized by the subspaces that contribute to their unexpected patterns with respect to the rest of the data. However, mining for the relevant subspaces is a challenging task [5]. Firstly, in high dimensional datasets, the search space is very large because the number of potential subspaces grows exponentially with the number of features. In some

works [2][6], the relevant subspaces are searched in all possible combinations, making these solutions computationally expensive. On the other hand, some works [7] have presented feature ranking-based solutions. Although the feature ranking solutions are faster, they suffer from a low characterization efficacy and precision, especially in high dimensional data. Secondly, there are two problems related to the scoring function. On the one hand, the function can suffer from bias; in this case, scores of subspaces of different sizes are not comparable. On the other hand, setting a threshold above which a subspace is considered as relevant is not an easy task. Finally, when the solution is based on the nearest neighbors, another issue arises. As explained in [5], the choice of the nearest neighbors of the considered outlier is very important, especially when the data are multiply distributed.

The above-mentioned issues are addressed in the present work. In the proposed solution, the features are ranked based on their relevance for the considered outlier. The relevance is measured for each data attribute individually, i.e., without exploring different feature combinations. Thus, the search space is considerably restricted, and the efficiency is enhanced. In addition, characterizing outliers in high dimensional data is tackled by vertically partitioning the data, allowing, in this way, the improvement of the characterization precision. Furthermore, by adopting a feature ranking solution, the problem of subspaces comparability is not encountered. Our approach is based on the Local Outlier Factor (LOF) [8] as a scoring function. With this function, the threshold can be easily set. Indeed, a baseline value of approximately 1 allows one to differentiate an outlier from an inlier. LOF has been originally proposed for outlier detection in the full feature space; in our case, the algorithm is adopted to the characterization task and is modified in order to tackle the problem of meaningful nearest neighbor queries.

The main contributions in the present article are as follows:

- An approach for outlier characterization in the IoT is proposed. It aims to improve the security and reliability of the IoT.
- A modified LOF is used as the features scoring function. The function evaluates the relevance of each feature w.r.t. the meaningful nearest neighbors of the considered outlier and has the advantage of the threshold choice facility.
- The proposed approach tackles the problem of outlier characterization in high dimensional data by relying on vertical data partitioning.
- The proposed method is evaluated using both synthetic and real-world datasets. Comparison results show that it outperforms existing solutions.

The rest of this paper is organized as follows. The related works are reviewed in Section 2. Section 3 presents the details of the proposed approach. In Section 4, the evaluation of the method is conducted, in addition to the comparison. Section 5 concludes the article.

2 Related work

In this section, we review some works related to anomaly detection in IoT, in addition to the outlier characterization strategies.

Anomaly detection is a well established field. Multiple outlier detection methods have been proposed and can be broadly categorized into supervised and unsupervised techniques [9]. A finer classification of these methods categorizes them into clustering-based, classification-based, nearest-neighbor-based, distance-based, density-based, etc. Anomaly detection methods have been applied in various domains including IoT. Indeed, anomaly detection techniques have been adopted to solve problems at different levels of the IoT architecture. For instance, data from sensing nodes are gathered and analyzed to detect any faulty records [4][10][11]. At the network level, several works have proposed intrusion detection solutions [12][13]. At the application level, abnormal behaviors have been discovered in power consumption [14], in assisted living systems [15], in autonomous vehicles [16], etc.

Traditionally, for fault detection in sensor networks, the different fault types are manually modeled beforehand. Lately, more autonomous and adaptive methods have been proposed [10][11]. However, with these methods, especially when applied to multivariate data, no additional information about the faults is brought to the analyst. The same issue can be noticed in the intrusion detection solutions [12][13] where no explanations of the attacks are reported, while with the growing interest in unseen attacks detection, the interpretation and characterization becomes necessary. To the best of our knowledge, outlier characterization has not been conducted on any IoT-related problem. Therefore, in the present work, we propose a new outlier characterization approach to interpret the anomalies in IoT, more precisely, we characterize attacks and faulty records in sensor networks.

Lately, the outlier characterization issue has gained a great interest. Indeed, it is important to gain a better understanding and interpretation of an outlier after its detection. Most of the time, the outliers are characterized by their relevant subspaces. In some works, although the relevant subspaces of the outliers are examined in order to improve the detection accuracy, these subspaces are not reported to the analyst [17] [18] [19]. However, in other works, the outlier characterization has been the main contribution.

One of the earliest works on outlier characterization was presented in [1], herein, the authors use a measure based on a simple concept of relative frequency to score the exceptionality of combinations of attribute values featured by a given anomaly with respect to the entire data set or its subset. However, this solution considers only categorical attributes. In [2], the OAMiner is proposed, and it relies on a systematic search of the relevant subspaces and adopts an in-depth-first strategy to explore the search space. The rank based on kernel density estimate of the outlier in a subspace is used as a scoring function. Later, the subspaces in which the outlier has the best ranking are retained as the relevant subspaces. The authors in [6] have presented a set of scoring functions and analyzed them in terms of the different desiderata that they

should possess, especially, the dimension unbiasedness. In addition, the beam search strategy has been adopted in order to explore the subspaces for the considered outlier query with the discussed scoring functions. The OARank framework was presented in [7]. It is a two-stage outlier characterization approach (the second stage being optional, if it is considered that the framework is called OARank search). The first stage is a feature ranking based on mutual information, and the second stage is an efficient score-and-search strategy conducted on the features obtained in the first stage. In [20], a technique based on supervised feature selection is proposed. For each outlier, the method constructs a binary classifier to separate the outlier from the inliers. The class of the outlier is constructed by a set of samples generated as a Gaussian distribution centered in the outlier, while the inliers class is composed of specific samples from the rest of the data. Intuitively, the scoring function herein is the classification accuracy; i.e., the features that allow one to obtain well-separable classes (i.e. high accuracy) will be considered as a good subspace for characterization. For the experiments, the authors adopted two different feature selection techniques, namely, forward selection by SVM and lasso.

The solutions presented in [2] and [6] rely on a systematic search strategy for finding the relevant subspace of each outlier, so they are computationally expensive. On the other hand, in [20], the method is based on a feature selection strategy and the solution in [7] is hybrid. These two methods are computationally better than the systematic search-based methods; however, their characterization precision is low, especially in high dimensional data. In our case, the proposed approach is a feature ranking one, so it is efficient and appropriate for IoT. It is also effective in terms of characterization due to the proposed scoring function and vertical data partitioning.

3 Outlier characterization

In this section, we present the modified LOF-based scoring function to quantify feature relevance. Next, we explain our approach to deal with high dimensional data and to make our scoring function scalable with respect to data dimension. Subsequently, we present the general approach.

3.1 A scoring function based on a modified LOF

LOF is a density-based anomaly detection approach that assigns a degree of outlierness to each point in the data. To compute the degree of outlierness, firstly, a k-distance is assigned to each point p that represents its distance to the k-th nearest neighbor. Hence, the k-nearest neighbors of p are all the points within this distance. Secondly, in order to reduce the statistical fluctuations of the distances of point p to its nearest neighbors, the actual distances are replaced by the k-distance of p. Thirdly, the local reachability density of point p is computed. This density is defined as the inverse of the average reachability

distance based on the k-nearest neighbors of p. Finally, the local outlier factor is computed as the average of the ratio of the local reachability density of p and those of p's k-nearest neighbors. For interested readers, more details on the algorithm can be found in [8].

Unlike conventional works that use LOF for anomaly detection, in our case, we use LOF as a feature scoring function. Hence, we bring the following changes to the algorithm:

3.1.1 Two parameters: C and K

The original algorithm uses two bounds for the number of nearest neighbors, an upper and a lower bound. LOF is then computed by using all the values ranging within the two bounds, and the maximum is retained as the final result. In our case, we also use two parameters C and K for the number of nearest neighbors. However, their functions are different.

We define the first parameter C as the number of nearest neighbors of a point p in the full feature space. This parameter should be large enough to capture the data points that belong to the same class and that we assume as having the same generating mechanism as p. Indeed, as explained in [5], nearest neighbor queries are both theoretically and practically meaningful if the search is limited to objects from the same cluster (distribution) as the query point (i.e., any point belonging to the same cluster is considered a valid answer to a nearest neighbor query).

The second parameter K is defined as the number of nearest neighbors of point p in a specific feature dimension f. K should be smaller than C, and the K-nearest neighbors are found in the ensemble of C-nearest neighbors. Herein, K has the same function as the parameter k defined in the original algorithm, the only difference being that the K-nearest neighbors are found at the level of each feature dimension; hence, point p can have different sets of neighbors from a feature dimension to another.

3.1.2 Dimension-wise LOF (DLOF)

In the original algorithm, LOF is defined to compute the degree of outlierness of a data point in the full feature set, while in our case, the LOF of a point p is computed at each feature dimension f. This modified LOF is used to quantify the relevance of a feature dimension f for characterizing a given outlier o.

Suppose **DS** is an arbitrary d-dimensional dataset of n points, and $\mathbf{F}=f_1,f_2,$ ..., f_d is the feature set. Let the point $o \in$ **DS** be an outlier. The computation of the LOF of o at the level of the feature dimension f_i is conducted as follows:

Firstly, we find C-distance(o), the distance of point o to its C-th nearest neighbor in **DS** and in the full feature space. Subsequently, we find $N_C(o)$, the C-nearest neighbors of o, in **DS** and in the full feature space.

Secondly, we compute the $K\text{-}distance_{fi}(o)$, the distance of o to its K-th nearest neighbor in $N_C(o)$ and in the feature dimension f_i. Later, we find the

set $N_{K,fi}(o)$ of K-nearest neighbors of the outlier o in the C-nearest neighbors $N_C(o)$ and in the feature dimension f_i.

Thirdly, the $K\text{-}distance_{fi}(o)$ is used to compute the reachability distance of the outlier o with respect to point p, denoted $reachDist_{K,fi}(o,p)$, as in the following:

$$reachDist_{K,fi}(o,p) = max\{K - distance_{fi}(p), d(o,p)\} \tag{1}$$

Later, the local reachability density in the feature dimension fi of the outlier o is defined as

$$lrd_{fi}(o) = 1/\left[\frac{\sum_{p\in N_{K,fi}(o)} reachDist_{K,fi}(o,p)}{|N_{K,fi}(o)|}\right] \tag{2}$$

Finally, the local outlier factor of an outlier o in the feature dimension fi can be computed as follows:

$$LOF_{fi}(o) = \frac{\sum_{p\in N_{K,fi}(o)} \frac{lrd_{fi}(p)}{lrd_{fi}(o)}}{|N_{K,fi}(o)|} \tag{3}$$

where $lrd_{fi}(p)$ is the reachability density of the point p, which is computed in the same way as for the outlier o.

3.2 Dealing with high dimensional data

LOF has several advantages as mentioned previously. However, since it is based on the distance measure for computing the density, and due to the curse of dimensionality, the algorithm does not perform well in high dimensional data. Especially, in our case, previously defined LOF will not perform well in finding the nearest neighbors in the full feature set.

In order to tackle this problem, in our case, we transform the problem of characterizing an outlier in a high dimensional dataset into a characterization in an ensemble of data partitions, i.e. outlier characterization in vertical data partitions of the original data. Hence, instead of computing the LOF in the full feature space, we divide the feature set into several subsets of size s.

3.3 General approach

In order to characterize a set of outliers in a given dataset by using our approach, we proceed as follows:

Algorithm 1 Local Outlierness Degrees Computation

Input:
DS: nxd tabular dataset
F : Feature set of size d
s : Size of feature subset
OS: Set of m outliers
C, K : LOF parameters
Output:
LOD : mxd table of local outlier factors of $o \in OS$ in $f \in FS$ (Initially **LOD** $= \emptyset$)
Begin
$subNbr = d/s$; //number of subspaces
Divide **F** into $subNbr$ subsets **FS** of size s. //if (d modulo s)is not equal to 0, then the last subset will be of size s + (d modulo s)
for $i \leftarrow 1, subNbr$ **do**
  Generate the data partition **DSub** of **DS** in the feature subset $\mathbf{FS}_i$
  for $j \leftarrow 1, sizeof(\mathbf{FS}_i)$ **do**
    $\mathbf{LOD}_{1..n,j}$= DLOF(**DSub**, **OS**, j, C, K)
    LOD ={**LOD**, $\mathbf{LOD}_{1..n,j}$};
  end for
end for
End

Algorithm 2 Dimension-wise Local Outlier Factor (DLOF)

Input:
DS: nxs tabular dataset
OS: Set of m outliers
f : Feature index
C: Number of nearest neighbors in the full feature set
K: Number of nearest neighbors in feature dimension f
Output:
LOD : $mx1$ table of local outlier factors of **OS** in feature dimension f
Begin
for $i \leftarrow 1, n$ **do**
  Assign to point p_i its C-distance
  Find the C-NNs of point p_i, $N_C(pi)$
  Assign to point p_i its K-distance, $K\text{-}distance_f(p_i)$
  Find the K-NNs of point p_i, $N_{K,f}(pi)$
end for
for $i \leftarrow 1, m$ **do**
  Compute the reachability distance of outlier o_i based on equation 1
  Compute the reachability density of outlier o_i based on equation 2
  Compute the LOF of outlier o_i based on equation 3
  LOD(i) = LOF;
end for
End

Step 1. DLOF computation : At this step, we assume that the data are normalized to avoid problems related to features belonging to different ranges.

In order to implement the solution that has been discussed in Subsection 3.2, i.e. to deal with high dimensional data, we follow the instructions in Algorithm 1. As we can see, the dataset **DS** is vertically partitioned by splitting the feature set **F** into several subsets **FS** of size s. Subsequently, the LOF of the outliers in **OS** is computed in each feature dimension f of each data

partition **DSub**. The resulted LOFs are then saved in matrix **LOD**. The LOF computation is presented in Algorithm 2, where after the C-nearest neighbors and the K- nearest neighbors of each point $p \in \mathbf{DS}$ are computed, the LOF of each outlier $o \in \mathbf{OS}$ is computed based on Equations 1, 2, and 3.

Step 2. Characterization : After computing the LOFs of the outliers at each feature dimension as explained in the first step, the resulted LOFs are used in order to discover the most relevant features for each outlier. To this purpose, as can be seen in Algorithm 3, for each outlier, the feature dimensions where an LOF greater than a predefined threshold is obtained are retained as its characterizing features. The user also has the ability to choose a minimum and a maximum number of characterizing features.

Algorithm 3 Outlier Characterization

Input:
LOD: mxd matrice of LOFs of m outliers in each feature dimension
t : threshold
$MinNbr$: minimum number of features
$MaxNbr$: maximum number of features
Output:
characterizingFeatures : $mx1$ table of lists of relevant features

Begin
for $i \leftarrow 1, m$ **do**
  characterizingFeatures(i)= $IndiceOf$(**LOD**(i) $> t$)
  [**Idx, R**] = $descendingOrder$(**LOD**($i1, i2, ..., ij$))
  If (sizeof(**characterizingFeatures**(i)) $<$ $MinNbr$) **then**
  characterizingFeatures(i)= Idx(i,j=1..MinNbr)
  ElseIf (sizeof(**characterizingFeatures**(i)) $>$ $MaxNbr$) **then**
  characterizingFeatures(i) = **characterizingFeatures**(i,j=1..MaxNbr)
  EndIf
end for

End

4 Experiments

In order to evaluate the performance of the proposed outlier characterization approach, a set of experiments has been conducted on both synthetic and real-world datasets.

4.1 Synthetic data

The synthetic dataset introduced in [17] has been used for evaluating our approach. The collection includes datasets with 1000 normal points and 19–136 outliers. The datasets are of different dimensions, more precisely, 10, 20,

30, 40, 50, 75, and 100 dimensions. The specificity of these datasets consists in the fact that the outliers are hidden in dataset subspaces of 2–5 dimensions, so the detection and characterization are challenging tasks.

Since the ground truth regarding the relevant features for the outliers is available in these datasets, for evaluating and comparing the performance of the proposed approach, we use the following measures:

$$Precision(T, P) \triangleq \frac{|T \cap P|}{|P|} \tag{4}$$

$$Jaccard(T, P) \triangleq \frac{|T \cap P|}{|T \cup P|} \tag{5}$$

In both measures, T represents the true characterizing features, and P represents the discovered characterizing features. For each outlier, the precision depicts the fraction of correctly predicted features among all discovered features, while the Jaccard index depicts the similarity between the discovered features and the true characterizing features.

We compare our method's results with those of the OARank+search and svmFS approaches presented in [7] and[20], respectively, where the same synthetic datasets and the same evaluation metrics have been used. Figure 1 presents the obtained average precision and Jaccard index by the three approaches over all the outliers and in each dataset. Our approach's results have been obtained by computing the LOFs with the first parameter C=300 and the second parameter K=65. For the characterization, the threshold t for the decision on the feature relevance was set to 1.3, while the minimum number and maximum number of features to retain were set to 2 and 5, respectively. The comparison shows that our approach outperforms both svmFS and OARank+search approaches in terms of the two evaluation measures and almost for all the datasets. This can be explained by the fact that these approaches transform the problem of characterization into a feature selection one, so the returned features characterize a whole set of data points instead of the only considered outlier.

4.2 Real-world data

In order to improve the reliability and security in the IoT, we have applied the proposed approach on two real-world datasets. More precisely, we gain more information about some attacks in the NSLKDD dataset [21] and about faults in the Intel Berkeley Research Lab (IBRL) dataset [22]. In the NSLKDD dataset, the U2R and R2L are analyzed and in the IBRL dataset, some faults are injected and then characterized.

The IBRL dataset was collected in the Intel Berkeley Research Lab where a network of 54 Mica2Dot sensors was deployed. The collection was carried between February 28 and April 25, 2004. Four sensory measurements were collected, namely, humidity, temperature, light, and voltage. Temperature is in

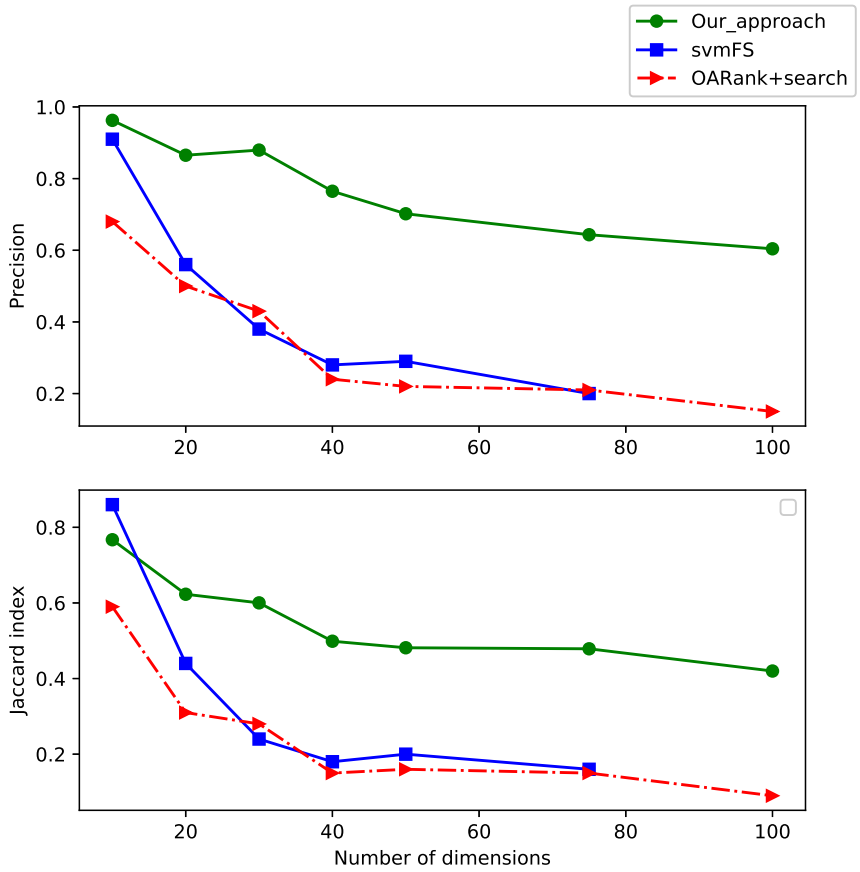

Fig. 1 Results of the comparison of our characterization approach in terms of Precision and Jaccard index

degrees Celsius. Humidity is temperature-corrected relative humidity, ranging from 0 to 100%. Light is in lux (a value of 1 lux corresponds to moonlight, 400 lux to a bright office, and 100,000 lux to full sunlight.). Voltage is expressed in volts, ranging from 2 to 3. For our experiment, we use the data collected on February 28 in the afternoon (after 2 pm) from Sensors 34, 35, and 36. Several faults can occur in the sensor networks and their impact can be reflected in the gathered data as explained in [4]. In our case, we alter some data records in order to obtain faults of types offset in Sensors 34 and 35 and stuck-at in Sensor 36.

The faults and the results of their characterization in the different sensors are presented in Table 1. Our approach has been applied with C=200 and K=65 for computing the LOFs, the threshold t=1.3, and the number of features to return was set to a minimum of 1 and a maximum of 3. As can be seen,

Table 1 Results of offset and stuck-at faults characterization in the IBRL dataset

Sensor	Fault	Features(parameter)	Characterization (LOF)
34	59	light(500)	**{light (0.85)}**
	116	temperature(10)	**{temperature (4.65)}**
	131	temperature(5)	**{temperature (3.44)}**
	134	light(800)	**{light (1.44)}**
	344	humidity(10)	{temperature (1.76), humidity (26.7), voltage (2.68)}
	435	humidity(7)	**{humidity (20.18)}**
35	27	temperature(10), humidity(8), light(1000)	**{temperature (8.7), humidity (4.97), light (4.5)}**
	31	temperature(10), light (500)	**{temperature (8.62), light (2.32)}**
	33	temperature(9), humidity (7)	**{temperature (7.76), humidity (4.48)}**
	81	temperature (7), humidity(10), light (700)	**{temperature (5.79), humidity (6.96), light (2.33)}**
	124	temperature (10), light (600)	**temperature (7.71), light (1.76)**
	422	temperature (5), humidity (10)	**{temperature (19.84), humidity (18.92)**}
36	40-80 except 72	temperature (0)	**{temperature ($\tilde{9}$)}**
	72	temperature (0)	{temperature (10.1), voltage (1.32) }

different offsets have been added to each of the three first feature dimensions of the data records from Sensor 34. Different offsets have been added to two or three feature dimensions at the same time in the data records from Sensor 35. In the data from Sensor 36, a stuck-at fault was injected to records from 20 to 80 to which the value 0 has been assigned to the temperature feature. Almost all the faults have been correctly characterized, including the faults in which more than one feature are affected. In addition, the quantification of the fault severity is also accurate. For example, the two faults 116 and 131 in Sensor 34, to which the offsets added are 10 and 5, respectively, have the LOFs 4.65 and 3.44, respectively. Thus, these LOFs can indicate that Fault 116 is more severe than Fault 131. The severity quantification is also accurate for distinguishing important faults characterized with more than one feature. However, in the case of these data, this holds true only when considering errors occurring in a relatively close period of time. Indeed, for example, while the first faults in Sensor 35 occurred at the beginning of the afternoon when the temperature and light are continuously decreasing, Fault 422 occurred at the evening when the light and the temperature are stable; thus, the latter fault is more distinguishable and its LOF is large. In addition, since the faults in Sensor 36 are successive and have the same characterizing features with a relatively similar LOF, the analyst can easily conclude that it is a stuck-at fault.

In the NSLKDD dataset, each record consists of a network connection with 41 attributes, which are labeled as normal or one of the 24 attack types (e.g.,

Table 2 Results of some U2R and R2L attacks characterization in the NSLKDD dataset

Attack	Type	Characterization
1327	pod	{Dst.host.srv.rerror.rate, Srv.diff.host.rate, Dst.host.same.srv.rate}
2567	rootkit	{Dst.bytes, Duration, Src.bytes}
3879	pod	{Dst.host.srv.rerror.rate, Srv.diff.host.rate, Dst.host.same.srv.rate}
4759	rootkit	{Dst.bytes, Duration, Src.bytes}
4806	pod	{Dst.host.srv.rerror.rate, Srv.diff.host.rate, Dst.host.same.srv.rate}
4962	ps	{Dst.bytes}
5869	ps	{Dst.bytes, Dst.host.serro r.rate, Src.bytes}
7640	buffer over-flow	{Num.comp romised, Src.bytes, Dst.bytes}

Probe, DoS, U2R, and R2L). In our case, the U2R and R2L attacks are retained, and the categorical attributes removed. We assume that the attacks have already been detected; thus, in order to help the analyst to understand the attacks, we apply our approach to characterize them. Results of the characterization of some attacks are presented in Table 2, where the relevant features can be seen. Herein, the important point to mention is that some attacks of the same type are characterized by the same features, e.g., the pod attacks 1327, 3879, and 4806.

5 Conclusion

In this article, a new approach for outlier characterization is proposed. Furthermore, the approach has been used to characterize anomalous events in problems related to the reliability and security in IoT. Several problems related to high dimensional data and feature relevance scoring have been tackled in the proposed approach. However, improvements can be brought in the future. For example, for the scoring function, since it is locally based, the strategy for choosing the nearest neighbors can be dependent on the data type. For instance, for time series data, e.g., in the IBRL dataset, it might be more interesting to make nearest neighbor queries by using time as a similarity measure instead of distance. Furthermore, for more efficiency, the LOFs in the different data partitions can be computed at the same time by using parallelization technologies. In addition, the proposed approach can be used for other IoT problems.

Acknowledgements This work was funded by the National Natural Science Foundation of China, Grant number 61272420 and 61472189.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Angiulli, F., Fassetti, F., & Palopoli, L. (2009). Detecting outlying properties of exceptional objects. *ACM Transactions on Database Systems*, 34(1), 1–62.
2. Duan, L., Tang, G., Pei, J., Bailey, J., et al. (2015). Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery*, 29(5), 1116–1151.
3. Asghari, P., Rahmani, A.M., & Javadi, H.H.S. (2019). Internet of Things applications: A systematic review. *Computer Networks*, 148(2019), 241–261.
4. Muhammed, T.,& Shaikh, R.A.(2017). An analysis of fault detection strategies in wireless sensor networks. *Journal of Network and Computer Applications*, 78, 267–287.
5. Zimek, A., Schubert, E., & Kriegel, H.P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5), 363–387.
6. Vinh, N.X., Chan, J., Romano, S., Bailey, J., Leckie, C., Ramamohanarao, K., & Pei, J. (2016). Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery*, 30, 1520–1555.
7. Vinh,N.X., Chan, J., Bailey, J., Leckie, C., Ramamohanarao, K., & Pei, J. (2015). Scalable outlying-inlying aspects discovery via feature ranking. *Advances in Knowledge Discovery and Data Mining*. 422–434.
8. Breunig, M.M., Kriegel, H.P., Ng, R.T., & Sander, J. (2000). LOF: Identifying Density Based Local Outliers. *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104.
9. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey, *ACM Computing Surveys*, 41(3), 1-58.
10. Zidi, S., Moulahi, T., & Alayam, B.(2018). Fault Detection in Wireless Sensor Networks Through SVM Classifier. *IEEE Sensors Journal*, 18(1), 340-347.
11. Dang, T., Tran, M., Le, D., Zalyubovskiy, V.V., Ahn, H., & Choo, H. (2018). Trend-adaptive multi-scale PCA for data fault detection in IoT networks. *2018 International Conference on Information Networking (ICOIN)*, 744-749.
12. Zarpelão, B. B., Miani, R. S., Kawakan, C. T. & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84, 25-37.
13. Pajouh, H. H., Javidan, R., Khayami, R., Ali, D. & Choo, K. K. R. (2016). A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, 7(2), 314 - 323.
14. Jakkula, V., & Cook, D. (2010). Outlier Detection in Smart Environment Structured Power Datasets. *Sixth International Conference on Intelligent Environments*, 29-33.
15. Zhu, C., Sheng, W. & Liu, M. (2015). Wearable sensor-based behavioral anomaly detection in smart assisted living systems. *IEEE Transactions on Automation Science and Engineering*, 12(4), 1225-1234.
16. Alotibi, F., & Abdelhakim, M. (2020). Anomaly Detection for Cooperative Adaptive Cruise Control in Autonomous Vehicles Using Statistical Learning and Kinematic Model. *IEEE Transactions on Intelligent Transportation Systems*.
doi: 10.1109/TITS.2020.2983392.
17. Keller, F., Müller, E., & Böhm, K. (2012). Hics: high contrast subspaces for density-based outlier ranking. *Proceedings of the 28th International Conference on Data Engineering (ICDE)*, 1037–1048
18. Kriegel, H.P., Kroger, P.,Schubert, & Zimek, A. (2009). Outlier detection in axis-parallel subspaces of high dimensional data. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 831-838.
19. Zhao, X., Zhang, J., & Qin, X.(2017). Loma: A local outlier mining algorithm based on attribute relevance analysis. *Expert Systems with Applications*, 84, 272–280.
20. Micenkova, B., Dang, X.H., Assent, I., & Ng, R.T. (2013). Explaining outliers by subspace separability. *2013 IEEE 13th International Conference on Data Mining (ICDM)*, 518–527.
21. http://nsl.cs.unb.ca/NSL-KDD/
22. http://db.csail.mit.edu/labdata/labdata.html.